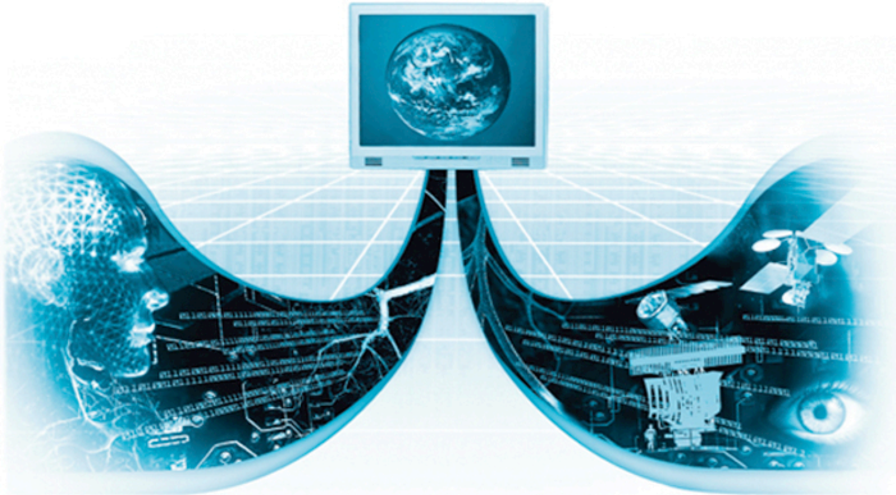
Éloi Bossé
Jean Roy
Steve Wark



CONCEPTS, MODELS, AND TOOLS FOR INFORMATION FUSION

Concepts, Models, and Tools for Information Fusion

For a listing of recent titles in the *Artech House Intelligence and Information Operations Library*, turn to the back of this book.

Concepts, Models, and Tools for Information Fusion

Éloi Bossé
Jean Roy
Steve Wark

ARTECH HOUSE
BOSTON | LONDON
artechhouse.com

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the U.S. Library of Congress.

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library.

Cover design by Igor Valdman

685 Canton Street
Norwood, MA 02062

10 9 8 7 6 5 4 3 2 1

Contents

Foreword

The essence of command and control in military and public security operations is people making timely decisions in the face of uncertainty and acting on them. At the heart of this process is the provision of decision quality information to the decision-maker, thereby enabling timely situation awareness. This is a timeless requirement, which has been immeasurably complicated by the overwhelming and increasing volume of raw data and information available in the current age.

In this context, data and information fusion clearly has a critical role to play in future command and control systems; it is a key enabler in achieving high-quality situation awareness for optimal decision-making. Whatever the definition used, data and information fusion is not something that happens in a vacuum, and it should not be decoupled from the decision-making process. This is the reason why this book has reviewed many existing models of decision-making and tried to put in perspective information fusion with respect to the overall command and control process.

Data fusion can be characterized broadly as the process of utilizing one or more data sources over time to assemble a representation of aspects of interest in an environment. The traditional roots of the data fusion community are in *sensor fusion*, where the "data sources" are established sensors and the "aspects of interest in the environment" are moving objects, each typically represented by a set of state vectors. However, demands on the data fusion community are beginning to exceed this narrow intent in at least two respects.

First, in the national security context, the threat of terrorism and the impetus of network centric warfare (NCW) are expanding the "aspects of interest in the environment" beyond military target tracking considerations to include issues pertaining to: biography, economy, society, transport and telecommunications, geography, and politics, in addition to combinations of the aforementioned. Commensurately, the "data sources" are exceeding military sensor systems to include: communication systems, databases, Web sites, public media, human sources, and so forth. Moreover, the demand for this kind of capability is also emerging in the commercial and wider social context, as advances in transport and telecommunications foster a cascading effect across organizational location, structure, function, and change. Globalization, decentralization, and strategic alliance are creating a need for machine based data fusion that can process more information to allow faster decisions by fewer people. The increasing demand requires our machines to extend well beyond sensor fusion into so-called *higher-level fusion*.

This effort could not have been undertaken and the results achieved without the TTCP (The Technical Cooperation Program, http://www.dtic.mil/ttcp/) organization and the many face-to-face workshops/meetings (three per year) and the monthly video teleconferences devoted to the development of the individual chapters for this book over the course of five years by the members of the Information Fusion Technical Panel (TP1). In November 2006, the NAMRAD (Non-Atomic Research And Development) Principals approved a 2006 achievement award to the TP1 team cited as:

> This award is made for a significant contribution in development of a scientific book on "Concepts, Models, and Tools for Information Fusion" that is being submitted to Artech House. Each nation has in-house programs that provided substantially to the development of the book. Canada has extensive research activities involving models of Situation Analysis. Australia has novel research in agent based approaches that addresses machine data fusion and mental data fusion. The United States had broad research activities addressing information fusion architectures, and the United Kingdom provided key personnel and programmatic support to international research programs such as the Coalition Agents eXperiment (CoAX). Information Fusion is of paramount importance in supporting coalition operations in both peace and wartime, and has been recognized as a fundamental requirement for future military and national security operations. The distinctive accomplishments of the Concepts, Models and Tools for Information Fusion Team reflect great credit upon themselves, their individual Defense Organizations and The Technical Cooperation Program.

Michael Hinman
Chairman of the C3I Information Fusion Technical Panel (TP1)
Air Force Research Laboratory (AFRL), Information Directorate,
Rome, New York
February 2007

The team members are:

Canada
Defence Research & Development Canada–Valcartier
Decision Support Systems Section
2459 Pie-XI Blvd. North
Quebec (Quebec), G3J 1X5, Canada
Éloi Bossé, e-mail: eloi.bosse@drdc-rddc.gc.ca
Jean Roy, e-mail: jean.roy@drdc-rddc.gc.ca

United Kingdom
Defence Science and Technology Laboratory, H56
St. Andrews Road
Great Malvern
WORCS WR14 3PS
United Kingdom
John Howes (retired)

Australia
Defense Science and Technology Organization (DSTO)
Information Technology Division
L205
P.O. Box 1500
Salisbury, SA, 5108
Australia
Dale Lambert, e-mail: Dale.Lambert@dsto.defence.gov.au
Steve Wark, e-mail: Steven.Wark@dsto.defence.gov.au

United States of America
SPAWAR Systems Center San Diego
53560 Hull Street, # 2731
San Diego, CA 92152
United States
Mark Owen, e-mail: mark.owen@navy.mil
Frank White (retired)

U.S. Army CERDEC
Intelligence & Information Warfare Directorate
Bldg. 600
Attn: AMSRD-CER-IW-BM
Fort Monmouth, NJ 07703-5211
United States
Joseph A. Karakowski, e-mail: Joseph.A.Karakowski@us.army.mil

CHAPTER 1

Introduction

Understanding command and control (C2) is no longer an option; it is a requirement [1]. We need to understand C2 thoroughly if we want to make significant progress on defense and public-security transformation or succeed in twenty-first-century operations.

Clearly, the challenges of twenty-first-century missions have increased significantly. Today's missions differ from traditional missions as they are simultaneously more complex and more dynamic, requiring the collective capabilities and efforts of many organizations in order to succeed [1]. This requirement for assembling a diverse set of capabilities and organizations into an effective partnership is accompanied by shrinking windows of response opportunity.

Simply stated, C2 is the common military term for the management of personnel and resources [2]. This being said, a consensus has not yet been attained as to a single complete and precise definition of the term C2. In this regard, the official definition provided by the U.S. Department of Defense (DoD) in the United States is often quoted. According to the DoD *Dictionary of Military and Associated Terms* [3], C2 is defined as

> The exercise of authority and direction by a properly designated commander over assigned and attached forces in the accomplishment of the mission. Command and control functions are performed through an arrangement of personnel, equipment, communications, facilities, and procedures employed by a commander in planning, directing, coordinating, and controlling forces and operations in the accomplishment of the mission.

Beyond its definition, C2 is not an end in itself but a means toward creating value (e.g., the accomplishment of a mission). Specifically, C2 is about focusing the efforts of a number of entities (individuals and organizations) and resources, including information, toward the achievement of some task, objective, or goal [1]. How C2 (or management) is done, or may have been done, in industry and military organizations should not be equated with why C2 (or management) is needed or what functions need to be successfully performed to create value.

Command and control is a relatively recent term that for millennia was referred to simply as command [2]. Command concepts both predate, and have evolved separately from, politics and industrial management. This is because warfare is qualitatively different from the management of other human enterprises by virtue of its time criticality and the high cost of error. Both of these characteristics of warfare have shaped thinking about C2.

Although the purpose of C2 has not changed since the earliest military forces engaged one another, the way we have thought about it, as well as the means by which the functions of C2 have been accomplished, have changed significantly over the course of history [1].

1.1 Traditional View of the Command-and-Control Process: The OODA Loop

Boyd [4] introduced the observe, orient, decide, and act (OODA) loop in order to support the analysis of pilot decision-making at a tactical level [5]. The idea, illustrated in Figure 1.1, is that decision-making begins with observing the physical domain. The observations are then placed in the context of other information and prior knowledge (so that they become useful information) in order to orient the individual, which (in turn) allows this individual to decide what is to be done and act accordingly. The concept has proved to have considerable intuitive appeal and has been used for decades as the basis of both analysis and training. The phrase "turning inside the enemy's OODA loop," while originating in air-to-air combat, has become the shorthand way of understanding that the speed of the C2 process can provide advantage in combat situations.

In the language of Alberts et al. [5], the act of observation must begin in the physical domain, may pass through some fusion with other observations, and is brought to the individual's attention through the information domain. The process of orientation occurs in the cognitive domain as the information content of the observations is internalized and placed in the context of the individual's prior knowledge, experience, and training. This is seen as providing the basis for a decision—also a cognitive activity. Finally, the decision itself must pass through the information domain (e.g., the controls of an aircraft, the directives of a commander) in order to become the basis for action.

The OODA loop has proven seductively robust and has been applied not only to pilot's activities in air-to-air combat, but also to organizational behavior at all levels. However, in the view of Alberts et al. [5], this is an error. The OODA loop both oversimplifies the command-and-control process in ways that confound analysis and reifies military organizations, implying that they have a single mind

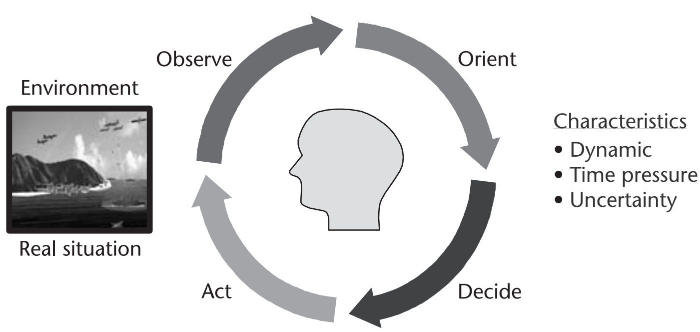


Figure 1.1 The OODA loop model [4].

and make a single, coordinated decision across echelons and functions. Alberts et al. [5] believe that the OODA loop is outdated because it fails to differentiate crucial elements that must be considered in information-age analyses. Moreover, the OODA loop greatly oversimplifies the joint hierarchical model underlying military operations.

1.2 An Information-Age View of the Traditional Command-and-Control Process

Figure 1.2 provides an information-age view of the traditional C2 process as it has been understood for several decades. However, it uses much richer constructs than those in the OODA loop [5]. In contrast to the logic in the simpler OODA loop construct, which sees the output of the cognitive processes as a decision, the information-age C2 process is understood to generate a richer product—command intent. This choice of language has two important, direct implications: (1) the product is much richer, and (2) more than one individual is involved.

As illustrated in Figure 1.2, given that the term *command and control* encompasses as much as it does, its elements span all of the three domains of warfare (i.e., physical, information, cognitive) introduced in Section 1.1. C2 sensors, systems, platforms, and facilities exist in the physical domain. The information collected, posted, pulled, displayed, processed, and stored exists in the information domain. The perceptions and understanding of what this information states and means exist in the cognitive domain. Also in the cognitive domain are the mental models, preconceptions, biases, and values that serve to influence how information is interpreted and understood, as well as the nature of the responses that may be considered. Finally, C2 processes and the interactions between and among individuals and entities that fundamentally define organization and doctrine exist in the social domain (not shown in Figure 1.2).

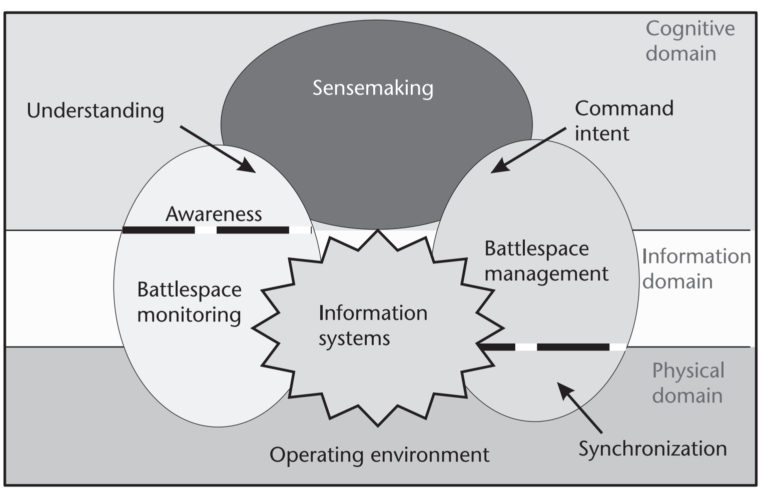


Figure 1.2 An information-age view of the C2 process. (*After:* [5].)

More recently, Alberts and Hayes [1] identified the following functions associated with the command and control (or management) of a given undertaking:

- Establishing intent (the goal or objective);
- Determining roles, responsibilities, and relationships;
- Establishing rules and constraints (e.g., schedules);
- Monitoring and assessing the situation and progress;
- Inspiring, motivating, and engendering trust;
- Training and education;
- Provisioning.

These C2 functions are applicable not only to military endeavors but also to civil-military and, indeed, civilian and industrial enterprises. Each of these functions can be seen in the context of a particular time horizon.

In the continuity of their effort to better understand C2, Alberts and Hayes [1] have also proposed another conceptual model of command and control, shown in Figure 1.3.

According to the U.S. DoD *Dictionary of Military and Associated Terms* [3], *command* is defined as

> The authority that a commander in the Armed Forces lawfully exercises over subordinates by virtue of rank or assignment. Command includes the authority and responsibility for effectively using available resources and for planning the employment of, organizing, directing, coordinating, and controlling military forces for the accomplishment of assigned missions. It also includes responsibility for health, welfare, morale, and discipline of assigned personnel.

This definition subsumes control as a part of command. However, many have tried to draw a distinction between *command* and *control* [2]. Distinctions that

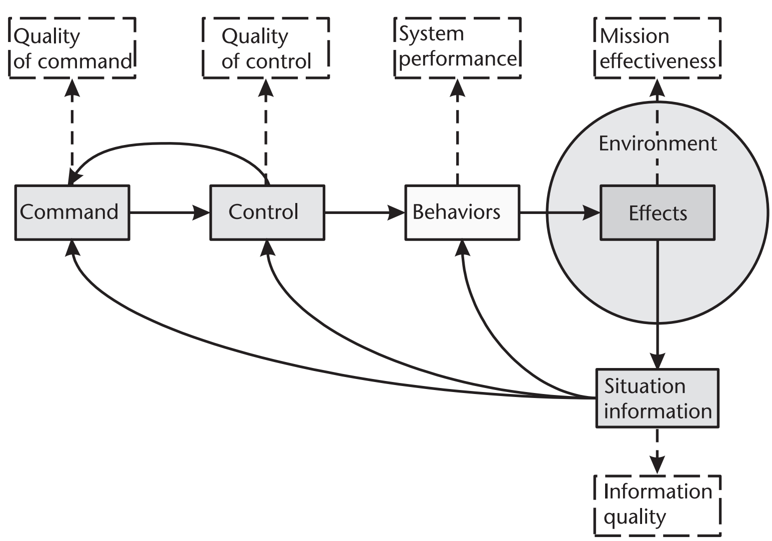


Figure 1.3 Command-and-control conceptual model. (*From:* [1].)

have been drawn define one as art (command) and the other as science (control), and as one being the bailiwick of the commander (command) and the other of the staff (control). Hence, rather than treating C2 as a single concept, Alberts and Hayes [1] have chosen to separate command from control to maintain the greatest degree of flexibility. This enables them to examine each concept on its own and combine different approaches to each in ways that have not been considered before. Thus, they start constructing their conceptual model (Figure 1.3) with two boxes, one representing the concept of command and the other representing the concept of control. Together, these two boxes define the C2 space.

Prior to the commencement of an operation, intent (a command function) needs to be established [1]. This intent can consist of merely recognizing that there is a situation to deal with or a problem to solve. It does not require that a solution or an approach be developed. Roles, responsibilities, and relationships may be predetermined, or they may be established or modified to suit the circumstances (intent and situation). The establishment of a role determines whether or not the entity is considered part of the team or part of the environment. Likewise, rules, constraints, and resource allocations can be predetermined or tailored to the situation.

Once an operation begins (and this dates from the establishment of intent, not from the commencement of a response, where response can include preemptive action), intent can change, as can roles, responsibilities, allocations of resources, and the like [1]. All of these changes to the set of initial conditions, with the exception of a change to intent, should be considered control functions. Changing intent is a command function. The ability to make timely and appropriate changes is directly related to the agility of the specific instantiation of a C2 approach. Given the complexity of the twenty-first-century security environment and the missions that twenty-first-century militaries are, and will be, called upon to accomplish, C2 agility is perhaps the most important attribute of a C2 approach. The establishment and communication of the initial set of conditions, the continuing assessment of the situation, and changes to intent are functions of command. The ability to exercise command (the accomplishment of the functions associated with command) is affected or influenced by, among other things, the quality of information available.

The function of control is to determine whether current or planned efforts are on track [1]. If adjustments are required, the function of control is to make them if they are within the guidelines established by command. The essence of control is to keep the values of specific elements of the operating environment within the bounds established by command, primarily in the form of intent.

Behaviors include [1]:

1. Those actions and interactions among the individuals and organizations that accomplish the functions associated with C2 (e.g., establishing intent, conveying intent);
2. Those that are associated with understanding or making sense of the situation and how to respond;
3. Those that are associated with the response (that is, with creating the desired effects, such as maneuver and engagement).

The first two types of behaviors constitute C2. The second type is a subset of C2 called sense making, while the third type of behaviors can be referred to as actions or execution. All are functions of an enterprise (organization or endeavor). Both the objective of sense making and execution and how they are accomplished are determined by command and control.

Sense making consists of a set of activities or processes in the cognitive and social domains that begins on the edge of the information domain with the perception of available information and ends prior to the taking of action(s) meant to create effects in any or all of the domains. Examples are [1]:

- The employment of kinetic weapons with direct effects in the physical domain and indirect effects in the other domains;
- The employment of psychological or information operations designed to create direct effects in the cognitive and information domains with indirect effects in the physical domain.

The actions involved in execution may take place in any of the domains with direct and indirect effects in multiple domains. The nature of the effects created by a particular action are a function of (1) the action itself, (2) when and under what conditions the action is taken, (3) the quality of the execution, and (4) other related actions. The selection of what actions to take and when to take them is part of the sense-making process.

The operating environment includes everything outside of the C2 processes and the systems that support them [5]. The physical environment (e.g., terrain, weather) is one key dimension. Adversary forces form another. Own forces, to the extent that they are not part of C2 processes, are also in the environment. They represent the most controllable factors in the environment, but even they are imperfectly controllable due to the fog and friction of war. Other, neutral forces may also be present in the portion of the operating environment of interest. Their potential involvement or interference must also be considered. The operating environment also includes a host of political, social, and economic factors and actors, ranging from refugee populations to the infrastructure (e.g., communications, transportation) in the area.

Alberts and Hayes [1] state that the context of C2 can greatly vary. The tasks at hand differ widely in nature, ranging from the creation or transformation of an enterprise at the strategic level, to employing the enterprise in a major undertaking at the operational level, to the completion of a specific task at the tactical level. The nature of the resources involved varies according to the nature of the task. Where as some can be accomplished with organic assets, others requires putting together a large heterogeneous coalition with resources of many types.

In the past, much of the discussion about C2 focused on a single commander, the one in charge. In fact, command and control in modern warfare is a distributed responsibility. Actually, the C2 conceptual model depicted in Figure 1.3 is elemental, or fractal [1]. An enterprise of the complexity necessary to undertake military and civil-military missions will have many concurrent, nested, and even overlapping instances of this elemental model, each one of (or collection of) which may exhibit different C2 approaches. At the enterprise level, the functions associated with command will determine the number and nature of these fractals and the relationships among them. Thus, if we consider Figure 1.3 to be a view at the enterprise level, then there will be a great many little Figures 1.3 contained in the enterprise view of the behaviors box or, for that matter, the boxes for command and for control. Command at one level determines the conditions under which fractals within its purview operate. There will be cases of sovereign fractals in which the fractals are not nested but have peer-to-peer and/or overlapping relationships. In these cases, the functions associated with C2 are achieved in a manner different from that of traditionally nested fractals.

The conceptual model of Figure 1.3 consists of two kinds of concepts: (1) functional, or process, concepts, and (2) concepts related to value. A generic process view of the conceptual model is depicted in Figure 1.4.

The process view organizes functions and processes, whether past, current, or future, into a small number of conceptual bins. The bin "Situation information" represents the host of information-related assets and processes that sense, collect, process, protect, disseminate, and display information. The product of these processes (information) provides data about the environment, including the effects of interest. This information is used as an input to all of the other process concepts.

1.3 About This Book on Data and Information Fusion

As highlighted in the previous section, the essence of command and control in military and public-security operations is people making timely decisions in the face of uncertainty, then acting on them. At the heart of this process is the provision of decision-quality information to the decision-maker, enabling timely situation awareness (SAW). This is a timeless requirement, which has been immeasurably complicated by the overwhelming and increasing volume of raw data and information available in the current age.

In this context, data and information fusion (DIF) will clearly play a critical role in future command-and-control systems; it is a key enabler in achieving high-quality situation awareness for optimal decision-making. An initial lexicon defined

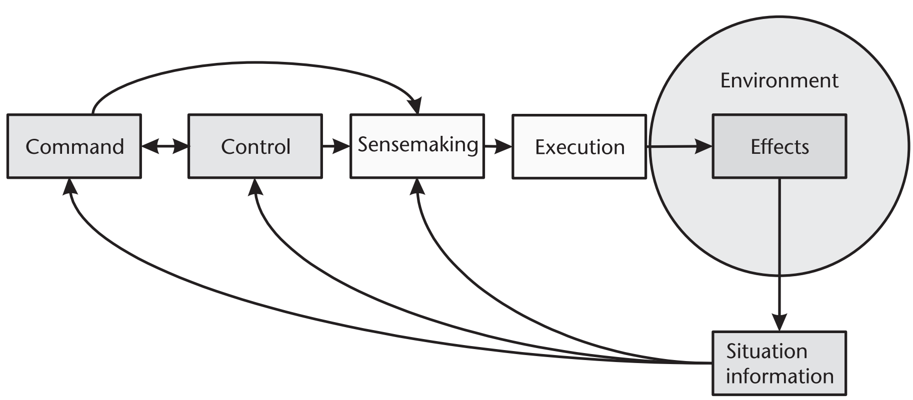


Figure 1.4 C2 conceptual model—a process view. (*From:* [1].)

data fusion as a process dealing with the association, correlation, and combination of data and information from single and multiple sources to achieve refined position and identity estimates, in addition to complete and timely assessments of situations and threats, as well as their significance [6]. As discussed in this book, this definition has evolved over the years, and multiple variants have been proposed.

Whatever the definition used, data and information fusion is not something that happens in a vacuum, and it should not be decoupled from the decision-making process. This is why we have reviewed many existing models of decision-making and tried to put information fusion into perspective with respect to the overall command-and-control process.

As a discipline, data and information fusion draws together concepts from a wide range of diverse fields, such as psychology, human factors, knowledge representation, artificial intelligence, mathematical logic, and signal processing. Most of these aspects are discussed in this book to varying degrees. In fact, this book's content can be organized into three main categories:

1. Concepts, definitions, and models (Chapters 2–5);
2. Mathematical and logical approaches (Chapters 6–9);
3. Computational aspects of information fusion (Chapters 10–12).

1.3.1 Concepts, Definitions, and Models

Chapters 2 through 5 provide the common foundation for the analysis and/or the development of information-fusion capabilities. It does so through a review of many concepts, definitions, and models regarding decision-making, situation analysis (SA) and awareness, and data and information fusion.

Decision-making is involved in all aspects of our lives, and it is of particular importance regarding command-and-control activities in military and public-security settings. Over the years, multiple efforts have thus been deployed to better understand and explain decision-making. As a result, the decision-making process has been the root of several theoretical models. Chapter 2 provides a discussion of two general classes of decision-making models: (1) the rational models, and (2) the naturalistic models.

The term *situation awareness* has emerged as an important concept in dynamic human decision-making. Chapter 3 talks about the benefits of SAW and about the fact that as a general concept, SAW can be of interest in a very large number of settings. It also proposes another concept, *situation analysis*, as an attempt to synthesize the main notions put forward by well-established data-fusion and situation-awareness models.

Chapter 4 addresses data and information fusion. DIF is still expected to play a crucial role in the next generation of support systems for aiding decision-makers in military and public-security operations. Chapter 4 reviews the main models that have been developed over the years to better understand and describe data and information fusion. Certainly, each one of these models has value as it provides particular insights into this important field. Hence, our purpose in describing them is not to argue for one or the other but to give the reader a good sense of these various perspectives, mainly to put the rest of this book into context.

Within the last decade or so, technological development has raised new issues and challenges regarding the design process of fusion, situation analysis, and decision-support systems. Actually, the issues and challenges have shifted from identifying technological possibilities and limitations to determining how these systems must be designed to fit with human information processing.

Chapter 5 discusses various issues related to the design of technological tools and their and insertion into the decision-making process. Supporting information fusion, situation analysis, and decision-making in complex military and public-security operations indeed requires balancing the human-factors perspective with that of the system designer, as well as coordinating efforts in designing a cognitively fitted system.

1.3.2 Mathematical and Logical Approaches to Information Fusion

Knowledge, belief, and uncertainty are three key notions of the situation-analysis process (through data and information fusion). Belief and knowledge representation is a crucial step needed to transform data into knowledge. The data and information coming from the different sources must be converted into a certain language or presented by other means (e.g., visualization) so they can be processed and used by the human to build his mental model in order to decide and act. One great challenge in designing a support system is to make use of the mathematical and logical tools that can allow for measuring and reasoning about the situation using a common analysis framework. A *formalization* is necessary to be able to deal with knowledge or uncertainty, that is, a formal framework in which knowledge, information, and uncertainty can be represented, combined, managed, reduced, increased, and updated. Chapter 6 discusses the key notions of knowledge, belief, and uncertainty in relation to information fusion. The objective is (1) to build a model of the situation directly usable by the different theories of reasoning under uncertainty, and (2) to be able to deal with both knowledge and uncertainty. The potential theoretical frameworks available to model the situation-analysis process can be divided into two main categories: qualitative approaches (Chapter 7) and quantitative approaches (Chapter 8). Qualitative approaches seem better suited to reasoning about knowledge, while quantitative approaches are better candidates for uncertainty representation and management. Hence, a good solution for a global modelization of the situation could be a hybrid approach (Chapter 9) mixing quantified evaluations of uncertainty and high reasoning capabilities.

1.3.3 Computational Aspects of Information Fusion

The last part of the book (Chapters 10–12) reviews the computational implementations of information fusion. It addresses the key characteristics of the information-fusion domain and the performance requirements they impose on information-fusion systems. It reviews the key elements of computational infrastructure relevant to the design and performance of information-fusion systems, including system architecture, computer networks, software middleware, issues with information sources, and human-computer interfaces. We also consider key concepts in knowledge-based and artificial intelligence systems that impact higher-level fusion processes, including expert systems, reasoning systems, neural networks, and computational complexity. Software architectures that can be used to implement information-fusion systems are reviewed, as are issues associated with the blackboard and multiagent architectures as they can be applied to information-fusion systems.

References

[1] Alberts, D. S., and R. E. Hayes, *Understanding Command and Control*, CCRP Publication Series, Department of Defense Command and Control Research Program, 2006, http://www.dodccrp.org, accessed November 10, 2006.

[2] Alberts, D. S., and R. E. Hayes, *Power to the Edge*, CCRP Publication Series, Department of Defense Command and Control Research Program, 2003, http://www.dodccrp.org, accessed November 10, 2006.

[3] Department of Defense, *Department of Defense Dictionary of Military and Associated Terms*. Joint Pubs. 1-02 (as amended through August 31, 2005), May 2006, http://www.dtic.mil/doctrine/jel/doddict, accessed November 10, 2006.

[4] Boyd, J. R., *A Discourse on Winning and Losing*, unpublished briefing, August 1987.

[5] Alberts, D. S., et al., *Understanding Information Age Warfare*, CCRP Publication Series, Department of Defense Command and Control Research Program, 2001, http://www.dodccrp.org, accessed November 10, 2006.

[6] White, F. E., Jr., *Data Fusion Lexicon*, San Diego, CA: Joint Directors of Laboratories, Technical Panel for C3, Data Fusion Subpanel, Naval Ocean Systems Center, 1987.

CHAPTER 2

Decision-Making Models

Jean Roy, Richard Breton, and Robert Rousseau

2.1 Introduction

Decision-making is involved in all aspects of our lives, and it is of particular importance regarding command-and-control activities in military and public-security settings. Over the years, multiple efforts have thus been deployed to better understand and explain decision-making. As a result, the decision-making process has been the root of several theoretical models. Some of these models are tagged as "traditional," by comparison with the new wave of naturalistic decision-making (NDM) models that have been proposed more recently.

Decision-making models can be separated into two principal categories: normative models and descriptive models. Normative models propose a norm, a formal process that people should follow in order to reach the optimal and ideal decision. Descriptive models include a more psychologically valid description of the decision-making process. Researchers also often make a distinction between two types of decision-making: analytical and intuitive decision-making. Analytical decisions are slower, conscious, controlled, and deliberate and demand effort and energy, whereas intuitive decisions correspond to a quick and relatively automatic decision process [1]. Each of these types is activated under particular kinds of circumstances and, respectively, requires distinctive strategies to be achieved.

This chapter, based mostly on the work of Breton and Rousseau [2], does not provide an exhaustive survey of the various decision-making models found in the literature. Instead, the discussion is limited to two general classes of decision-making models:

1. The rational models;
2. The naturalistic models.

The main objective is to compare the rational and naturalistic models according to their capacity to represent the decision-making activity performed in a command-and-control environment. Features of the models related to the consideration of the uncertainty and time-stress factors are highlighted. Some traditional models are briefly presented next, with a focus on the working principles of the decision-making process.

2.2 Rational Models

Rational models range from purely normative models, like the expected utility models, to more descriptive models.

2.2.1 Expected Utility Theory

The expected utility theory (EUT) is the best-known normative model. EUT is in fact a family of models [3]. It models the way people should behave within the decision-making process if they follow some requirements of rational decision-making. According to this theory, the decision-making process must be grounded in the combined evaluation of the probability of occurrence and the value (utility) of the different consequences and alternatives of the situation. Uncertainty is thus represented in EUT by the probability of the occurrence of an event and the probability of a contingent outcome. In fact, EUT is based on an unbounded rationality principle that assumes an unlimited availability of time and information about alternatives and outcomes. It assumes that a decider has complete information on the consequences and probabilities associated with each possible solution. It also assumes that a decision-maker understands this information and is able to calculate, implicitly or explicitly, the advantages and disadvantages of all alternatives.

Formally, the choice and the decision rely on an expected value obtained from the multiplication of the value of an alternative by its probability of occurrence. The result is a utility function enabling a form of scaling of the various gambles from which a decider has to make a choice. A decision-maker compares the utility value of several alternatives and selects the one that maximizes the expected utility. EUT is based on six principles: (1) ordering of alternatives, (2) dominance, (3) cancellation, (4) transitivity, (5) continuity, and (6) invariance. If one (or many) of these principles is violated during the decision-making process, then the expected utility is not maximized. It is generally accepted that most of these principles are often violated by humans, notably the cancellation and transitivity principles. For instance, Lichtenstein and Slovic [4] showed that given a choice between two bets of equal expected value, people base their choice on the probability of winning, but if asked to set a price on how valuable the bet is, they rely on the size of the outcome payoff. Even though EUT does not reproduce human behavior in a number of situations, the elegance of the formal model based on probability theory has maintained interest in that approach.

The EUT supposes that an individual will use objective probability values, based on observed statistics, to conduct the decision-making process. Although very sound on a theoretical basis, such statistics are often not available. Furthermore, it is very likely that the human decision-making process will rely on more cognitive assessments of probability. In consequence, a variant of EUT, called subjective expected utility theory (SEUT), has been developed [5] based on the notion of subjective values.

SEUT proposes that the decision-maker uses a subjective estimation to reach a decision. Thus, the decision-making process concerning each alternative is the result of the combination of the value given to a consequence by an individual, and the value that he or she believes to be its real probability of occurrence. It is important to note that the decision, albeit subjective, is still the outcome of a comparison of utility values.

The EUT and SEUT models represent the uncertainty factor using the concept of probabilities (perhaps subjective) associated with all alternatives and outcomes. However, it implies that the decision-maker has a complete set of all alternatives and knows and understands the related outcomes. Thus, it may be difficult to apply these models to situations like those encountered in command and control, in which often all alternatives cannot be identified. Similarly, the EUT model cannot be applied to the decision-making process in time-pressed situations since it is implicitly assumed that the decision-maker has all the time required for a complete analysis of all alternatives.

2.2.2 Prospect Theory

The prospect theory, a descriptive model developed by Kahneman and Tversky [6], retains the overall structure of the EUT and the notion that the decision is made from the selection of an option from among a set of uncertain gambles. The most noticeable distinction is in the substitution of the concept of value for the concept of utility, referring to a deviation in terms of gains or losses relative to a specified neutral reference point. The value function for losses is different from the value function for gains. The value function for losses is convex and steep, and the function for gains is concave and shallow. Thus, it follows that the theory predicts that a decision-maker will be "risk adverse" when it comes to gains and "risk seeking" when it comes to losses.

Prospect theory also predicts the certainty effect, according to which the reduction of the probability of an event or of a consequence by a constant factor has a greater impact if the event is certain rather than just probable. However, this idea has been modified to add a pseudocertainty effect that includes a more apparent than real certainty.

Prospect theory is essentially a psychological and behavioral theory. It incorporates cognitive factors through a number of subjective assessments of parameters of the model. The inclusion of subjective assessments reduces the amount of mental computation required for exact probability estimates and thus makes the model more appropriate for use in situations of limited mental resources or involving time constraints. Like the EUT, prospect theory requires the knowledge of a complete set of alternatives. Thus, it may be difficult to apply this theory to situations in which all the alternatives cannot be identified. Moreover, by retaining the calculations of values across prospects or gambles, it still requires a systematic analysis and comparison of all known alternatives for determining a choice based on a value criterion.

2.2.3 Regret Theory

The fundamental element of regret theory is counterfactual reasoning, which implies the evaluation of hypothetical events using the consequences of another choice as a reference point. This theory relies on two premises: (1) most people experience sensations like regret, and (2) when decision-making implies uncertainty, decision-makers try to anticipate and account for these kinds of feelings.

2.2.4 Bounded Rationality

Another set of models, the bounded rationality models, relate to situations where a person has to choose between a number of alternatives characterized on a number of dimensions or cues when uncertainty about the outcome is not relevant. In these situations, there is not a single dimension on which the options can be scaled (i.e., value or utility). These models, based on bounded rationality, have two basic characteristics: (1) decisions do not follow a principle of optimization or maximization of expected utility or value, and (2) decisions do not follow an exhaustive analysis of all alternatives.

2.2.5 Satisfying

Simon [7] proposed that people would "satisfy" when making a decision, rather than "optimize" as proposed by EUT. The satisfying theory, one of the first descriptive models ever elaborated, has been developed in the context of organizational decision-making even though it can be applied to individual decision-making. It takes into account the limitation in human information processing by determining a criterion that stops the decision process; this process does not then require an exhaustive search before reaching a decision. Satisfying refers to the elaboration of an aspiration level (a goal) and the selection of the alternatives that enable reaching this goal. Decision-makers seek a satisfactory alternative rather than an optimal one. Thus, a decision-maker chooses the alternative that fulfills the most important needs and requirements of the situation, without wasting time on the evaluation of each alternative to find the best one [8].

Satisfying is a simple heuristic that reduces the cost of examining all alternatives and computing probabilities required by the rational models. However, it does require an important amount of processing in the determination of the actual level of aspiration that will serve as the criterion [9]. Satisfying can deal with uncertainty implicitly since it does not require complete knowledge about a situation. It can accommodate partial ignorance rather than vague information. Because no time is wasted on the evaluation of each alternative, it may represent the decision-making process in situations marked by time pressure.

2.2.6 Heuristic Multiattribute Decision Strategies

A number of heuristics, simple nonoptimal decision rules, have been proposed for multiattribute decision-making [10]. Table 2.1 presents a selection of heuristic strategies based on the following three principles:

1. The information is processed alternativewise or cuewise. Alternativewise strategies process all cue information on a given alternative before going to

Table 2.1 Most Common Strategies in Multiattribute Decision-Making

Strategy Label	*Description*	*Compensatory or Noncompensatory*	*Cuewise or Alternativewise*	*Stopping Rule*
Franklin's rule	This strategy calculates for each alternative the sum of the cue values multiplied by the corresponding cue weights and selects the alternative with the highest score.	Compensatory	Alternativewise	No
Weighted pro	This strategy selects the alternative with the highest sum of pros. A cue that has a higher value for an alternative than for others is considered a pro for that alternative. The weight of a pro is determined by the validity of the particular cue.	Compensatory	Cuewise or alternativewise	No
Lexicographic (LEX)	This strategy selects the alternative with the highest cue value on the cue with the highest validity. If there is a tie, the cue with the second highest validity is considered, and so on.	Noncompensatory	Cuewise	Yes
Recognition	If one of two alternatives is recognized, and the other is not, the recognized alternative is given a higher value. In a set of pairwise comparisons, the alternative with the higher total value is selected.	Compensatory	Alternativewise	No
Elimination by aspects (EBA)	This strategy eliminates all alternatives that do not exceed a specified value on the first cue examined. The procedure is repeated until a single alternative is left. The order is determined by cue validity.	Noncompensatory	Cuewise	Yes

the next alternative. Cuewise strategies compare all alternatives on a given cue before going to the next cue.

2. A strategy may or may not allow for compensation. A compensatory strategy integrates information over cues. Furthermore, a given cue can be outweighed by other cues.
3. A strategy may have a stopping rule that makes possible the interruption of information gathering and analysis.

Rieskamp and Hoffrage [10] report data showing that under high time pressure, deciders tend to adopt a simple noncompensatory strategy like the lexicographic strategy. Under low time pressure, the weighted pro strategy is adopted. Even though it is not explicitly stated, it can be inferred that under uncertainty, noncompensatory strategies are likely to be adopted since they do not require an exhaustive search of information.

2.3 Naturalistic Decision-Making Models

Naturalistic decision-making (NDM) can be defined as "how experienced people, working as individuals or groups in dynamic, uncertain, and often fast-paced environments, identify and assess their situation, make decisions, and take actions whose consequences are meaningful to them and to the larger organization in which they operate" [11]. Thus, NDM can be summarized as an individual's resorting to his or her experience to reach a decision in his or her sphere of activity.

Models and research in NDM are based on some particular factors that appear to characterize and influence decision-making in natural settings. These contextual factors are

- Nonstructured (that is, nonartificial) situations and problems;
- Uncertain and dynamic environments;
- Ill-defined, conflicting, or changing objectives;
- A decision-action-feedback cycle;
- Time pressure;
- Involvement of several individuals;
- Existence of organizational norms and objectives;
- Presence of high and potentially personal stakes.

Thus, NDM defines a very different context for decision-making than the one proposed by traditional models, and this context is also more realistic regarding the situations that decision-makers have to face in their professional lives [11].

2.3.1 Recognition-Primed Decision Model

Elaborated by Klein [12] (see also [13]), the Recognition-Primed Decision (RPD) model stipulates that rather than weighting the advantages and disadvantages of several alternatives, experienced decision-makers use their experience to evaluate a situation and determine a solution or a possible course of action from the first attempt. In that sense, the model can be applied to represent situations characterized by uncertain and incomplete information and marked by time pressure.

The RPD model, illustrated in Figure 2.1, has four principal components. The first is the recognition of the current situation, which is based on the degree of the familiarity of the situation by comparison to one's mental index and experience. In RPD, recognition is applied differently than in multiattribute decisions. Recognition of a valid situation directly activates a process of evaluation. There is no computation of a recognition index leading to a choice among a set of alternatives. Thus, on one hand, if the situation appears unfamiliar, the decision-maker will seek further information. On the other hand, if the situation is categorized as familiar, the second component is applied. This second component consists of understanding the situation in light of expectations, cues, and goals. The third component consists of recalling relevant prototypic actions from prior experience. The fourth component is the evaluation, through mental simulation, of the potential and plausible consequences of each considered course of action. Each action is

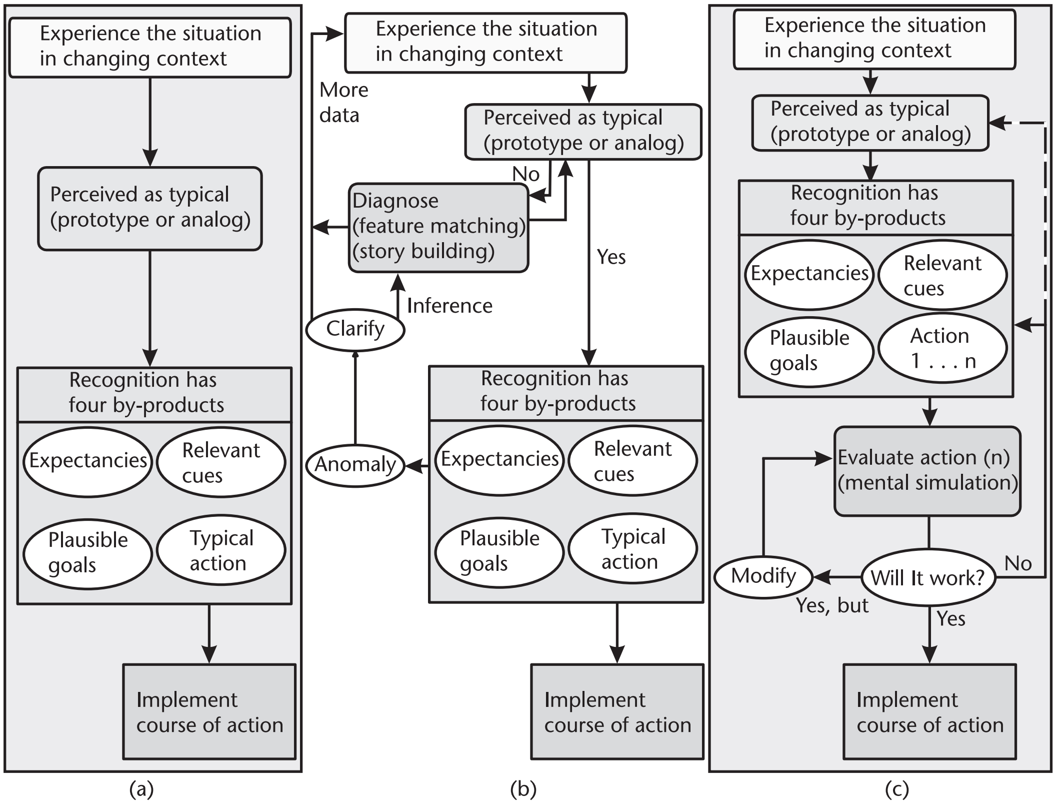


Figure 2.1 The Recognition-Primed Decision model: (a) simple match, (b) diagnosis of the situation, and (c) evaluation of the course of action [14].

then evaluated independently, that is, not in comparison with other alternatives, as is the case with traditional decision models. Thus, the decision-maker mentally visualizes how the situation could potentially evolve if a particular action were implemented.

Following the RPD model, naturalistic decision-making can be achieved at three different levels: simple match [Figure 2.1(a)], diagnosis of the situation [Figure 2.1(b)], and evaluation of the course of action [Figure 2.1(c)]. The result of the decision-making process for each of these levels is the implementation of the action. However, each level includes various steps, allowing for adaptation to the complexity and familiarity of the situation.

The first level (i.e., simple match) is activated when the current situation is simple and straightforward; that is, the crucial elements of the situation, the objectives, and the typical course of action to implement are easily recognized and identified [14, 15].

The second level (i.e., diagnosis) is activated by the presence of uncertainty concerning the situation. As diagnosis is an attempt to establish a relationship between an event and causal factors, this process allows for the decision-maker to define the situation and to find an explanation for it. Because the actions chosen depend on the evaluation of the situation, diagnosis appears to be particularly important in situations involving uncertainty, like command-and-control settings [8, 14].

The third level (i.e., evaluate course of action) is required for more complex situations and requires the mental simulation of the envisaged course of action to evaluate potential difficulties and possible solutions and, consequently, to determine if this action must be implemented or if further evaluation is required to identify a new course of action [14].

According to Klein [15], decision-makers use different strategies in relation to the level involved in the decision-making process (level 2 or 3). Hence, for the diagnosis level (i.e., level 2), strategies are grouped around a principal objective, that is, sizing up the situation and identifying singularities. The strategy most frequently used is feature matching that consists of the comparison between the characteristics of the current situation and different hypotheses explaining the nature of the situation, that is, the hypotheses resulting from the identification process. Strategies that are related to the third level of the RPD model, those implicated in the selection of a course of action, require a deeper elaboration from the decision-maker because of the complexity of the situation. Thus, strategies of the evaluate course of action level (i.e., level 3) are based on the mental simulation of the envisaged course of action, first, to determine how it will be implemented and executed and, second, to identify potential difficulties and possible solutions.

2.3.2 Image Theory

According to the image theory, the knowledge on which a decision-maker bases his or her decision can be divided into three categories, or, more precisely, three images. Beach [16] uses the term *image* to refer to the vision developed by a decision-maker concerning the course of action to implement.

The first image, called the *value image*, is composed of the decision-maker's principles, that is, his or her values, ethical rules, and morals. These principles set up a reference frame within which the chosen course of action and behavior of the concerned individual must be situated. The second image, called the *trajectory image*, represents the expectations and objectives on which the adopted behavior is based. The objectives influencing the decision-making process are those associated with one's principles for one part, and those linked with the current situation for the other part. Finally, the third image is the *strategic image*, which is composed of the plans corresponding to each objective aimed at in the trajectory image. Each plan contains a tactical aspect (i.e., the behavior to adopt in relation to environmental characteristics and constraints) and a prospective aspect (i.e., a prediction of the consequences of the action implemented). Decision-makers must organize the various plans in order to prevent them from interfering with each other in the achievement of the objectives.

According to Beach's model, when a decision-maker is faced with a particular situation, he or she uses contextual cues to search and recall from memory relevant elements regarding the three images that will constitute the decision frame. This recognition process allows the decision-maker not only to determine important components but to obtain information about pursued aims and adopted plans in previous situations that appear similar.

Image theory proposes the existence of two types of decisions: (1) adoption decisions, and (2) progress decisions. Adoption decisions concern the adoption of new objectives or plans regarding the current situation, following the identification of elements through the recognition process. This type of decision can be achieved in two steps, first by determining the different available alternatives, and second by selecting the best option (if a choice is possible). The second type of decision (i.e., progress decisions) refers to the evolution of the plan with regard to the achievement of the objectives. This evolution is based on the action's anticipated consequences and will vary in relation to the progression of the situation.

Furthermore, according to Beach, image theory also includes two decisional mechanisms. The first, the compatibility test, includes in the comparison of each option with the decision frame defined from the three images. This mechanism thus evaluates the options in terms of their quality, and the option is rejected if negative aspects (violations) exceed a particular level. The second mechanism, the profitability test, corresponds to the quantity of consequences associated with the action and refers to the inventory of strategies within which the decision-maker can choose a specific strategy regarding the adopted alternative. This mechanism is based on the postulate that each person possesses an index of strategies and that the selected strategy will depend on three factors:

1. Characteristics of the choice (e.g., uncertainty, complexity, instability);
2. Environment (time and resources available, irreversibility of the choice, possibility of failure);
3. The decision-maker (e.g., knowledge, strategies, expertise, motivation).

Christensen-Szalanski [17] has shown that the profitability test can be described in terms of a SEUT model that includes the subjective utility of correct and incorrect choices. Profitability is then represented as the difference between the expected benefit and the expected cost.

2.3.3 Scenario Model

This model is grounded in the idea that decision-making requires the elaboration of predictions about the consequences of particular events. The use of scenarios permits this kind of forecast development. The scenario model implies four principal steps. First, using as a reference point the objectives and frame within which the decision must be reached, an individual searches his or her memory for relevant information to build if-then propositions (causal relations). Second, these propositions are used to elaborate a cognitive network of causal relations, including both relations that are known (through the information recalled from memory) and those inferred. Third, the decision-maker applies a possible value to the "if" part of the proposition, building as many different scenarios. Finally, the decision-maker determines the logical consequences of each scenario, based on his or her model, to achieve predictions [16].

2.3.4 Argument-Driven Models

Argumentative-type models postulate that the core of decision-making processes is the establishment of arguments for and against a given possible course of action. Thus, according to Lipshitz's [18] argument-driven action (ADA), the decision-maker begins his or her evaluation of the situation based on his or her knowledge and experience. Two basic mechanisms operate in ADA: matching and reassessment.

Matching takes the form of a sequential selection of an action on the basis of an assessment of a situation. It does not require a complete comparison of all alternatives or an explicit consideration of the consequences of the actions. It recognizes that selection can result from a form of pure recognition without any reference to the consequences of the action. An action is selected based on its compatibility with some value or rule of conduct. The selection is supported by an argument relating the action to the "why" based on which it has been chosen.

Reassessment is a form of reevaluation of actions to which one is at least committed. It often takes the form of criticizing beliefs, arguments, and even selected action, which can lead to a modification of the selected course of action. This is viewed as a basis for learning from errors. After this, the envisaged decision is reexamined in light of arguments previously identified that can be applied to the decision, with the individual being able to modify his or her decision.

Argumentative models propose that the arguments considered are not solely selected with regard to the decision to be made but also in reference to the eventuality of the obligation to justify the decision to a particular person or authority.

In ADA, uncertainty concerns either the nature of the situation or the required action. Uncertainty can affect decisions by complicating the matching mechanism. In that case, it will be difficult to provide a valid argument since the situation on which it is based will be fuzzy or incomplete. ADA proposes that uncertainty will be likely to interrupt the ongoing action and provoke reassessment. It will induce a shift from automatic decisions to reflective selection. Consequently, the uncertainty factor is adequately represented in this model. However, the argument process may require considerable time, making its application in time-stressed situations difficult.

2.4 Other Related Cognitive Models

As they provide some additional relevant insight into the decision-making process, this section briefly reviews two cognitive models: (1) the skill-rule-knowledge (SRK) model, and (2) the integrated model of real-world decision-making.

2.4.1 The Skill-Rule-Knowledge Model

The SRK model, elaborated by Rasmussen [19–21], describes three distinctive levels of cognitive control that a person may use to perform a decision task. The level on which a person operates is a function of the complexity of the task and his or her experience and knowledge relevant to the particular situation. Furthermore, the three levels may be used to characterize different degrees of experience and expertise. A novice will process at the knowledge-based level, whereas intermediate learners will possess some rules gained from experience, as well as a qualitatively different knowledge, and will then be able to operate primarily at the rule-based level. An expert will be apt to perform at the skill-based level and, depending of the particularities of the task, to switch between levels (e.g., if the situation is new, he or she may switch to the knowledge-based level). Figure 2.2 presents the three levels of cognitive control: skill-based, rule-based, and knowledge-based behaviors. The information processed at each of these levels enters the system through the attentional process, sensory input in Figure 2.2.

People operate at the skill-based level when they are greatly experienced with the task; that is, they act in an automatic and subconscious manner toward raw perceptual elements. Performance at this level of cognitive control implies stimulus-response associations generated at a neurological level, such that attentional resources are minimally required. The behavior is automatic, and one does not have to process and analyze information or evaluate possible courses of action to produce a response. Errors occurring at the skill-based levels are generally due either to misdirected attention or to paying attention to the task (thus, discontinuing the automatic process) [1].

If a person is familiar with the task but does not have much experience, he or she will operate at the rule-based level. This level implies the processing of information in terms of recognition of certain meanings and signs. These signs initiate if-then rules acquired from past experience, linking a particular sign with a specific action. Errors at this level come from misclassification of cues, resulting in the application of the wrong rule [1].

The third level (i.e., the knowledge-based level) is used by people when the situation is entirely new, so they do not have any rule collected from past experience to apply. This level refers to analytical processing and entails the use of conceptual

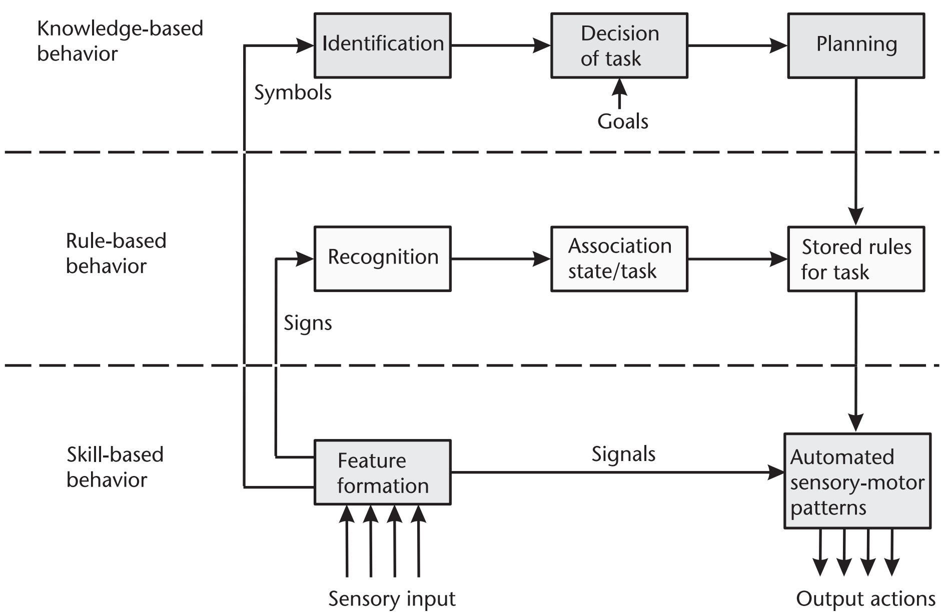


Figure 2.2 Rasmussen's SRK model of cognitive control.

information. A person will begin by assigning signification to information. Then, he or she will integrate all these meanings in an identification frame. Finally, these cues will be processed in light of one's goals in working memory, and mental models will often be used to perform simulations and assess action plans [1]. Errors at the knowledge-based level originate from factors associated with analytical processing, such as limited cognitive resources (e.g., attention, working memory), biases, heuristics, and so forth [22].

In order to enhance the description provided in this book, another view of the Rasmussen's SRK framework is presented in Figure 2.3.

The SRK framework is compatible with the notions of bottom-up and top-down processing. Bottom-up processing stresses the importance of the stimulus in the environment. Data arrive from the sensory receptors and directly influence the perception of the information. Top-down processing stresses the importance of a person's knowledge and concepts in the perception process. Human knowledge

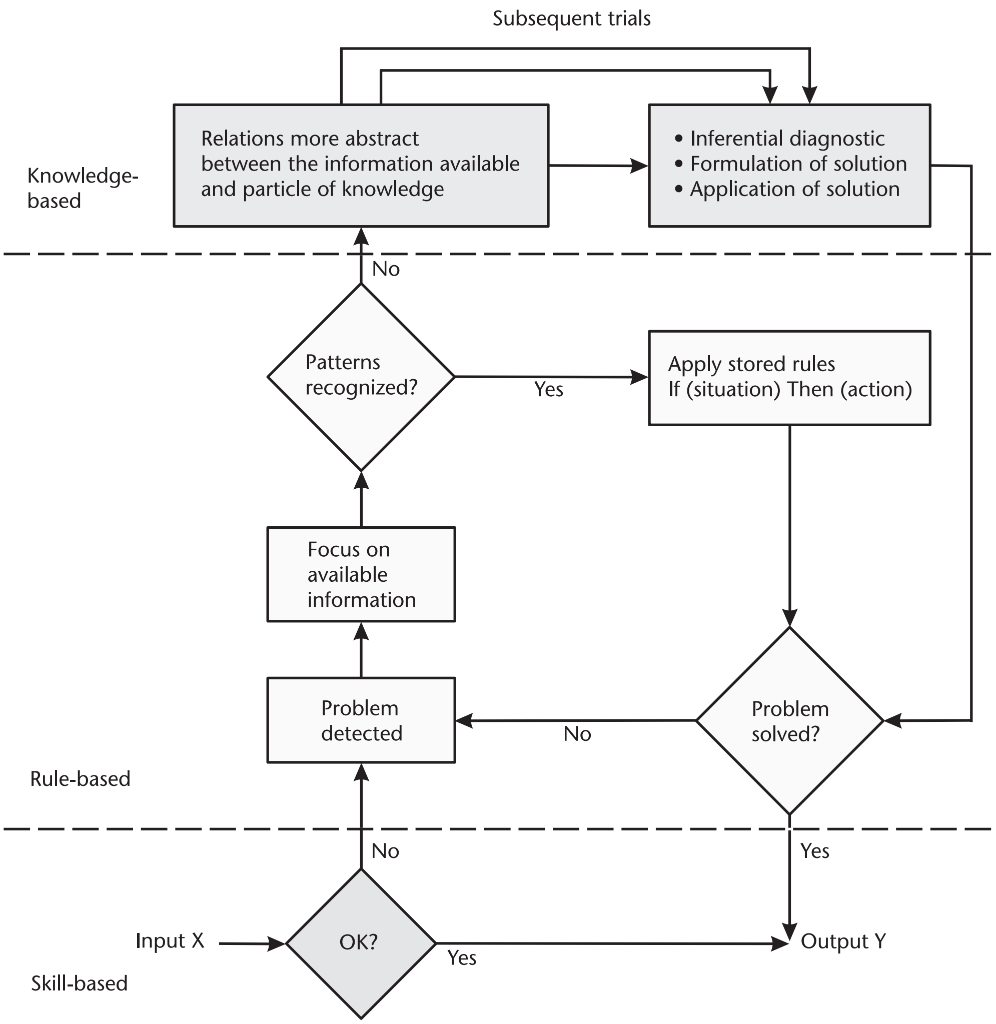


Figure 2.3 Rasmussen's SRK framework.

about how the world is organized helps the individual to perceive and understand the environment. Even if these two approaches to processing are opposite in nature, they are not incompatible. In fact, it is probable that in any perceptual process of the environment, both are involved. Since top-down processing rests on the person's concepts and knowledge, this approach is compatible with the Rasmussen's theory of human performance. This processing approach is also related to training and practice. Top-down processing happens if concepts and knowledge have been previously stored in the long-term memory.

2.4.2 Integrated Model of Real-World Decision-Making

The integrated model combines various views of naturalistic decision-making into a generic information-processing model [23]. Based on Rasmussen's SRK model [19–21], the cognitive control levels are expanded into an information-processing model that reconciles several processes observed and postulated by others. Figure 2.4 illustrates the processing of information, which enters the system through attentional resources (perceive cues in Figure 2.4). It entails three levels of processing

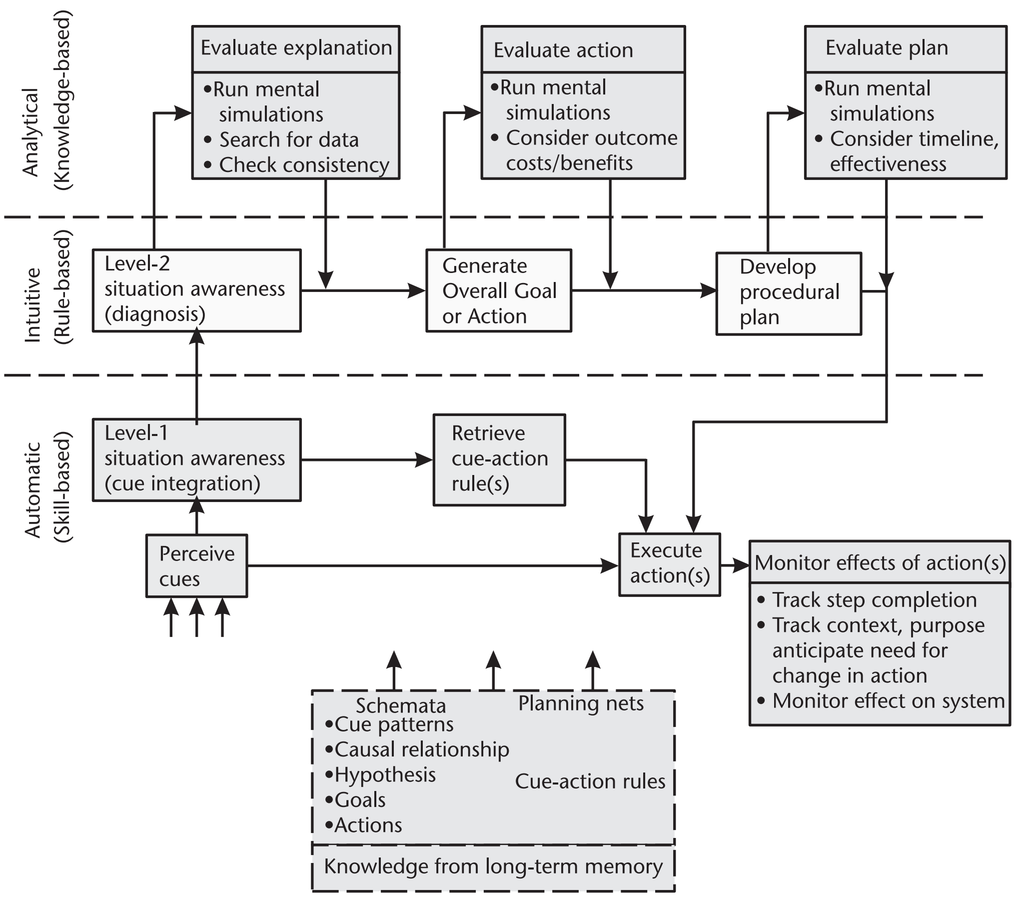


Figure 2.4 A generic information-processing model of naturalistic decision-making. (*After:* [1].)

similar to those of the SRK model: an automatic skill-based level, an intuitive rule-based level, and an analytical knowledge-based level [1].

The automatic skill-based level depends on the environmental cues sensed and is influenced by selective attention. There are no other demands placed on cognitive resources at this level. Strictly speaking, it is not really a decision-making process since the information automatically activates a course of action, often without the person's even knowing consciously which element exactly generated the response [1].

Intuitive rule-based processing, corresponding to level 1 of Endsley's model of situation awareness, implies a greater cognitive effort because the decision-maker must heed a variety of information. These cues initiate rules stored in long-term memory about the adequate association between a specific cue and a particular course of action to be executed, in light of one's goals. The response results from these relations retrieved from memory; hence, a decision-maker may not be able to explain his or her decision because reasoning, per se, is absent [24].

In a case where rule-based processing does not allow for an adequate solution, or when there is uncertainty and no time pressure, a person may switch to the upper analytical level (i.e., knowledge-based processing) and utilize the more evaluative processes present at the top of Figure 2.4. This level begins with level 2 of situation awareness, or the diagnosis in Figure 2.4. The decision-maker may move to the third level at any point in the process (diagnosis, generation of a goal or an action, development of a procedural plan) to achieve a deeper analysis of a situation or a task, mainly relying on mental simulation [25]. Processing can lead to a search for additional information from the environment, the generation of ideas and hypotheses, and so forth. Nevertheless, knowledge-based processing does not necessarily mean that several hypotheses and potential actions will be produced since, under some circumstances, only one hypothesis and action will be considered and executed. However, if the first hypothesis or action fails, a new one may be generated and evaluated until one is finally selected [1].

References

[1] Wickens, C. D., S. E. Gordon, and Y. Liu, *An Introduction to Human Factors Engineering*, Reading, MA: Addison-Wesley Longman, 1998.

[2] Breton, R., and R. Rousseau, *Decision Making in C2: From a Person-Task Perspective*, DREV-TM-2001-219, Valcartier, Canada, 2002.

[3] von Winterfeldt, D., and W. Edwards, *Decision Analysis and Behavioral Research*, Cambridge, U.K.: Cambridge University Press, 1986.

[4] Lichtenstein, S., and P. Slovic, "Reversals of Preference between Bids and Choices in Gambling Decisions," *Journal of Experimental Psychology*, Vol. 89, 1971, pp. 46–55.

[5] Karmarkar, U., "Subjectively Weighted Utility: A Descriptive Extension of the Expected Utility Model," *Organizational Behavior and Human Performance*, Vol. 21, 1978, pp. 61–72.

[6] Kahneman, D., and A. Tversky, "Prospect Theory: An Analysis of Decision Under Risk," *Econometrica*, Vol, 47, 1979, pp. 263–291.

[7] Simon, H. A., "Rational Choice and the Structure of the Environment," *Psychological Review*, Vol. 63, 1956, pp. 129–138.

[8] Plous, S., *The Psychology of Judgment and Decision-Making*, New York: McGraw-Hill, 1993.

[9] Gigerenzer, G., and P. M. Todd, "Fast and Frugal Heuristics: The Adaptive Tool Box," in G. Gigerenzer, P. M. Todd, and the ABC Research Group, (eds.), *Simple Heuristics That Make Us Smart*, New York: Oxford University Press, 1999, pp. 3–35.

[10] Rieskamp, J., and U. Hoffrage, "When Do People Use Simple Heuristics, and How Can We Tell?" in G. Gigerenzer, P. M. Todd, and the ABC Research Group, (eds.), *Simple Heuristics That Make Us Smart,* New York: Oxford University Press, 1999, pp. 141–167.

[11] Zsambok, C., "Naturalistic Decision Making: Where Are We Now?" in C. Zsambok and G. Klein, (eds.), *Naturalistic Decision Making*, Mahwah, NJ: Lawrence Erlbaum, 1997, pp. 3–16.

[12] Klein, G., "Naturalistic Models C3 Decision-Making," in S. Johnson and A. Levis, (eds.), *Science of Command and Control: Coping with Uncertainty*, Fairfax, VA: AFCEA International Press, 1988, pp. 86–92.

[13] Klein, G., R. Calderwood, and D. MacGregor, "Critical Decision Method for Eliciting Knowledge," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 19, 1989, pp. 462–472.

[14] Klein, G., "The Recognition-Primed Decision (RPM) Model: Looking Back, Looking Forward," in C. Zsambok, and G. Klein, (eds.), *Naturalistic Decision Making*, Mahwah, NJ: Lawrence Erlbaum, 1997, pp. 285–292.

[15] Klein, G., *Sources of Power: The Study of Naturalistic Decision Making*, Mahwah, NJ: Lawrence Erlbaum, 1996.

[16] Beach, L. R., *The Psychology of Decision-Making: People in Organizations*, Thousands Oaks, CA: Sage Publications, 1997.

[17] Christensen-Szalanski, J. J., "Problem Solving Strategies: A Selection Mechanism, Some Implications and Some Data," *Organizational Behavior and Human Performance*, Vol. 22, 1978, pp. 307–323.

[18] Lipshitz, R., "Decision Making as Argument-Driven Action," in G. Klein, et al., (eds.), *Decision Making in Action: Models and Methods*, Norwood, NJ: Ablex Publishing Corporation, 1993, pp. 172–181.

[19] Rasmussen, J., "Skills, Rules, Knowledge: Signals, Signs, and Symbols and Other Distinctions in Human Performance Models," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-13, No. 3, 1983, pp. 257–267.

[20] Rasmussen, J., *Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering*, New York: North-Holland Publishing, 1986.

[21] Rasmussen, J., "Deciding and Doing: Decision-Making in Natural Contexts," in G. Klein, et al., (eds.), *Decision-Making in Action: Models and Methods*, Norwood, NJ: Ablex, 1993.

[22] Reason, J., "Framework Models of Human Performance and Error: A Consumer Guide," in L. P. Goodstein, H. B. Andersen, and S. E. Olsen, (eds.), *Tasks, Errors, and Mental Models*, London: Taylor and Francis, 1988.

[23] Gordon, S. E., "An Information-Processing Model of Naturalistic Decision Making," *Presentation at Annual Meeting of the Idaho Psychological Association*, Sun Valley, ID, 1997.

[24] Gordon, S. E., "Implications of Cognitive Theory for Knowledge Acquisition," in R. Hoffman, (ed.), *The Psychology of Expertise: Cognitive Research and Empirical AI*, New York: Springer-Verlag, 1992, pp. 99–120.

[25] Orasanu, J., "Decision-Making in the Cockpit," in E. L. Weiner, B. G. Kanki, and R. L. Helmreich, (eds.), *Cockpit Resource Management*, San Diego, CA: Academic Press, 1993.

CHAPTER 3

Situation Awareness and Analysis Models

Jean Roy, Richard Breton, and Robert Rousseau

3.1 Introduction

The term *situation awareness* (SAW) has emerged as an important concept in dynamic human decision-making. When experts speak about the existence of a general phenomenon called situation awareness, most discussions are reasonably consensual. When the same experts attempt to define this expression in concrete words, the result is debatable.

Actually, many authors have expanded their definition of SAW by developing models of SAW. The complexity of defining the cognitive side of SAW has led a number of authors to develop models that are complex enough to make possible an explicit presentation and definition of this aspect. Among these models, Endsley's model clearly stands out as the reference for most work done on SAW.

The concept of *situation analysis* (SA) synthesizes the main notions put forward by well-established data-fusion and situation-awareness models. The resulting SA model both defines the scope of the situation-analysis process and establishes a comprehensive definitional, conceptual, and functional framework to facilitate the dialogue between researchers, technologists, developers, and users of situation-analysis and command-decision-support systems for military and public-security purposes. Threat analysis is given a particular emphasis as this is a very important aspect of situation analysis. While conducting command-and-control activities, decision-makers eventually have to manage actual or potential threats to some high-value units. Very often, the protection of such units is indeed under the authority and responsibility of these decision-makers.

3.2 General Discussion of Situation Awareness

While the formal scientific study of SAW appeared during the 1980s, that concept has been worded in more or less appropriate terms for centuries. As far as SAW is a natural component of human cognitive organization, the benefit resulting from better situation awareness can be seen as early as prehistoric times [1]. It can participate in all aspects of life. The champion chess player needs particularly acute SAW. The car driver must exploit all cues in a dynamic environment. Situation awareness may result in very simple, everyday decisions, such as leaving home with an umbrella or not.

It can be argued that, historically, on most battlefields, the advantage was with strategists who achieved better awareness of the ongoing situation. SAW was basically concerned with strategic planning issues. While the role of SAW was acknowledged, the scientific knowledge required to develop means to improve SAW in a given situation was not available.

World War I was probably a turning point for the scientific study of SAW. The availability of aircraft considerably increased the capacity to gather intelligence on enemy positions and movements and on the effectiveness of Allied actions. Boelke (in [2]) was one of the first to express the importance of SAW. Not only did critical aspects of the understanding of the battlefield situation depend on pilots' observational abilities, but the air fight situation in itself revealed the need to develop better SAW. Gilson [3] mentions that the advantage of the Red Baron, in World War I, was probably based on his outstanding SAW capabilities. In that period, SAW was not only critical for headquarters decisions; it was also critical for individual soldiers in the field, notably those working with technology.

After World War I, the rapid developments in technology yielded important benefits that far outweighed the increase in mental workload associated with the operation of the new technological tools. The concern about scientific studies of SAW was thus judged marginal up to the mid-1980s, when a 1986 report by the U.S. Air Force, entitled "Intraflight Command, Control, and Communication Symposium Final Report," finally declared SAW "the single most important factor where the mission effectiveness could be improved" [3]. This statement positioned the U.S. Air Force as a first key stakeholder of the development of research on SAW [4]. Soon, this concept was generally adopted in all domains of aviation, notably the commercial airlines. Research on SAW and the application of its findings were especially successful in piloting and air traffic control.

3.2.1 Identifying the Benefits of Situation Awareness

The first important benefit of SAW was the discovery that most errors in air traffic control were the consequence of a failure to maintain appropriate SAW [5, 6]. Hartel et al. [5] found that SAW errors were the main cause of military aviation mishaps. In commercial airlines, Endsley [7] reported that 88% of human errors were related to inadequate SAW. A bad perception of needed information is present in 76% of SAW errors, while a problem with the comprehension of the information perceived was noted in 20% of SAW errors [8]. When a pilot neglects to check the flaps at take off and consequently crashes, the error can hardly be attributed to inadequate training, lack of practice (because that task has been practiced hundred, if not thousands, of times), or scarce cognitive resources. Considering the risk of a deadly error, such a mistake is certainly not the consequence of a gross negligence. Inappropriate SAW has been suggested as a prime explanation for such accidents. An improvement in SAW could lead to a reduction in costly errors. SAW enables the development of new abilities, leading to high proficiency levels in terms of planning, decision-making, and action. Klein [9] gives four reasons why SAW is important:

- SAW appears to be linked to performance.
- Limitations in SAW may result in errors.
- SAW may be related to expertise.
- SAW is the basis of decision-making.

In all cases, the focus is on the extra benefits a hyperproficient agent can derive from taking advantage of a situation. As stated before, the SAW concept and its measurement were initially developed in the context of explaining operator or pilot mistakes that were otherwise hard to understand. A relative consensus emerged about SAW's being a helpful concept, notably to explain errors in complex and dynamic technological environments. It is only recently, inspired by the U.S. Twenty-First Army, that research on SAW has been seen as a major contributor to strategic advantages on the battlefield. For the infantry, the focus of SAW is not so much error reduction but obtaining strategic advantage in the field.

3.2.2 Situation Awareness as a General Concept in Multiple Domains

Although most recent work on SAW has been linked to aviation [1], the interest in SAW rapidly expanded into several other fields of activity where humans have to monitor high-tech devices in complex, dynamic, and constantly changing environments. The question was then asked, Is SAW, outside the aviation domain, just a marginal concept?

During the last decade or so, one of the aims of most research was to include SAW in the list of human factors as an independent entity compared to perception, attention, working memory, or mental workload, which are the same whatever the application field is. It was then important to investigate SAW in a large variety of fields. If SAW had no general properties, this would limit its acceptability as a basic human-factors concept and would consequently reduce interest in the search for a general definition and measurement methodology.

The potential generalization of SAW is confirmed by the works of Gaba and Howard [10]. They have described a strong analogy between SAW requirements in aviation and anesthesiology. Both fields imply dynamism, complexity, a high information load, a variable workload, and risk. The existence of realistic simulators where surgical problems are replicated showed that SAW could be investigated with the same methods in both fields [11, 12].

The results in the field of nuclear power plant process control were less fruitful. Early work by Woods, O'Brien, and Hanes [13] suggested that SAW could be applied with benefits to this specific field. However, their approach is at odds with the key researches that are now considered as the base of the contemporary conceptualization of SAW. A recent attempt to study team SAW during the normal operations of a nuclear power plant failed because all operations were perfectly executed [14]. The investigated situation offered no opportunity to make mistakes, thus few opportunities to observe fluctuations in SAW.

Gugerty and Tirre [15] investigated SAW in car driving with approximately the same techniques as those used in aviation. One can argue that driving a car might not be fundamentally different from piloting. Their results did show that the SAW methodology could be applied in the same way to everyday tasks as it has been in tasks requiring maximal human performance. Jenner et al. [16] linked a lack of SAW to a variety of accidents investigated by the National Transportation Safety Board in the marine, the pipeline, railroad, and aviation industries. According to Molloy [17], surface transportation entered into the same era of automation pitfalls that aviation had to overcome several years ago. Overconfidence in the technology and a consequent loss of SAW is observed in investigated accidents.

Very diverse applications appeared recently. Klein [9] introduced methods to investigate SAW that are similarly applicable to the analysis of errors in both aviation and neonatal diagnosis of an extremely dangerous systemic infection in newborns. The work on team SAW (e.g., [18]) yielded another important cue about the generalization of this concept from individuals to groups. Although Endsley et al. [19] developed two different models for individual and team SAW, both issue from the same conceptual framework. Trafton [20] also used knowledge about SAW to study how expert navy forecasters build their weather forecasts. Finally, Endsley [1] added to this list the study of advanced manufacturing systems, education, maintenance, and operator interfaces.

Thus, SAW appears to be a general concept that can be of interest in a large number of settings. The pervasive use of computer devices for the control of various processes or machines has produced an increase in the overall information available to an operator and in the amount of information provided by various interface systems. Compared to nontechnological situations, where SAW is mostly about the perceived natural environment, technological devices also require SAW to take into account the state and functions of the device itself. The overall increase in the complexity of information sources in current technological environments makes an exhaustive SAW much more difficult. The overflow of information generated by computer systems has strengthened the need for a more systematic study of SAW. More data does not mean more information [1]. On one hand, these computer systems often aim at maximal exploitation of human cognitive resources in information processing. On the other hand, if they are badly designed, they may contribute to information overload by providing too much data and not enough information, consequently creating a situation where too few cognitive resources are available for information comprehension and interpretation essential to higher-level SAW.

There is another potential problem with the use of computer systems. The automation of data acquisition, analysis, and assisted decision-making may sometimes result into a loss of SAW. For instance, the automation of the data-acquisition process may reduce a person's awareness of the actual state of the world. To address this in relation to automation, the relationship between automation and SAW was the main question discussed during the third meeting on SAW held in Savannah, Georgia, in October 2000.

Although SAW has been a constant topic of discussion at every annual meeting of the Human Factors and Ergonomics Society since 1993, most of the now available works had already been published by 1995. The following years were a time of implementation rather than fundamental research, notably at the commercial level. At times, device designers claimed that their new products sustained better SAW. This assumption was optimistic but often lacked sound scientific grounding. In fact, the problems with the SAW definition and its measurement are still preventing the successful implementation of devices and methodologies for SAW support at the scale that was then expected.

3.3 Defining Situation Awareness

When experts speak about the existence of a general phenomenon called *situation awareness*, most discussions are reasonably consensual. In practice, a long list of concrete examples exists to persuade people that SAW has its own reality and its own importance. Endsley et al. [19] adopted this approach with success. When the same experts attempt to define this expression in concrete words, the result is debatable. At times, the definition is considered imprecise, impossible to measure, circular, or too bound to the characteristics of a particular situation. In other cases, it is too general and cannot be differentiated from other related concepts.

As Pew [21] pointed out, the term *situation awareness* shares a common history with several psychological concepts, such as intelligence, vigilance, attention, fatigue, stress, compatibility, or workload. For decades, all of these terms were poorly defined. However, each became important because it attracted attention to critical processes or mental states that were previously unknown. Ultimately, these terms changed how we study human-factors problems, and they brought new benefits.

As a result of the persistent, unsuccessful efforts to define SAW, Sarter and Woods [22, p. 16] proposed that "the term *situation awareness* should be viewed just as a label for a variety of cognitive processing activities that are critical to dynamic, event-driven, and multitasks fields of practice." Such a point of view enables applied work on SAW to proceed, but in the long run can be detrimental to the field and foremost to the development of general SAW measurement tools. The acceptance of a precise and universal definition of SAW would bring considerable advantages to the field. However, unrealistic endeavors can be counterproductive and eventually ruin this new field of human factors by delaying applications that can benefit operators in various domains. Thus, in order to assess the SAW measurement tools currently available, it is important to address the issue of the definition of the object these tools aim at measuring, that is *situation awareness*.

3.3.1 The Notion of Awareness

Browsing through some standard, everyday dictionary (e.g., Merriam-Webster), we find that *awareness* has to do with having knowledge of something; this is illustrated in Figure 3.1.

In addition to the cognition facet, awareness is also linked with the notions of perception and understanding or comprehension. While perception is defined as becoming aware of something through the senses, comprehension and understanding are both defined as knowledge gained by grasping, with the intellect, the nature, significance, or meaning of something.

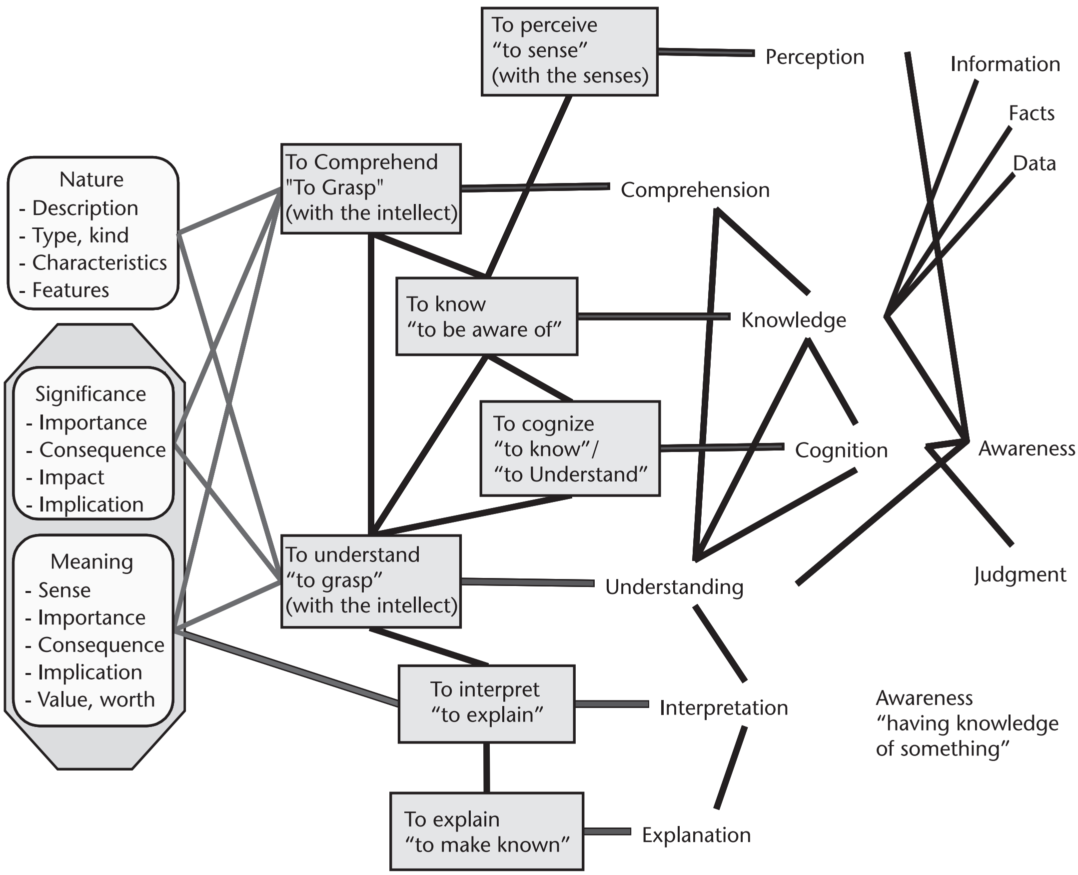


Figure 3.1 The notion of awareness.

3.3.2 The Notion of Situation

Once again browsing through the dictionary, we define a *situation* as a specific combination of circumstances (i.e., conditions, facts, or states of affairs) at a certain moment. We then say that a situation is a combination of *situation elements*. Figure 3.2 is an attempt to list some basic situation elements relevant to most military and public-security operations and to illustrate, at a very high level, some of the relationships between these elements. Clearly, this list in Figure 3.2 is far from exhaustive, and a multitude of aspects must be considered in typical situations of interest.

The partitioning, in Figure 3.2, of the situation elements into physical, metaphysical, intentional, and social categories is inspired by Nowak [23] and his effort to develop an appropriate conceptualization to support information fusion.

The main two basic situation elements are potentially the *entity* and the *event*. An entity is an existing thing (as contrasted with its attributes), that is, something that has independent, separate, self-contained, and/or distinct existence and objective or conceptual reality. An *event* is something that happens (especially a noteworthy happening). Hence, entities exist, while events occur.

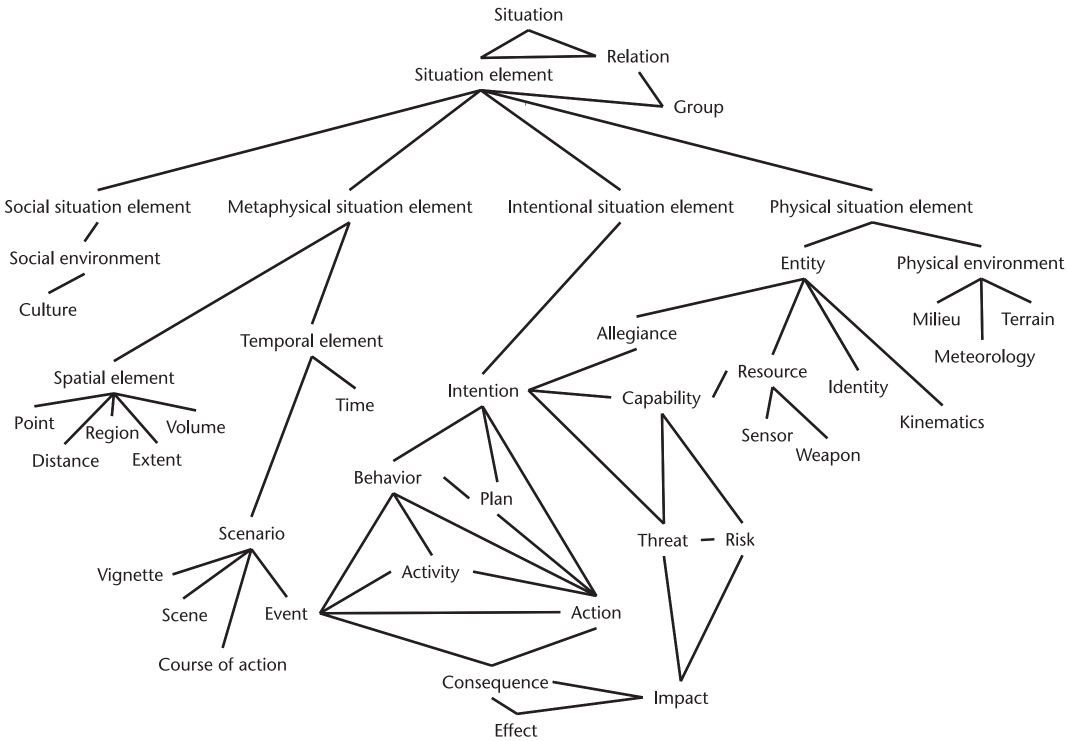


Figure 3.2 Some basic situation elements.

Physical and human resources could potentially be used to act upon the situation and provide capabilities to the entities. The term *activity* refers to the notions of action, movement, and motion. It is appropriate when something has the quality or state of being active, that is, when something is characterized by action or expresses action, as distinct from merely existing or having a state. A *scenario* is defined as a sequence of events. The term *scene* is defined as a single situation in a play. As one may think of a global situation as comprising a set of local situations, the term *scene* could be used to refer to a local situation, that is, a single situation in this set.

A *group* represents a number of situation elements assembled together or having some unifying relationship, for instance, an assemblage of entities or events regarded as a unit. Finally, the *person* who needs to acquire and maintain SAW, the *goals* of this person, and the technological *systems* and the other *persons* supporting the acquisition and maintenance of SAW are additional situation elements not shown on Figure 3.2 that should typically be included to provide a description of a situation. Once again, the discussion here is far from being exhaustive.

The dynamic relationships between the situation elements are also highly important. A *relation* can be defined as an aspect or quality that connects two or more things or parts as being or belonging or working together or as being of the same kind. Note that relations between situation elements can themselves be considered elements of situation (e.g., relations between the sensors, weapons, and identity, as shown in Figure 3.2).

3.3.3 A General Framework for a SAW Definition

Several definitions of SAW are disseminated here and there through a wide range of papers. Jeannot [24] presents a table of definitions from Dominguez [25]. It is an interesting starting point for a repertory of SA definitions.

In order to extract the basis for a general definition of SAW, one approach is to start with the essential elements involved in SAW. This can be done simply with the following illustration presenting the two basic elements in SAW: the *situation* and the *person*.

In Figure 3.3, the *situation* can be defined in terms of events, entities, systems, other persons, and so forth, and their mutual interactions. The *person* can be defined according to the cognitive processes involved in SAW, or simply by a mental or internal state representing the *situation*.

The simple representation illustrated in Figure 3.3 is not a model in itself. It is a simple schema of the general elements, from both the situation and the person sides, which should appear in a global definition of SAW.

3.3.3.1 The Person Side of SAW

From the person side, a given definition may be *process* oriented, focusing on the link between the situation and the cognitive processes generating SAW. This is well illustrated in Dominguez [25]. Her definition presents a set of four processes, or functions, on which SAW depends: *information extraction, information integration, mental picture formation*, and *projection and anticipation*.

Other definitions are *state* oriented, focusing on the link between the situation and an internal representation of the elements present in the situation. Adam [26] provides a clear example of a state-oriented definition that defines SAW as "*knowing* what is going on so I can figure out what to do." State-oriented definitions limit the description of the processes involved in SAW. In fact, that distinction is on the same line as the more basic opposition between the concept of *direct perception* and the *indirect perception* concerning the theoretical status of perception as a basic mental process. Derived from the work of Gibson [27], direct perception is based on a number of principles, two of them being of interest for SAW:

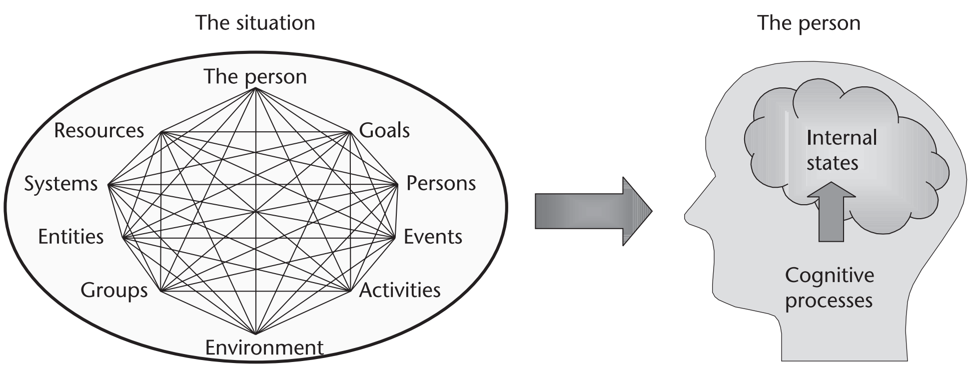


Figure 3.3 A simple illustration of the elements involved in SAW.

1. All the information necessary for perception is contained in the environment.
2. Perception is immediate and spontaneous.

It follows that in order to understand perception, the priority must be on understanding the environment. There is no need to develop theories of perception based on inferred mental mechanisms of information processing from which perception would result. On the other hand, an information-processing approach considers that a mental representation of the world is based on processing with specific functions [28, 29]. That approach requires an explicit description of the processes involved in providing humans with cognition.

This simple distinction between "process" and "state" is of considerable importance. One of the major difficulties in working with SAW is avoiding confusing SAW knowledge with underlying cognitive processes, such as perception, memory, attention, categorization, or decision-making. This difficulty is particularly acute when SAW has to be measured. In agreement with Adams, Tenney, and Pew [30], Endsley [31] limited the term *situation awareness* to the achieved knowledge (state) about a situation. She proposed the expression *situation assessment* to designate the cognitive processes that produce the knowledge (state).

Defining SAW as a state of relevant knowledge of which an operator is aware is not without problems. Smith and Hancock [32] suggest defining SAW with regard to an external goal to avoid the dead methodological introspection issue. *Introspection* is a process by which people come to be attentively conscious of their current mental state. This focused consciousness on one's concurrent mental state is distinct from the relatively casual, fleeting, diffuse way we are ordinarily conscious of many of our mental states [33]. Methods based on introspection rely on verbal reports about one's mental states. It is now well accepted that introspection is the result of these mental states and not a mere reflection of their current status. Furthermore, introspection often fails to report on mental states that operate on a more automatic cognitive level, like implicit memory or skilled performance on which expert performance is often based. Hence, SAW cannot simply be equated with any verbal report of the content of consciousness about a situation. According to Smith and Hancock, "To equate SAW with momentary knowledge and mental models is to run the risk of allowing SAW to degenerate rapidly into whatever is inside your [skilled] head." These authors also state that "to comprehend SAW without a viable understanding of the interaction between agents and their task environment would be virtually impossible" [32, p. 140]. Actually, such comments stress the importance of considering SAW as a specific mental representation.

In the context of the development of a SAW definition, one is then left with a double problem. On one hand, if SAW is a state, it is essential to give a precise definition of the knowledge that defines the state. There should be a certain mapping between a situation schema and a knowledge schema. If one is to improve SAW, the elements of the situation critical for SAW should be specified, and the SAW content definition should follow from these elements. On the other hand, if SAW depends on a set of processes that are not an intrinsic part of SAW as a state, but on which SAW depends, it becomes important to specify which processes are essential to SAW. SAW improvement, for instance, will depend upon changes in the operation of these processes.

3.3.3.2 The Situation Side of SAW

From the situation side, SAW definitions can be classified as either *general* or *specific*. On one hand, a specific definition describes the situation in terms of the objects, actions, and variables related to the task being performed. They are detailed and precise. Prince and Salas [34] propose a specific definition stressing very specific elements. They define SAW as "the ability to maintain an accurate perception of the surrounding environment, both internal and external to the aircraft, as well as to identify problems and/or potential problems, recognize a need for action, note deviations in the mission, and maintain awareness of tasks performed." On the other hand, a general definition will refer to the situation in abstract, nonspecific terms. A definitely general definition is presented by Gibson and Garrett [35], who state that SAW is "the pilot's overall appreciation of his current world."

As we have argued before, it is important to distinguish both the general and the specific definitions. Perhaps a major part of the definition problem would be solved if the definitions were tightly bound to the sole situations or environments in which the studied process or mental state had a real, non-negligible impact. That is to say, SAW would only be considered in a certain set of situation conditions, when specific definitions could be proposed.

Adams, Tenney, and Pew [30] have pointed out that SAW is not always important. SAW is often needed in times of crisis. What makes a situation a crisis if not the characteristics of the situation itself? On one hand, restricting SAW to crisis situations would leave us with an efficient, specific operational definition; on the other hand, this definition would still be lacking in terms of general properties. Thus, if the specific definition is adopted as a solution for the SAW definition problem, one is then left with the problem of developing a useful definition.

Of course, if a new specific definition were required for every situation, efforts to provide a general definition of SAW would be senseless. Gaba and Howard [10] have written that SAW is as critical in anesthesiology as it is in aviation since both environments include dynamism, complexity, a high information load, a variable workload, and risk. Common environmental characteristics should be looked for. One way to address this problem is to consider the situation part of the definition as being based on the generic properties of one situation within a class of situations. The situation elements, while lacking in some detail, would remain the same for all situations belonging to a given class.

Pew [21, p. 34] defines a situation as "a set of environmental conditions and system states with which the participant is interacting that can be characterized uniquely by a set of information, knowledge, and response options." Pew then proposes that SAW should integrate, when applicable, five aspects of the situation:

1. The surrounding environment (spatial awareness);
2. The mission's goals (e.g., "to keep current with respect to the phase of the mission and the currently active goals that are to be satisfied");
3. The system (especially with complex automated systems);
4. The available physical and human resources;
5. The crew (e.g., "Each crew member must know the current activities of other crew members so that their availability for critical tasks is known").

The role of the general definition is then to propose constraints as far as what can be included in a specific definition of SAW. In that view, *situation* takes on a very large meaning. It includes task and mission features, as well as the other human agents in the significant environment. Propositions like Pew's provide the basis for a better understanding of what is meant by *situation* in SAW.

3.3.4 An Analysis of SAW Definitions

To summarize the preceding discussions, one may claim that a definition of SAW can be either *process* or *state* oriented from a *person*-side perspective. Also, such a definition can be seen as *general* or *specific* from the *situation*-side perspective.

In order to better understand the variety of efforts deployed at defining SAW, 27 definitions have been analyzed by Breton and Rousseau through simply classifying them as process oriented or state oriented, and as being general or specific [36]. Among these definitions, the three-level definition of SAW proposed by Endsley [37] has been adopted by a majority of researchers. According to Endsley, SAW can be defined as "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future."

At the end Breton and Rousseau's analysis [36], the question remains as to whether a definition of SAW should be limited to content or if it should include the processes or functions linked to the awareness of the situation. Should SAW include what some authors refer to as situation assessment? It is not possible to provide an answer to that question from the strict analysis of SAW definitions. However, many authors have expanded their definition of SAW by developing models of SAW. In fact, the complexity of defining the cognitive side of SAW has led a number of authors to develop models of SAW that are complex enough to make possible an explicit presentation and definition of this aspect. Such models are discussed in the following sections.

3.4 Endsley's Model of Situation Awareness

A systematic analysis of SAW models is provided in Endsley et al. [19]. They present a list of eight models, all but one of which has been developed since the mid-1990s. They are presented as "models of how people achieve SAW in complex domains" [19, p. 34]. So, the models are not necessarily formal SAW models but, most often, are descriptions of the status of SAW in a general model of cognitive processing, taking into account noncognitive factors affecting the development of SAW.

Among these models, Endsley's model [1, 31, 37] clearly stands out as the reference for most work done on SAW. A number of other models focus on a specific aspect of SAW but remain within the constraints of Endsley's model. For instance, Maggart and Hubal [38] describe SAW in the context of infantry operations. They explicitly rely on Endsley's [31] model as a basis for describing the specific elements of SAW in that context, while focusing on the environment, physical or organizational, in which infantrymen operate. Similarly, Endsley and

Jones [39] and Salas et al. [40] address the important issue of team or shared SAW from the point of view of the Endsley's model. Likewise, McGuinness and Foy [41] base the development of a SAW measure on Endsley's model, while proposing some modifications to the original. Actually, the current status of SAW as a scientific concept in the field of human factors owes very much to the sustained efforts by Endsley and her collaborators.

Endsley et al. describe an extended version of Endsley's model adapted to infantry operations [19]. It is, by far, the most extensive SAW model currently available. The model has two main parts: the core SAW model and the various sets of factors affecting SAW. The first part we call the core SAW model since it represents the processes directly responsible for SAW. The core model follows Endsley's proposition that SAW is a three-level mental representation: perception, comprehension, and projection. The second, and much more elaborate, part describes in detail the various factors affecting SAW grouped into four broad classes: external world, task and environmental factors, individual factors, and military domain factors. These factors include contributions from all components of current human information-processing models, like goals, active schemas, past experience, attentional processes, and working memory. While this makes the model all encompassing in terms of factors affecting SAW, it does not push any further the modeling of core SAW. In a way, the model is now very large and not very tractable.

The model of SAW presented in Endsley [31] is thus the basis for much of the current modeling of core SAW; it is illustrated in Figure 3.4.

The three levels of SAW from that model will be briefly described next:

- *Level 1—Perception of the elements in the environment:* This is the first step in achieving SAW. It provides information about the status, attributes, and dynamics of the relevant elements in the environment. The perception of cues is fundamental. Without a basic perception of important information, the odds of forming an incorrect picture of the situation increase dramatically [42]. Level 1 SAW includes the classification of information into understood representations. Long-term memory stores contain knowledge that enables mental representations of the elements. Moreover, perceived elements are a subset of elements present in the environment. The subset is under attentional selection based on task requirements. The elements are structured into meaningful events situated in time and space. These events form an important part of level 1 awareness and make possible a dynamic mental representation sensitive to change. That content is active in working memory, thereby providing a basis for awareness of it.
- *Level 2—Comprehension of the current situation:* Endsley also states that SAW, as a construct, goes beyond mere perception. It also encompasses how people combine, interpret, store, and retain information. Thus, it includes more than perceiving or attending to information; it includes the integration of multiple pieces of information and a determination of their relevance to a person's goals [43]. Level 2 SAW is thus a synthesis of level 1 disjointed elements. It provides an organized picture of the elements with a comprehension of the significance of objects and events. Such a comprehension of a

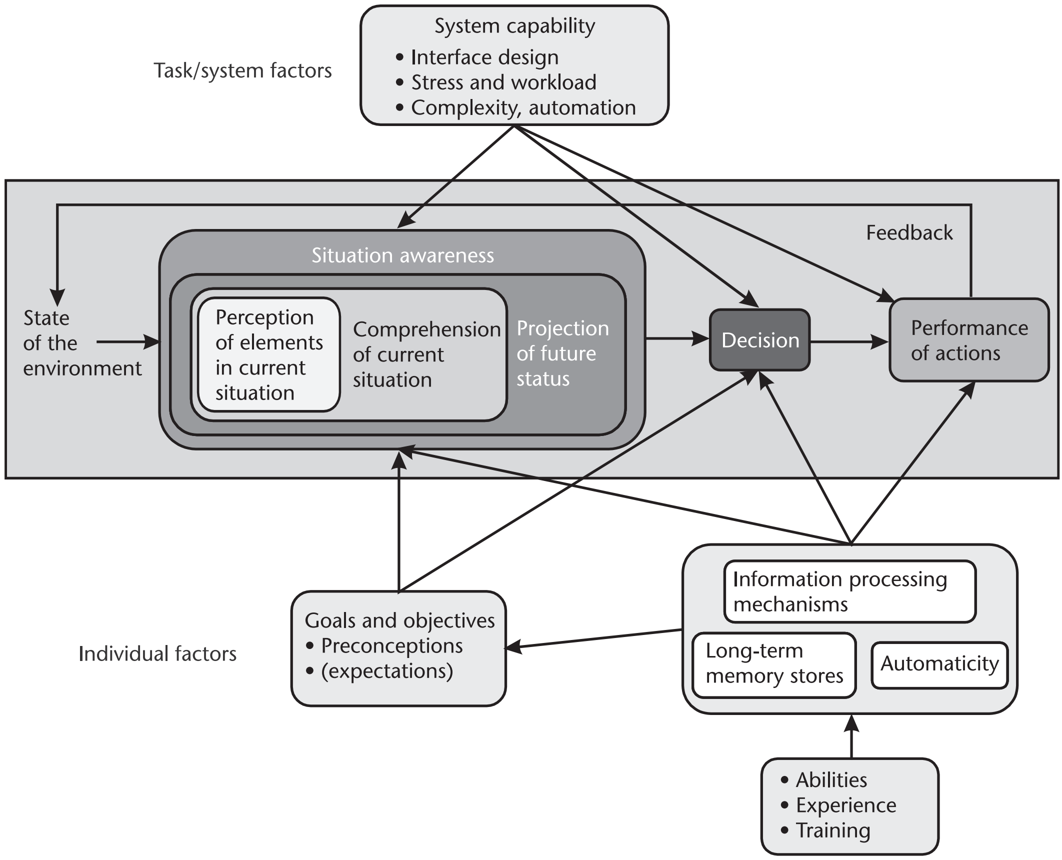


Figure 3.4 A model of situation awareness [31, 42].

situation demands that the problem of meaning be tackled head on. Meaning must be considered both in the sense of subjective interpretation and in the sense of objective significance or importance. A person with situation comprehension has been able to derive operationally relevant meaning and significance from the data perceived [42]. Schemata or mental models stored in long-term memory are the basis for level 2 SAW. Mental models are complex schemata representing a given system. Level 2 SAW is then defined as a situational model depicting the current state of the mental model.

- *Level 3—Projection of future status:* At the highest level of SAW, the ability to forecast future situation events and dynamics marks decision-makers who have the highest level of understanding of the situation. Level 3 SAW is achieved through knowledge of the status and the dynamics of the situation elements and the comprehension of the situation. It enables predictions about the states of the environment in the near future. The mental model provides a means to go from an understood situation to the generation of probable scenarios as to the possible future states of the system. This ability to project from current events and dynamics to anticipate future events (and their implications) allows for timely decision-making.

3.5 Defining and Modeling Situation Analysis

Based on the work of Roy, Paradis, and Allouche [43, 44], this section proposes another concept, *situation analysis* (SA), in an attempt to synthesize the main notions put forward by well-established data-fusion and situation-awareness models. The model of SAW put forward by Endsley was described in Section 3.4, and the models of the Joint Directors of Laboratories' Data Fusion Group (JDL DFG) [45–49] for data fusion are described in Chapter 4. In addition to clarifying the original concepts of these models, this synthesis expands on some of these ideas while achieving a fair depth in the level of detail of the resulting generic description.

Particular care has been devoted to the selection of an appropriate unifying terminology to designate the fundamental elements of SA. The proposed model both defines the scope of the situation-analysis process and establishes a comprehensive definitional, conceptual, and functional framework that could facilitate the dialogue between researchers, technologists, developers, and users of situation-analysis and command-decision-support systems for military and public-security purposes.

Although the elements of SAW may vary widely between domains, its nature and the mechanisms used for achieving it can be described generically, at an implementation-independent level of abstraction. In that sense, no algorithms or techniques to achieve SA are provided here; in line with Lambert [47], we are more interested in knowing what SA is rather than how it is done. The resulting description can thus be useful across multiple application areas.

Situation analysis, as described in this section, is a complex process, requiring deep knowledge of operations, doctrine, equipment characteristics, the effects of terrain and weather on operations and equipment, and a host of other factors, including even such intangibles as a sense of the will of the different actors to fight. It is a goal of this section to provide a foundation for understanding situation analysis.

3.5.1 Situation Awareness and Decision-Making

According to Endsley and Garland [42], there is a strong link between SAW and decision-making (DM) processes. Nevertheless, Endsley presents SAW as a stage separate from decision-making and action in her model. SAW is described as the decision-maker's internal model of the state of the environment; based on that representation, the decision-maker can decide what to do about the situation and carry out any necessary actions. SAW is therefore represented as the main precursor to decision-making. This is illustrated in Figure 3.5, built around Boyd's OODA loop. The first half of Boyd's loop (observe and orient) gathers a number of processes that mainly perceive, interpret, and project the status of the elements included in the environment. These processes yield the situation awareness required to complete the decision-making process. The second half (decide and act) of the OODA loop decides on the best course of action with respect to the mission and supports its implementation given the situation and the available resources.

From the perspective of Figure 3.5, a key factor determining decision quality is SAW. However, good SAW does not necessarily produce good DM. In some

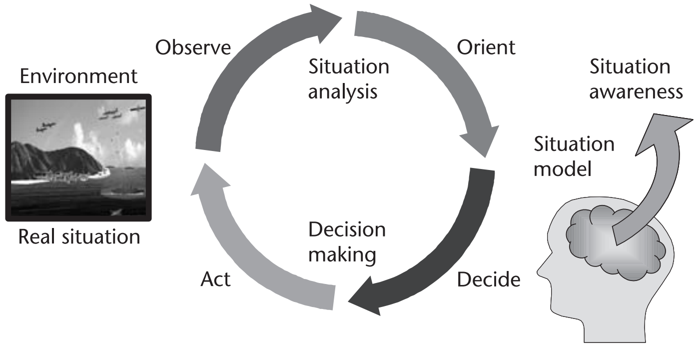


Figure 3.5 Situation analysis and decision-making.

circumstances, the best alternative can be selected without the presence of critical information. In others situations, bad choices can be made even with the availability of all the information defining good SAW. However, one can claim that enhancing SAW improves the probability of selecting the appropriate course of action in most of situations. Consequently, the improvement of the human DM process can be seen as highly related to the enhancement of SAW.

Other DM models are claiming the importance of SAW. Klein [50] raises the importance of a pattern-recognition process in his Recognition-Primed Decision (RPD) model. As discussed in Chapter 2, this model, influenced by the naturalistic decision-making (NDM) trend, suggests that humans are rapidly selecting a satisfying alternative through a pattern-recognition process instead of comparing many plausible alternatives to find the optimal one. This model is appropriate to represent the DM process in situations characterized by factors such as time pressure, stress, high stakes, conflicting goals, and ill-defined problems. In intelligent systems, such as case-based reasoning (CBR) and knowledge-based and rules-based models, the introduction of critical information is essential for the selection of the appropriate alternatives.

3.5.2 Situation-Analysis Definition

In this perspective, we define situation analysis as *a process, the examination of a situation, its elements, and their relations, to provide and maintain a product, that is, a state of situation awareness, for the decision-maker*. As shown in Figure 3.5, the SA process thus encapsulates that part of the overall decision-making cycle concerned with understanding the world. There is a real situation in the environment, and the SA process will create and maintain a mental representation of it, the situation model, in the head of the decision-maker(s).

At the highest level, making a strong parallel with Endsley's work, the SA process can be decomposed into four subprocesses: situation perception, comprehension, projection, and monitoring (note that we are talking about processes here, not states). This is illustrated in Figure 3.6.

Situation perception has to do with the "acquisition" of the situation through data and information collection with various sensors and other sources. Situation comprehension is about further developing one's knowledge of the situation with

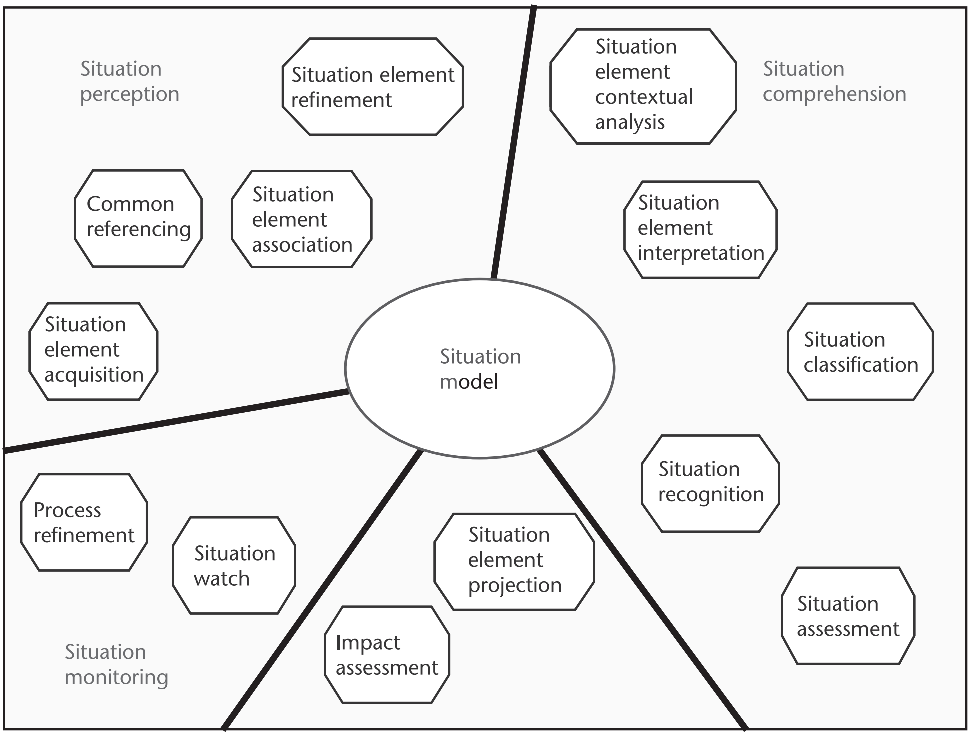


Figure 3.6 High-level view of the situation-analysis process.

respect to both its nature (i.e., the inherent character or basic constitution of the situation) and its significance or meaning (i.e., the importance of the situation). This subprocess must be able to grasp the nature of the situation and to derive operationally relevant meaning and significance from the results of situation perception. Situation projection must produce estimates of future possibilities for situation elements, based on current trends, and of the consequences, impact, or the implications of the situation. Finally, situation monitoring has to do with watching, observing, or checking the evolution of the situation in order ultimately to keep track of, regulate, or control the operation of the SA process.

Figure 3.7 is a much more detailed functional description of the SA process than Figure 3.6. From a data-driven perspective, it entails integrating and interpreting the whole spectrum of source data and information, ranging from radar returns to political factors. The SA process thus encompasses a vast range of activities, from the detailed signal processing associated with target acquisition and tracking to intelligence interpretation. Simply put, the process must provide answers to a great number of questions: What? Who? How many? How big? Where? What structure? When? What is it doing? Why? What's the build up? What could it do? How soon? What is outstanding? What has changed? What is delta from expectations? What is going wrong? The SA process thus consists of numerous dependent and independent subprocesses at multiple levels of abstraction. Every subprocess can itself be further decomposed hierarchically into multiple subprocesses. Clearly, the set of subprocesses included in Figure 3.7 is far from exhaustive; however, it is representative

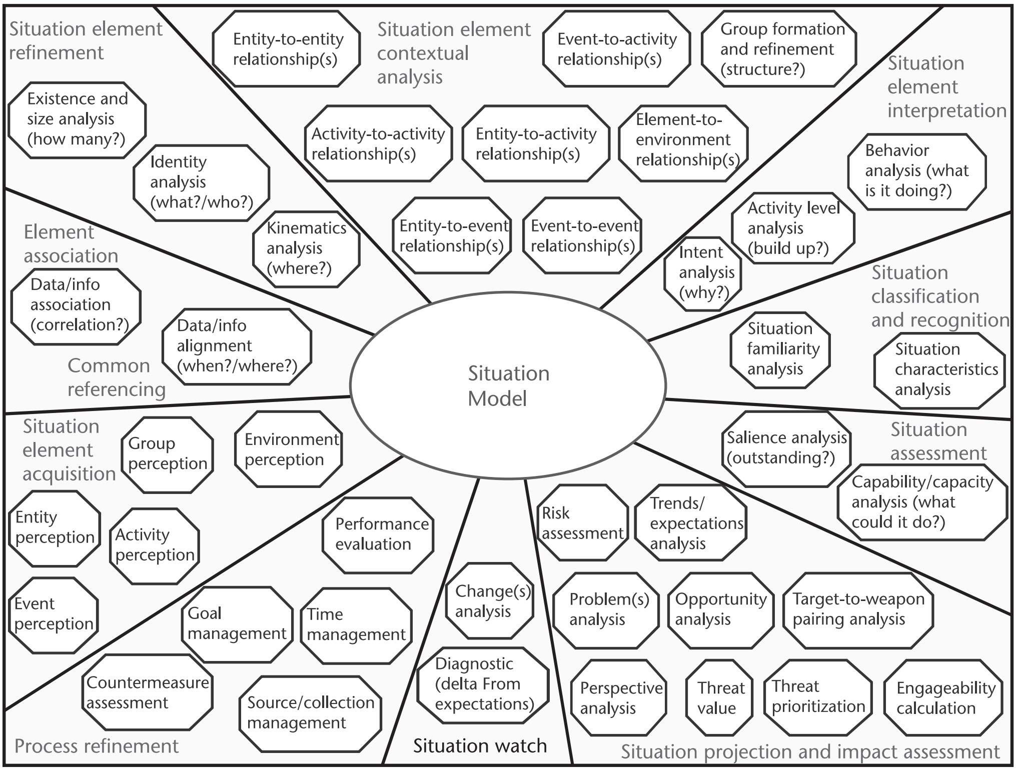


Figure 3.7 Detailed view of the situation-analysis process.

of most of the main subprocesses that are relevant in typical military or public-security operations.

These SA subprocesses must be integrated and interleaved into an overall processing flow. Regarding this issue, one should note that there are no arrows on the diagram in Figure 3.7. Moreover, although SA is clearly a multiple-level-of-abstraction activity, there are no explicit references to the JDL data-fusion "levels" in the model. According to Steinberg, Bowman, and White [45], the data-fusion levels are intended only as a convenient categorization of data-fusion functions. They were never intended to be, nor should they be taken as, a prescription for designing systems; that is, do level 0 fusion first, then level 1, then level 2, and so forth. The SA subprocesses should instead be regarded as "agents" having some degree of autonomy, each one interacting with its own changing environment, which could be the external physical world, or the other agents. Hence, there is a requirement that the SA subprocesses must, at a minimum, communicate with one another or, ideally, cooperate with one another. Any subprocess can communicate with any subprocess.

This is in line with the data-fusion notion of process refinement. The results (or "states") of the SA subprocesses are generally correlated with each other so that good estimation of certain states is likely to yield good estimates of other states, provided the SA process is cognizant of the underlying correlations [51].

The subprocesses at the higher levels of abstraction build on the results produced by the lower levels, and they also feed back conclusions to these lower levels in order to fill in unknowns. For example [52], entities perceived *near a river* (context) might be characteristic of *elements of an engineer battalion* (identification) and, because of their presence near a river and *on the opposite side from the friendly forces* (context), a *bridge-building mission* might be inferred (intent, situation classification/recognition).

Given the inherent data-driven bias of traditional data-fusion practitioners, there is a natural tendency to look at Figure 3.7 through a "clockwise rotation" perspective, that is, starting with acquisition/collection, then going sequentially through structured description (integration/abstraction), classification/recognition, assessment, projection, impact, and monitoring. But it doesn't have to be like that. On one hand, the SA process may start anywhere; there are multiple asynchronous entry points. On the other hand, human information processing in operating complex systems is seen as alternating between data-driven (bottom-up) and goal-driven (top-down) processing. This alternation is viewed as critical in the formation of SAW. In goal-driven processing, attention is directed across the environment in accordance with active goals [42]. The decision-maker actively seeks the information needed for goal attainment, and the goals simultaneously act as a filter in interpreting the information that is perceived. In data-driven (or stimulus-driven, reactive) processing, perceived environmental cues are processed to form SAW and may indicate new goals that need to be active. Dynamic switching between these two processing modes is one of the most important mechanisms underlying SAW.

Although the discussion so far has a slight "technology flavor," no particular a priori human-machine allocation is presumed for the proposed model of SA. At least in principle, each subprocess in Figure 3.7 could be performed by humans, computers, or both. Clearly, the current technology alone is not sufficient to implement the SA process in computers fully. Thus, an optimal mix of human intelligence and technology can be defined for SA. One could say that the proposed model should enable human-factors specialists and knowledge engineers to model the tasks, as well as the knowledge and data structures that are key ingredients in the SA process.

3.5.3 Situation Model

The main purpose of the SA process is to assemble a representation of aspects of interest in an environment. This is in line with the ideas of Lambert [46–48]. The SA process thus incorporates and develops an internal situation model of itself and the environment in which the process operates. This situational model, which the SA process endeavors to keep up-to-date, captures a representation not only of the various elements of the situation but also of how they relate to create a meaningful synthesis, that is, a comprehension of the situation. There is one real world, and the situation model is an abstraction of it [46]. Among other things, it is abstract in the sense of being incomplete (as our attention to the world is selective) [47].

3.5.3.1 Situation-Element Acquisition

A number of sources provide data and information to the SA process at a variety of levels ranging from sensor data, to a priori information from databases, to human input [53]. Hence, multiple types of dynamic and static data and information are made available to the SA process. Source examples include sensors, prisoners, local populace, human intelligence, reference information, and so forth. The sources typically provide only limited observables, coverage, resolution, and accuracy [54].

In the "acquisition" of the situation, various separate and distinct entities, events, or activities are perceived. It is also worth noting that some evidence comes in group form (e.g., a raid report), or the information has to be dealt with as a group because of sensor-resolution limitations. With respect to the environment-perception subprocess, terrain and weather analysis is focused on their effects on friendly and enemy capabilities to move, shoot, and communicate [51]. Given weather and terrain conditions, execution doctrine determines how the enemy will fight.

3.5.3.2 Common Referencing

As discussed above, data and information related to an entity, a battlefield event, a group, and so forth, will often be reported independently via a multiplicity of sensors or sources, each differing in coverage area, spectrum, resolution, response time, and observable sensed [51]. Common referencing is the processing of input reports typically to achieve a common time base and a common spatial reference [53]. Data alignment must remove any positional or sensing geometry and timing effects from the data and information [51]. The subprocess also transforms source data into a consistent set of units and coordinates for further processing [53].

Finally, this subprocess could also have to deal with other important issues, such as the alignment of different uncertainty frameworks.

3.5.3.3 Situation-Element Association

Association is a basic subprocess necessary to determine which situation elements at the input of the SA process associate to which situation elements currently being maintained in the situation representation (i.e., the situation model). Association is necessary to deal with the uncertainty attached to the situation elements. A classical example is to determine whether entity data, which have been reported by different sources, represent the same entity or different entities; in this case, one talks about data-origin uncertainty management.

The association process can make either hard decisions or soft decisions about which of a number of hypotheses best describes the association of input situation elements received from some sources with situation elements contained in the current situation model. A hard decision is a definitive association to one, and only one, interpretation possibility, while a soft decision allows the data to be associated to multiple interpretation possibilities, with each candidate association having a measure of uncertainty. The soft-decision approach typically results in multiple association hypotheses being maintained until additional input situation elements have been collected and there is enough information data available to reduce the uncertainty and to substantiate or refute the prior hypothetical associations.

Thus, multiple-hypothesis data association (MHYDA) inherently uses later input data to aid in evaluating prior correlation decisions. Note that ultimately, though, a final decision has to be made. In principle, this approach should lead to the most accurate association results. However, the computational requirements necessitated by the ability to retain multiple interpretations of the situation represent the main drawback of the standard (hypothesis-oriented) MHYDA algorithm [55]. To reduce (actually minimize) these computational requirements, the number of data association hypotheses must be limited (sometimes sacrificing optimality) through the use of hypothesis pruning and combining methods. Moreover, since the amount of computer storage and computation time grows exponentially with the number of situation elements for the MHYDA algorithm, the combinatorial problem associated with forming multiple temporally continuous hypotheses can also be significantly reduced by dividing the entire set of situation elements into separate groups or clusters. See [56–59] for more detail about the problem of multiple-hypothesis data association.

3.5.3.4 Situation-Element Refinement

In the presence of uncertainty for a complex environment, where an unknown number of entities is entering the volume of interest at any time, while some others are leaving this same volume or are being destroyed (we also include here the random false alarms and the clutter), there is an evident requirement for an existence-analysis subprocess. Referring to the target-tracking terminology, once tracks are formed and confirmed (so that background and other false targets are reduced) and low-quality tracks have been deleted, the number of targets can be estimated. Note that the number of groups and their size are also of interest.

Situation abstraction [54], which includes both situation generalization and situation specialization, is an interesting concept with respect to existence analysis. Situation generalization allows bottom-up abstraction of entities, events, or groups that are either not directly measurable or perceived or that must be inferred [54]. Situation specialization is a form of top-down reasoning where subordinate elements are deduced or inferred. Situation abstraction attempts to fill in missing information and to develop a more complete and integrated situation representation than is possible using reasoning based strictly on direct observables.

The kinematics-analysis subprocess assembles a representation of the kinematical properties of the situation elements maintained in the situation model. The usual kinematics properties are the position, velocity (course, speed, angular rates, Doppler), acceleration (maneuvers), and the attitude (pitch, roll, yaw) of an entity. Identity analysis is the subprocess by which some level of identity of a situation element is established, either as a member of a class, a type within a class, or a specific unit within a type [53].

Certainly, those three subprocesses (i.e., existence, kinematics, and identity analyses) are the aspects of information fusion that have been studied the most so far; there is a huge body of literature readily available to the reader that covers them in depth.

3.5.3.5 Situation-Element Contextual Analysis

One crucial aspect of successful SA is understanding that the various assessments optimally derive from examining the data and information from multiple contextual viewpoints [51]. The context is the interrelated conditions in which something exists (e.g., an entity) or occurs (e.g., an event). Hence, as we progress through it, we desire the SA process to represent more than just measurable properties of situation elements; relationships among them are also a key aspect of interest [45–48].

The contextual-analysis subprocess thus develops a description of all sorts of relationships among situation elements: physical (is composed of), spatial, temporal, structural, organizational, perceptual, functional (involves/requires/provides), causal, informational, and so forth. Clearly, however, we are talking here about relationships of interest between situation elements of interest.

Given such relationships, group formation and refinement is also possible. By forming individual situation elements into groups, further inferences on attributes, identity, allegiance, function, and mission may be possible. Groups also form a fundamental component of SA for inferring what tactics the total set of enemy entities is employing.

3.5.3.6 Situation-Element Interpretation

Once an entity has been perceived, its kinematics have been determined, and it has been identified, the decision-maker typically wants to know what the object is doing, that is, its behavior [60]. The behavior is the particular manner in which something bears, conducts, or comports itself. It is highly linked to the notions of performance and action (i.e., activities). Behavior related to threat assessment can include elements of positional information: direction, speed, and maneuvers. It can also include operation of equipment: jamming, using radar or laser systems, opening weapon bay doors, and releasing weapons. Not all of these aspects of behavior are likely to be found at the same time, and the same combination of behavior elements may have different threat connotations, depending on circumstances. Note that besides threat assessment, behavior can be a source of information for other SA subprocesses, such as kinematics and identity analyses.

The SA process also performs an analysis of the level of activity. For example, an increase in the level of communications may indicate movement of units. An increase in the level of use of active sensors may indicate abnormal activity. In general, the monitoring of the level of activity may highlight a build up during the development of a crisis. Decision-makers are often interested in a description of the latest known enemy activities in an area. Lastly, note that the absence of activity is also of interest.

3.5.3.7 Intent Analysis

Especially in military environments, account has to be taken of the intentions of the forces concerned [60]. Intent estimation has a lot to do with interpreting or explaining (i.e., give the "why," the reason for or cause of the presence of entities or their behavior). The notion of intent revolves around the ideas of aim, goal, target, objective, plan, and purpose.

More precisely, an intention is a determination to act in a certain way [44]. Given this definition, one can consider the intent formulation, that is, act in a certain way (e.g., "sink the protected unit"), and the intent strength, that is, the level of determination of the player (e.g., firm intent versus weak intent). Both aspects can be considered for intent analysis. Note that the intent formulation may be true or false, with some confidence level attached to each, while the intent strength can be a continuous value.

Intent analysis plays an important role in the calculation of the inherent-threat value. Knowing the true intent of a suspect entity can greatly support the projection of the current situation. For example, if a plane currently flying directly towards a protected unit is positively identified as a friendly civilian aircraft, then one might be tempted to conclude that the intent of the pilot is to pass over the protected unit with absolutely no consequence or impact on it. In such a situation, knowing that the true, firm intent of the pilot is actually to commit a "suicidal mission" is highly critical to predict the true upcoming consequence, that is, that the aircraft will crash into the protected unit, causing important damage.

Intent analysis also supports the plan-analysis process in the recognition of the current plan(s) being followed by some entity. For example, knowledge of the intent of a suspect entity may greatly decrease the size of the search space for the various plans to be matched with the situation elements perceived about this entity.

Finally, some estimated numerical values for the intent of an entity (e.g., a confidence on the intent formulation and a determination index) can be used as an input in some mathematical formula for threat-value calculation. However, this must be developed with great care since it is the consequences of actual actions that may ultimately have an impact on the protected unit, not the intent of any player. For example, it may happen that an entity overestimates its capabilities and consequently develops a firm, hostile intent towards the protected unit. In such a situation, if the true capabilities of the entity are correctly assessed as low by the defending unit, then the inherent-threat value assigned to this entity should also be low, even if its firm, hostile intent is well known. And the opposite is also true. A situation may also arise where the consequence of some action, if not stopped, will have a major impact on the protected unit, although the true intent of the player has nothing to do with this unit. As a simple example of this, how often have we heard a child say, "I had no intent to break this valuable thing"? This is often very true; the child really had no intention of breaking the thing. Unfortunately, this child has performed an action, and the corresponding consequence is there: the thing is broken. This is called *collateral damage*.

It is proposed that numerical values for the intent of an entity could play a role in the threat-value calculation only when one has a poor estimate of the damage power of an entity. In such a situation, if one knows for sure that the entity has the firm intent of harming the protected unit, then one might be tempted to eliminate the entity "just in case."

Unfortunately, intent analysis is not an easy task. Behind an intent are some interests or desires on the part of a player. These define why the enemy will fight, its high-level goals. They are mostly based on the perception or comprehension, beliefs, values, principles, and culture of the player (defining some player profile), and these are very intangible parameters. Nevertheless, if some classification or identity information is available, then some knowledge of an entity's interests and desires may be available from the intelligence process (e.g., there might be a history of conflicts). Note that the necessary information about the capabilities and vulnerabilities of the entity can also be obtained from a priori intelligence data. Spatial and temporal analyses are then performed to assess if there are favorable junctures of circumstances or opportunities for the suspect player to achieve its goals. This analysis of the opportunities must also take into account the constraints, such as the rules of engagement, of the suspect player and the environmental conditions. The latter includes the terrain and weather effects on mobility, sensors and weaponry, the constraints of international treaties or alliances, and so forth. Finally, it is by evaluating and jointly processing strengths, weaknesses, and opportunities with desire that some estimates of the intent and determination of a suspect player can be derived.

3.5.3.8 Plan Analysis

Military operations, even those that are covert or involve surprise, are typically guided by a plan or set of plans, because such operations are complex, involve multiple resources and goals, and require significant coordination [51]. The red war plan defines why, where, and when the enemy will enter into combat and with what force structures, schedules, and operations. If this assertion is true and if the general doctrines that guide red force actions are known to a blue force (at least in part), then the blue force can hypothesize the use of certain red force plans and, based on incoming multisensor data, assert the possible existence of particular red plans and use such assertions for decision-making and action. That is, making hypotheses (through the exploitation of doctrinal or exercise-based knowledge of hostile behavior) about the plan of a player provides a framework for the fusion of the data and information perceived and inferred about this player [51].

Clearly, plan recognition can be useful to infer the intent of unknown units whose presence was previously unexplained and, more importantly, to forecast the imminent actions of a player, leading to an eventual assessment of the threat value of the corresponding upcoming consequences. Clearly, also, behavior analysis has a significant role to play regarding plan analysis.

3.5.3.9 Situation Classification and Recognition

Situation classification is the systematic arrangement of situations into groups or categories according to established criteria. It has to do with the cataloging and sorting of situations. Multiple abstract models of situations may be available a priori. Associated with these models may be schemata of prototypical situations. Critical cues in the environment may be matched to such a priori schemata to indicate prototypical situations that provide instant situation classification and comprehension [42].

Situation recognition is the action of perceiving the situation to be something previously known. A very familiar situation, whatever the level of danger, may simplify decision-making. However note that such practice may lead to mental fixation, sometimes with deleterious or disastrous consequences (failing to foresee massive deception or a successful surprise attack, not anticipating enemy behavior on the battlefield, and the like) [51]. Unfamiliar situations may trigger various actions from the SA process (e.g., tasking the data-collection sources to gather more information).

3.5.3.10 Situation Assessment

To assess a situation is to determine its importance, size, or value. A *situation assessment* is thus a quantitative evaluation that has to do with the notions of judgment, appraisal, and relevance. While behavior analysis is about what entities are currently doing, the capability- or capacity-analysis subprocess is about what they can do. This includes various force-evaluation functions that will determine what the assets of the participants (own or enemy) are. Situation assessment also attempts to determine forces' important intangibles: morale, psychological state, level of training, stability under stress, strength of will, and so forth [51]. The *salience* of something is its striking point or feature. The perceptual salience of environmental cues is the degree to which they draw attention [42]. Salience analysis must assess what is outstanding in the current situation (e.g., an exceptional event, an entity that requires special attention).

3.5.3.11 Situation-Element Projection

This SA subprocess is necessary because one is not only concerned with what is happening but also with what events or activities are going to happen next. The decision-maker can never influence the present, only the future. Hence, knowledge of the current world state is only of value as a contribution to understanding the future. Situation-element projection must produce an estimate of future possibilities for situation elements based on current trends and expectations. Ultimately, the predictive capability can include story building, simulation, war gaming, engagement modeling, and so forth.

3.5.3.12 Impact Assessment

Impact is defined as one thing's force of impression on another, as an impelling or compelling effect. There is also the notion of *influence*, of one thing influencing another. According to the data-fusion model maintained by the JDL DFG, impact assessment has to do with the estimation and prediction of effects planned, estimated, or predicted actions by the participants [45]. It draws inferences about friendly and enemy strengths, vulnerabilities, and reinforcement capabilities, cost and utility implications of estimated situations, problems and opportunities for operations, and so forth. Accurate impact assessment requires applying the concept of shifting perspectives to the data and information to develop an optimum viewpoint of the situation [51]. This means examining the data from each of red, blue, and white viewpoints.

Impact assessment includes an analysis of the interactions between action plans of multiple players (e.g., assessing susceptibilities and vulnerabilities to estimated or predicted threatening actions, given one's own planned actions). Such interactions are illustrated in Figure 3.8. Note that a plan typically results in a sequence of actions from a given player. In turn, an action eventually produces some consequence, disturbing the environment, which may or may not be synchronized in time with the originating action. Ultimately, a consequence may have a large impact on the plan of another player, forcing this player to cancel upcoming planned actions.

Impact assessment should be implemented as a prediction function, drawing particular kinds of inferences from the current situation representation. Impact assessment estimates the outcome of various plans as they interact with one another and with the environment. The impact estimate can include likelihood and cost/utility measures associated with potential outcomes of a player's planned actions. Indeed, impact assessment is often about computing some cost, given an aggregation, for instance, computing the probability of killing a ship, given current and expected relational states between the ship and other entities [45]. Note that because Steinberg, Bowman, and White have defined situation assessment so broadly, impact assessment is actually a subset of situation assessment in their model. Whereas situation assessment involves estimating or predicting all types of relational states, impact assessment involves predicting some or all of the relationships

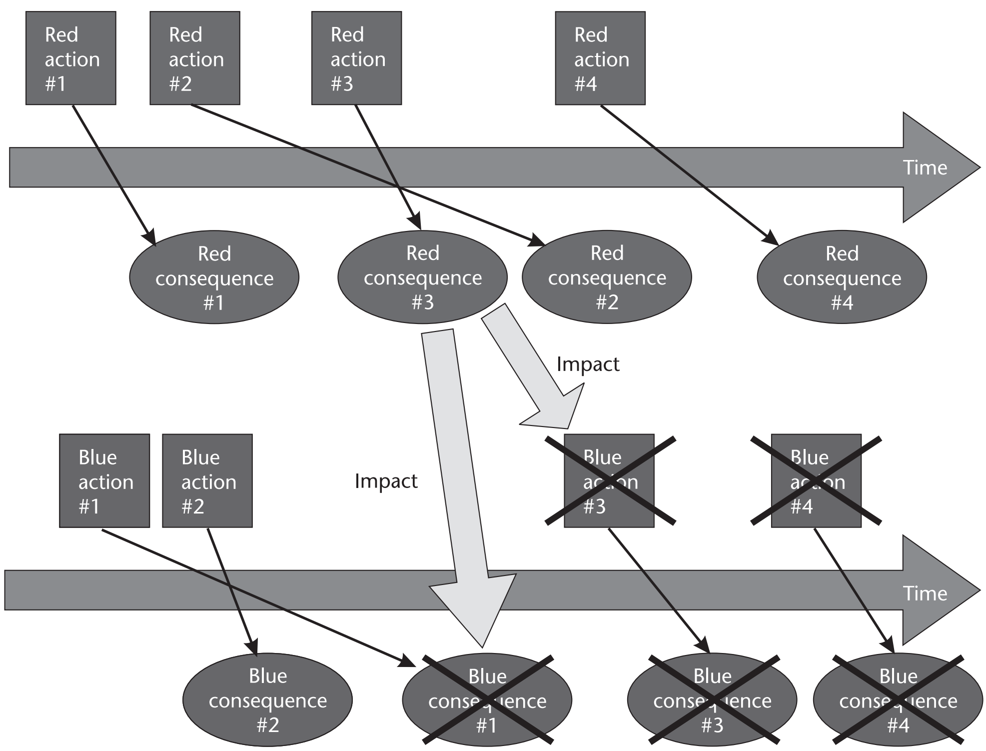


Figure 3.8 Interaction between the plans of two players.

between a player and his or her environment, to include interaction with other players' actions, given the player's action plan and that of every other player [45].

3.5.3.13 From Desire to Consequences

Impact assessment, as formulated in the JDL DFG model, is the foundation of threat analysis. However, the concepts presented in Section 3.5.3.12 need to be further refined for a practical threat-analysis application to be developed. In particular, some notions, like intent, capability, opportunity, and so forth, need to be taken into account. Planning means deciding on a course of action before acting [51]. A plan is thus a representation of a course of action. It can be an unordered list of goals, but usually a plan's goals have an implicit order. Most plans have a rich subplan structure; each goal can be replaced by a more detailed subplan to achieve it. This is illustrated in Figure 3.9.

Figure 3.10 is a different, more complete view, putting the notion of intent in relation to its driving factors. As previously mentioned, an intention is a determination to act in a certain way. Given an intent, a detailed formulation of a program of action is required to achieve the goal. This is the plan, that is, the method for achieving the desired end. Behind an intent are the player's interests or desires. A desire is defined as a conscious impulse toward something that promises enjoyment or satisfaction in its attainment. This is the starting point of the process; that is, one must have the desire to achieve something. However, mere desire is not enough. One must also have the basic capabilities, that is, the facility or potential for an indicated use or deployment required to achieve a goal. One must take into account any vulnerabilities, for instance, ways one is in a position to be physically wounded or is open to attack or damage. Finally, there might eventually be a favorable juncture of circumstances, that is, an opportunity to achieve the goal. Only by weighting strengths, weaknesses, and opportunities with desire will the real intent emerge.

Figure 3.10 describes well how players go from desires to actual actions in the environment. Because of the dynamic nature of its parameters, this process is far from static. Desires change with time. Opportunities come and go. Capabilities and vulnerabilities evolve with the situation. More importantly, there are interactions

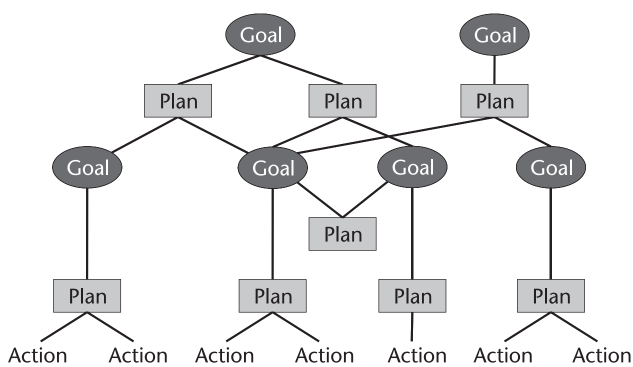


Figure 3.9 Example of a generic plan-goal graph.

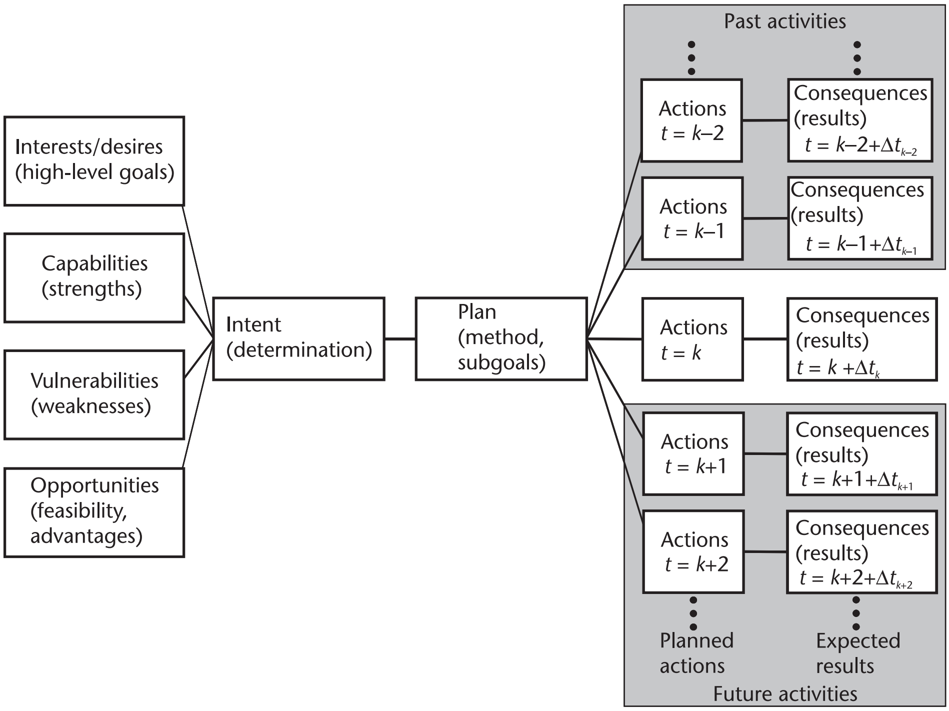


Figure 3.10 From desire to consequences for a given player.

between the various players, with the actions of one player potentially having an impact on the intent and planning process of another player. When operating in a given environment, one can follow proactive, predictive, or reactive strategies, or a combination of the three, regarding the activities of another player. This is illustrated in Figure 3.11.

A good example of proactive behavior is issuing a warning. Typical warnings are, "You are approaching our warship. Turn, or we will engage," or "I will defend myself." Such warnings may be sufficient to cause other players to change their desires and modify their current plans. One can also force the opponent into some location that will decrease the feasibility of its planned actions, thereby removing some opportunities that once were driving factors for this opponent.

If the desires, capabilities, and vulnerabilities of an opponent can be estimated (e.g., from intelligence reports), along with the opponent's opportunities (e.g., from the situation timing and geometry), then one can predict the opponent's intent and, consequently, infer its possible plans or course of action. Finally, by observing the past and current actions of a player and their consequences, one can potentially recognize the plans of this player and then infer its intent.

3.5.4 Threat Analysis

While conducting command-and-control activities, decision-makers eventually have to manage actual or potential threats to some protected units. By definition, a

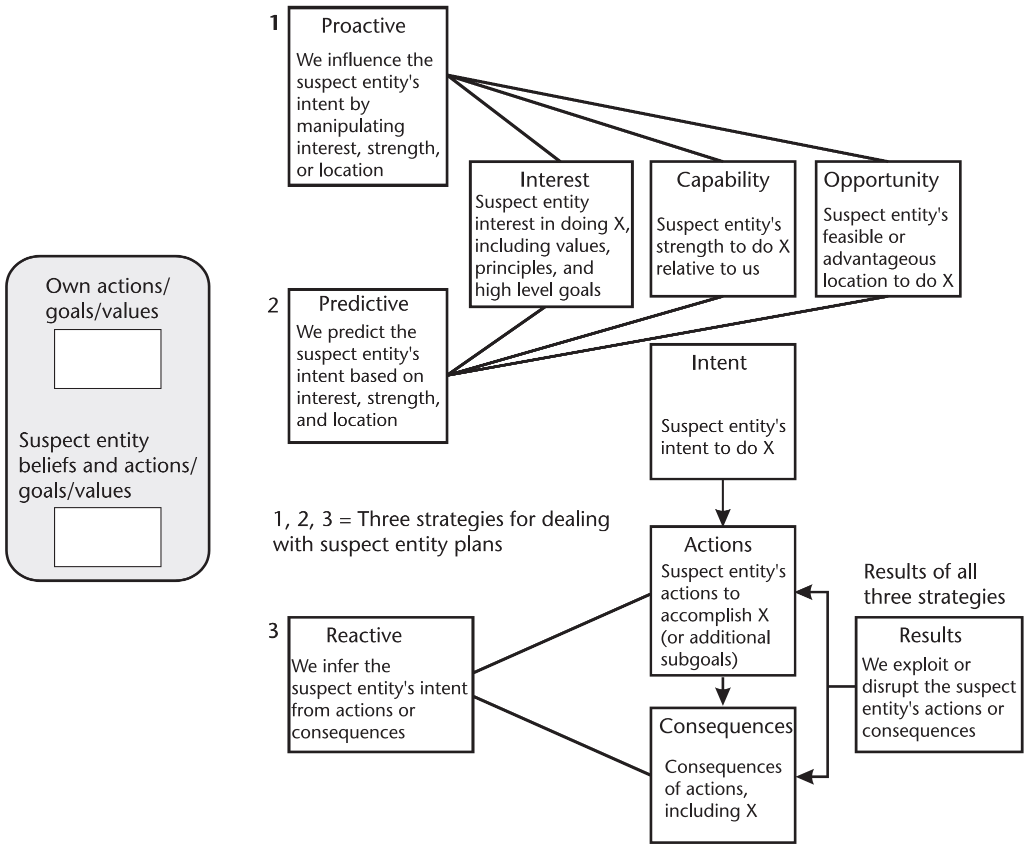


Figure 3.11 Strategies for dealing with suspect entity plans. (*After:* [61].)

threat is an expression of intention to inflict evil, injury, or damage. The focus of threat analysis is to assess the likelihood of truly hostile actions and, were they to occur, projected possible outcomes [51].

Typically, the situation of every entity must be evaluated to determine its degree of threat within the context of the mission's overall objectives. This first requires looking at the kinematics information (e.g., closest point of approach, range, speed) and the contextual and a priori information (e.g., lethality, doctrine, rules of engagement, tactics, resource inventory and status) regarding entities. Thus, mostly based on capability and intent, threat analysis generally attempts to compute some threat value that estimates the degree of severity with which engagement events will occur. This amounts to (1) quantitatively portraying the capability, and (2) coupling this picture with an estimate of intent [51]. Once the hostile intent of an entity has been clearly established, the significance of the threat is proportional to the perceived capability of this entity to carry out that threat.

Threat and risk assessments are eventually made, forming a basis for making decisions about the use of defensive means to maximize survivability and achieve the mission. In this regard, it is of utmost importance for decision-makers to be able to determine which of several threats represent the highest danger as errors, such as prioritizing a lesser threat as a greater threat and ultimately engaging the wrong target, can result in dramatic consequences (damage, injury, or even death). Clearly, threat analysis must be performed with great care.

In view of the introductory discussion above, we formally define threat analysis as:

> The analysis of the past, present and expected actions of external entities, and their consequences, to identify menacing situations and quantitatively establish the degree of their impact on the mission, the intents, the plans, the actions, and the human and material assets of some valuable units to be protected, taking into account the defensive actions that could be performed to reduce, avoid, or eliminate the identified menace.

Introducing the notions of *inherent-threat* and *actual-risk assessments*, Figure 3.12 illustrates the threat-analysis concept and its relation to response planning. The concept of inherent-threat assessment has to do with quantitatively establishing the degree of impact of each upcoming consequence resulting from actions performed by other players. The idea is to quantify the intrinsic level of danger or menace (i.e., the potential for causing harm, damage, or mission failure) of a consequence if nothing is done to prevent its happening.

As shown in Figure 3.12, the assessment of the inherent threat involves the threat-value-calculation and threat-value-ranking subprocesses. The former is a process (e.g., a mathematical or rule-based process) that assigns a numerical inherent-threat value to a consequence, reflecting the degree of inherent threat evaluated for the consequence according to a number of factors representing predetermined threat criteria. For example, this value could be a number between 0 (no threat) and 1 (the highest threat value). Threat ranking simply ranks the consequences, for instance, from the most threatening to the least threatening, based on their assigned threat values. Ultimately, a prioritized threat list is generated, listing the menacing consequences to be considered for actual-risk assessment. Note that in practice a consequence is often mixed with the entity (usually internally represented

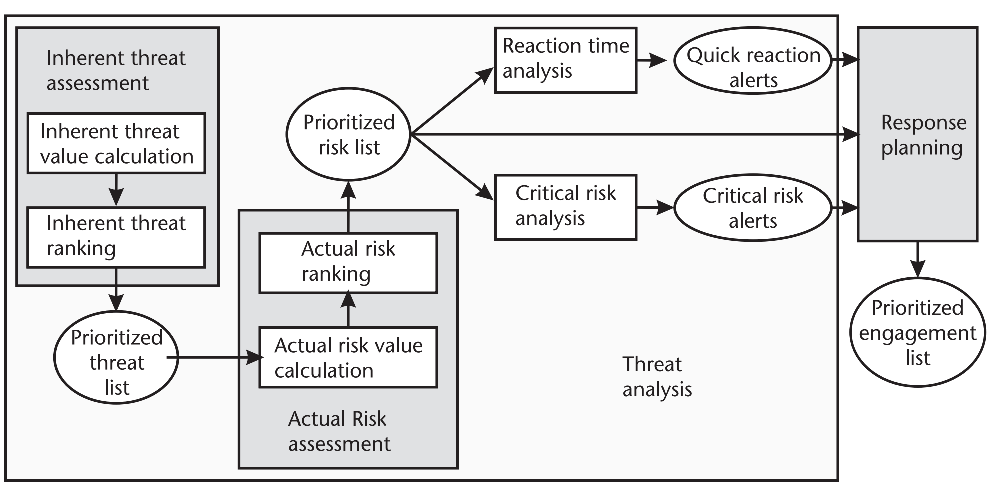


Figure 3.12 Threat analysis and response planning.

in a computer-based situation-analysis system as a track) that will actually produce it. Therefore, the prioritized threat list could contain a list of entities (or tracks), each with an attached inherent-threat value. This makes sense since eliminating a threatening consequence often boils down to eliminating the entity that will produce it.

The actual-risk-assessment concept takes into account the defensive actions that can be performed to reduce, avoid, or eliminate each menace previously identified through the inherent-threat-assessment process. It is an analysis step that asks the following questions to fine out how easy it is to avoid or defeat each individual threat on the prioritized threat list:

- Do we know how to tackle the problem posed by the threat?
- How many defensive options do we have to avoid or defeat the threat?
- What is the quality of each option?

Answering such questions should influence or modulate the threat value previously computed by the inherent threat-value-calculation function. The threat value is then transformed into an actual-risk value that better reflects the actual potential for danger. On one hand, an entity that has been assigned a very high inherent-threat value could ultimately represent a very small risk if it is very easy to take care of it (i.e., there are numerous, good-quality options to tackle the problem). On the other hand, a moderate threat entity may represent a high risk if there are no options to counter it. In the end, the actual-risk assessment process generates a prioritized risk list as an input to the response-planning process that, in turn, considers all of the threats and defensive options together to produce a prioritized engagement list.

Note that from the prioritized risk list, one can also perform reaction-time and critical-risk analyses. Concerning the former, a number of valid defensive options might be identified for a given threat. An analysis is then performed on each option to identify the time remaining before the option has to be initiated for the defensive response to be valid. From this analysis, high-risk entities requiring immediate responses can be identified. Quick reaction alerts can then be generated by the threat-analysis system for consideration by the defending decision-makers. The purpose of critical-risk analysis is to identify significant threats for which there are no, or only a few, defensive options available. Actually, it performs a sort of thresholding process on the actual-risk value to identify highly critical conditions. Again, critical-risk alerts can be generated for consideration by the defending decision-makers. Figure 3.13 illustrates some of the factors that influence threat analysis.

3.5.4.1 Inherent-Threat Assessment

Inherent-threat assessment is the part of threat analysis that looks at the past, present, and expected actions of external entities and their consequences to identify menacing situations and quantitatively establish the degree of their impact on the mission, intents, plans, actions, and human and material assets of some valuable units to be protected. At this stage of the analysis, an entity should not be categorized

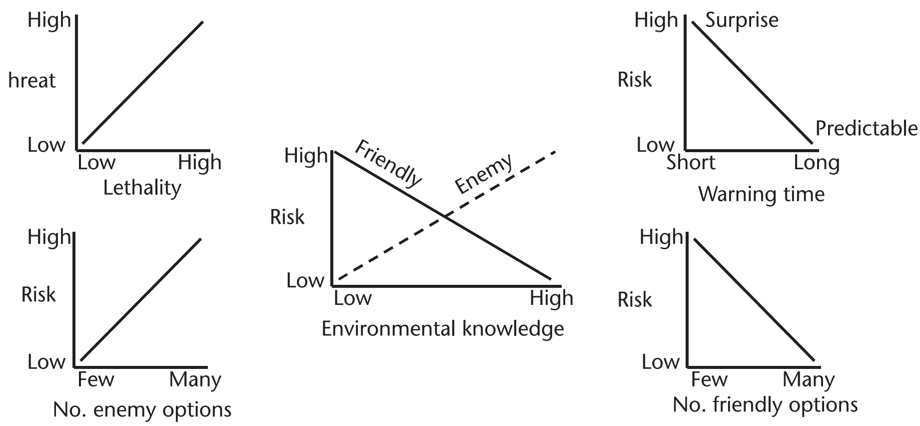


Figure 3.13 Examples of threat-analysis factors. (*After:* [61].)

a threat or nonthreat, based on a friendly weapon's ability or inability to engage it. The essential step of this assessment process is the inherent threat-value calculation function. A high-level view of the relevant elements of this function and their relationships is provided in Figure 3.14.

As a result of past and current actions by the players in the environment, a number of entities and events have already been perceived by the sensing system, providing data and information to construct a basic model of the current situation.

3.5.4.2 Situation-Geometry Analysis

By coupling the results of the kinematics analysis with knowledge of the environment and the capabilities, one can establish the current situation geometry. Figure 3.15 illustrates some typical components of the situation geometry that are relevant to threat analysis.

The threat reference point (TRP) is an important parameter, as it is the position on which the threat assessment is based. It can be static (e.g., a fixed position defined by a static special point) or dynamic (e.g., a friendly track or a dynamic special point). The TRP typically depends on the defending unit's defense role (e.g., point defense or supportive area defense) and the associated defense priority (i.e., priority is point defense, for instance of own ship in a naval operation, only, where priority is given to point defense over supportive area defense). Other typical geometry parameters illustrated are the sensor and weapons coverage zones, the red weapon lethality zone, and the seeker angle (in the case of a missile). Concerning vulnerability to fragment attacks, a keep-out range may be defined as the minimum range within which an incoming entity must be damaged to avoid any possible damage (from entity debris) to the protected or defending unit. The definition of such a keep-out zone involves key issues such as small fragment impact, multiple fragment impacts, fragment breakup and effects, blast, blast or fragment synergism, vulnerability or lethality modeling, response of missiles to damage and their post-damage trajectories, and so forth.

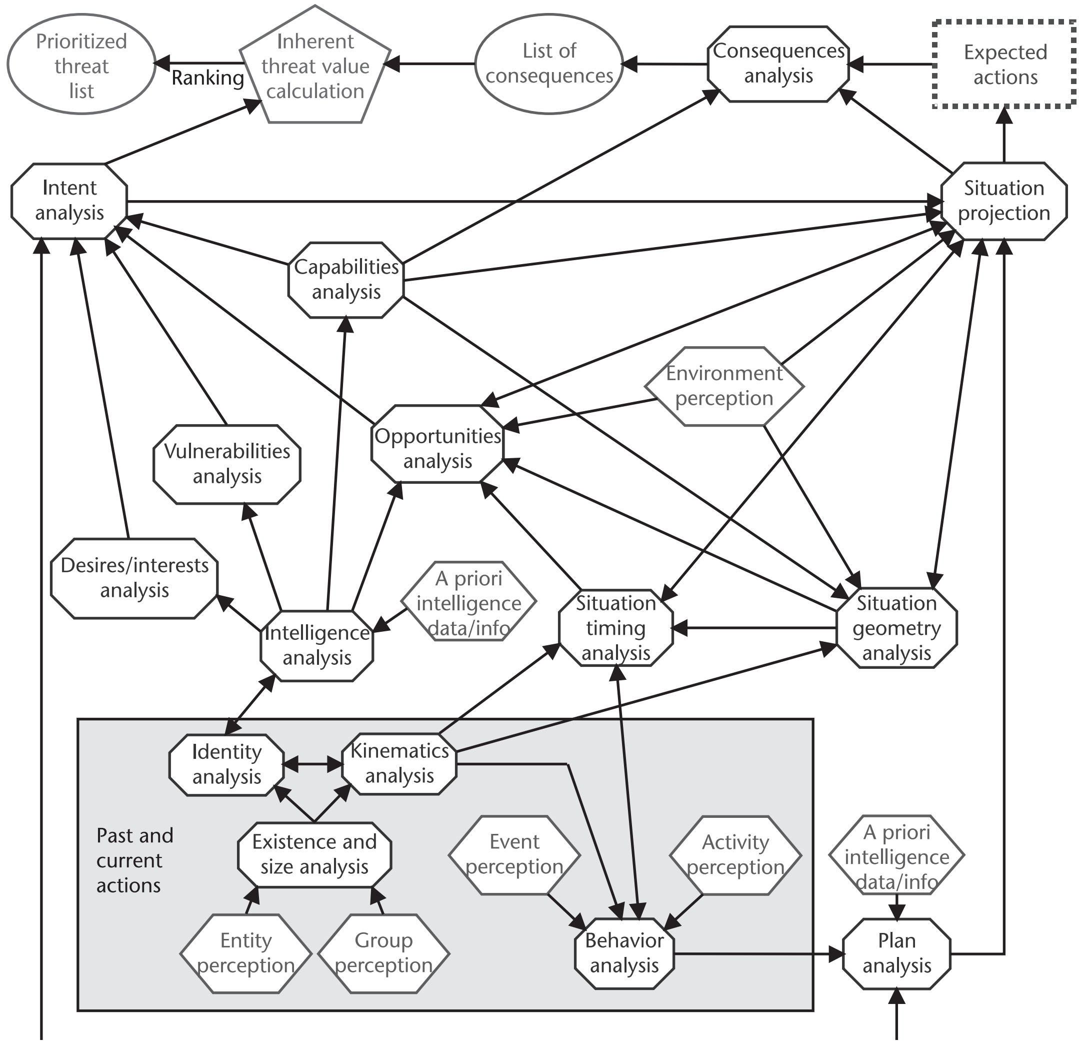


Figure 3.14 Inherent threat-value calculation.

As is also illustrated in Figure 3.15, one is concerned not only with what is currently happening but also with events or activities that are going to happen next. A key parameter obtained from situation projection and often used with threat-evaluation techniques is the closest point of approach (CPA) (see Figure 3.15). The CPA for an entity is the point at which it will be the closest to the TRP, given current trends and expectations. Typically, for a static TRP, an assumption is often made that the current velocity of the entity will remain constant for the duration of the evaluation. In such a situation, the CPA is the point at which the projection of the current course of the entity meets, at a 90° angle, the radius of the circle centered on the TRP and with a radius long enough that the current course of the entity is tangential to this circle. The radius of this circle, that is, the distance from the TRP to the entity's CPA, is called the range at CPA (RCPA). The time to CPA (TCPA) is then the time it will take for an entity to reach the

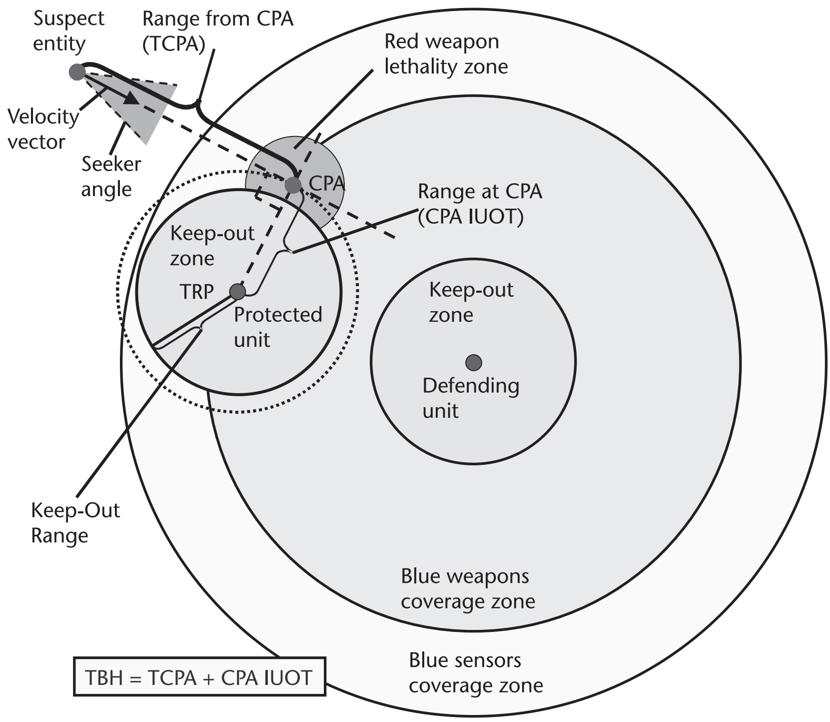


Figure 3.15 Some situation-geometry and timing elements for threat analysis.

CPA if its velocity remains constant (i.e., the entity's current distance from the CPA divided by its current speed).

If an assumption is made that an entity reaching its CPA will simply turn 90° toward the TRP and maintain its current speed, then a sense of the "proximity" of the entity to the TRP, when the entity is at its CPA, can be gained by converting the CPA distance into time. Dividing the CPA distance by the entity's current speed provides an indication of the time it will take the entity to hit the TRP from the CPA. This is referred to as the CPA in units of time (CPA IUOT). By converting the CPA distance into units of time, one can now plot this CPA distance on the same time-based plot as the TCPA, thereby getting a sense of the direction, relative to the location of the TRP, of the entity's course. The overall time before the entity reaches the TRP, that is, the TCPA plus the CPA IUOT, is called the time before hit (TBH). Other similar temporal parameters often mentioned are the time to go (defined as the time before "metal hits metal"), the time to penetrate the lethal zone, and the weapon time to intercept the target.

3.5.4.3 Situation-Element Kinematics Smoothing Techniques

The threat-analysis process may require the use of smoothing techniques for the estimation of the kinematics of entities in order to ensure that changes in the overall threat or risk priority of these entities take place gradually. This is most effective with maneuvering entities to ensure that one update will not remove from the threat or risk list an entity that only has to be readded during a subsequent update.

Whether the entity is an aircraft or a missile, it generally never goes in a straight line to its target (e.g., the protected unit). Instead, it will often maneuver while approaching the target. Hence, in order to compute meaningful threat or risk values (very often functions of the CPA and TCPA, among other parameters) to be assigned to each potentially menacing entity, some stabilization process is thus needed for the kinematics situation elements maintained in the SA system situation model. Such a process will essentially make a kinematics smoothing of the trajectory of each entity in order to stabilize its CPA and TCPA, as well as any other time-space parameters such as these.

Different approaches to kinematics smoothing can be considered. Mathematical approaches try to discover an analytical expression (equation) for the trajectory. The equation is then used to compute the equation of a straight line corresponding to the mean direction of advance of the entity. In some situations, real-time constraints require the use of simple methods to compute such a direction of advance. For instance, the least-squares regression line can be computed over the points that form the trajectory. Each time a new position of the entity is reported, a new regression line is computed. Even if this method is simple and attractive from a computational point of view, it may actually take more time than the analytical expression approach before the direction of advance stabilizes. Other methods are based on feature-extraction approaches. They attempt to extract the main features or characteristics of the trajectory, such as sharpness, size, and changes in oscillations. These characteristics are then used to determine the direction of advance of the entity. Neural networks are well adapted to feature extraction. Depending on the type of network and the information used to train it, the extraction of features can be thought of as a spatial analysis of the trajectory.

All methods of kinematics smoothing depend on a number of factors that directly influence their performance:

- *Oscillation frequency:* This parameter reflects the capacity of the entity to make sharp turns within a short period. Such sharp turns of different "forms" increase the difficulty of determining the main features of the trajectory. Hence, stabilizing the direction of advance of the entity may take much more time than that required for soft and regular turns.
- *Observation frequency:* The positions of the entity are usually generated using reports from sensors such as radars. The number of reports within a period has a direct impact on the time of stabilization of the direction of advance. On one hand, if the trajectory is oversampled, some performance and computational issues may arise. On the other hand, if it is undersampled, it may be too difficult to characterize the trajectory.
- *Temporal window:* Whatever the method used to stabilize the direction of advance, only a limited number of position points can be considered or one may encounter computational limitations. Generally, one selects a temporal window, that is, a period during which the new reported positions will be taken into account by the stabilization process. Obviously, the larger this window, the slower the stabilization will be. However, if this window is too small, it will not be sufficient to characterize the trajectory, hence, to stabilize the direction of advance.

All of these factors must be taken into account before choosing an approach to kinematics stabilization. For example, mathematical approaches would be misapplied in the case of sharp and highly irregular oscillations. However, they are well adapted to soft and regular oscillations. Neural approaches are well adapted to irregular oscillations and uncertain data, but mathematical approaches will give better solutions with regular oscillations.

3.5.4.4 Consequences Analysis

Getting an understanding of the consequences of the various actions performed by the various players in the environment is a critical step in threat-value calculation. One is concerned with identifying the nature of each consequence—for instance, an explosion, a physical hit ("metal hits metal"), or the release of highly toxic chemical substances—and quantifying the impact of this consequence on the mission, intents, plans, actions, and human and material assets of some valuable units to be protected. Regarding the latter, one must establish some scale to quantify, based on the potential for "blue losses," the degree of impact of the various types of consequences. Table 3.1 presents a very basic example of how various levels of impact can be defined.

Identifying the nature and impact of each expected consequence requires pretty good predictive capabilities that can include story building, simulation, war gaming, engagement modeling, and so forth. Knowledge of an entity's capabilities is required for the situation-projection function. One is interested in the maneuverability of the entity (e.g., turn rate, acceleration, thrust vector control or actuation), its guidance mechanisms (e.g., active or semiactive or -adaptive, dual-mode seeker), the fusing system, and so forth. One may also be interested in the weapons launcher's characteristics (e.g., the reload rates and stockpiles). Knowledge about the logistic networks (e.g., levels of resupply capability), intelligence capability, communication nets, level of training, and various other conditions may also be required. Finally, with respect to assessing the degree of an entity's potential impact, one is interested in the envelope of the hostile weapons (i.e., the effectiveness zone) in terms of the probability of hit and the lethality in terms of probability of kill or kill radius.

3.5.4.5 Inherent Threat-Value Calculation

Inherent threat-value calculation is a process (e.g., mathematical or rule-based) that assigns an numerical inherent-threat value to a consequence, reflecting the

Table 3.1 Value of the Impact of Consequences of Actions

Impact Scale	*Material Damage*	*Human Losses*	*Blue Planning*	*Blue Mission*
0	No damage	No losses	No change	No impact
1	Small damage	No losses	No change	No impact
2	Moderate damage	No losses	Small change	No impact
3	Severe damage	No losses	Moderate change	Mission delay
4	Moderate damage	Small human losses	Severe change	Mission delay
5	Severe damage	Severe human losses	New plans required	Mission abort
6	Sinking of the ship	Severe human losses	N/A	Mission abort

degree of inherent threat evaluated for the consequence according to the estimated intent of the entity causing the consequence and its estimated impact. For example, it could be a number between 0 (no threat) and 1 (the highest threat value). Based on the results of the inherent threat-value calculation process described above, the threat-assessment process must produce and update a prioritized threat list for presentation to the operator and subsequent use by the actual-risk assessment function. Threats must be ranked on the list, top to bottom, from greatest to least. Note that presenting tracks to decision-makers in a threat list, sorted from the most threatening to the least, is clearly in line with the cognitive demands associated with threat evaluation.

3.5.4.6 Actual-Risk Assessment

Through the calculation of an actual-risk value, one attempts to quantify how easy it is to avoid or defeat each individual threat on the prioritized threat list. This is illustrated in Figure 3.16. Note that the process reuses a number of results derived from the inherent threat-value calculation. This step of the threat analysis starts with a review of the various defensive means available and defensive actions that could be performed to reduce, avoid, or eliminate each threat on the prioritized threat list. In addition to the use of classical weaponry systems (both hard kill and soft kill), other options, for instance, changing the ship's disposition to avoid the threat or issuing a warning to the suspect entity to influence its intent, are also considered and matched to the set of expected consequences. This is where the available response time (e.g., the time until impact or until likely red weapon release) against aggressive threats becomes a very important issue. As the reaction time decreases, so does the number of defensive options applicable to the problem, thereby increasing the actual risk to the protected unit.

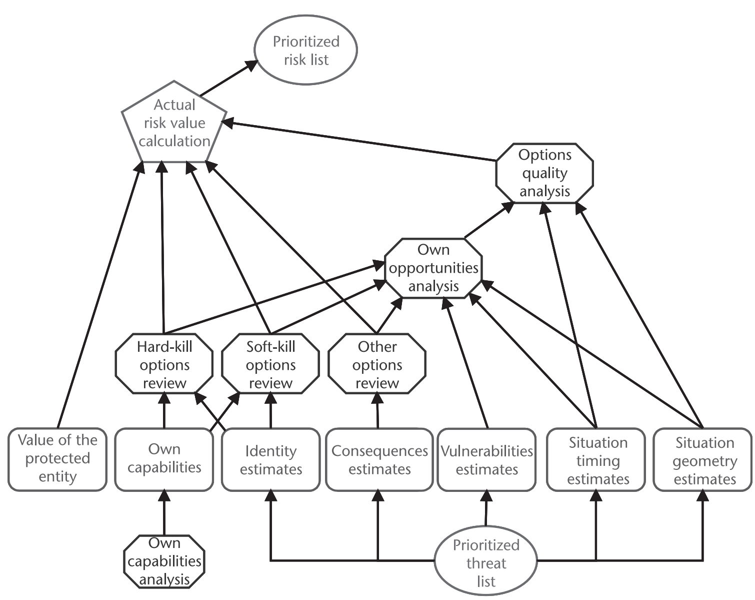


Figure 3.16 Actual-risk calculation.

Given all of the possible options to counter the various threats, potential opportunities for the blue forces in terms of their ability to engage the enemy effectively must be analyzed, requiring a quantitative evaluation of the quality of each option in terms of its probability of success and degree of threat eradication. Response time for the blue forces is again a big issue here. Ultimately, numerical values regarding the number and quality of options serve as inputs to the actual-risk-calculation function where they are used to influence or modulate the inherent-threat value previously computed, while taking into account a quantitative assessment of the value of the protected unit itself. The prior threat value is thereby transformed into an actual-risk value that better reflects the tangible potential for danger. An entity that has been assigned a very high inherent-threat value could ultimately represent a very small risk if very easy to take care of (i.e., there are numerous, good-quality options for tackling the problem). In the middle of the spectrum, a moderate threat entity may nevertheless represent a high risk if there are no options available to counter it.

3.5.4.7 Situation Watch

As part of situation monitoring, *situation watch* observes the situation's evolution closely. It must pay attention to changes in various aspects of the situation and provide alerts concerning significant ones. It must keep awake and vigilant to maintain a state of alert and continuous attention for the SA process to prevent it from missing important entities, events, or activities. For example, monitoring the outcome of an engagement (e.g., kill assessment) in real time is important. The diagnostic subprocess must estimate the difference between the current perceived situation and the projected one. For example, given predictions of enemy intentions, situation watch must identify areas of interest that should be monitored for verification of those predictions [51]. Subsequent observed cues can then either be bizarre, irrelevant, unexpected, or absent (i.e., the expected is absent). Discrepancies become the basis for requests, through the process-refinement capability, for additional collection and for expanding the scope of the analytical evaluation [51]. In any case, the decision-maker should be alerted to inconsistent or unexpectedly absent activity.

3.5.5 Process Refinement

Process refinement seeks to optimize the overall SA process with respect to the dynamic goals and restrictions of decision-makers and the process requirements and constraints by supporting global control of both the information collection and the analysis-process resources [52]. Goal management is clearly required for the SA process. At any given time, decision-makers have many tasks in their queue in various stages of completion [42]. The urgency associated with individual tasks changes with time or the acquisition of new information. With constantly changing priorities, information needs are also constantly changing. The manner in which the "attention" of the SA process is employed in a complex environment with multiple competing cues is essential in determining which aspects of the situation will be processed to form SAW. The object of attentional allocation is to maximize the information content gleaned from the sources [51]. When not controlled or guided by a global strategy, data and information sources act as "vacuum cleaners," collecting, along with vital information, totally redundant, unnecessary, and unwanted data and information. The analysis of massive amounts of irrelevant data can severely burden both manual and automated SA processes. Hence, based on recognized information deficiencies and potentially available collection assets, process refinement generates prioritized information requirements that are sent to the collection manager [52].

Process refinement also plays an important role in assessing the quality of the data to be analyzed. In this regard, some assessment of enemy countermeasure activity must be performed to better quantify the confidence the decision-maker can place in the abstraction and assessment of the situation derived from the multisource data and information. Attempts to disrupt SA (i.e., situation-estimating countermeasures) involve concealment, cover, and deception (CC&D) and the creation of ambiguity [51]. The SA process is vulnerable to CC&D at each step. Finally, system awareness is especially important in complex, highly automated systems. Resource awareness is needed to keep track of the state of currently available resources, including both physical and human resources.

References

[1] Endsley, M. R., "Theoretical Underpinnings of Situation Awareness: A Critical Review," in M. R. Endsley and D. J. Garland, (eds.), *Situation Awareness Analysis and Measurement*, Mahwah, NJ: Lawrence Erlbaum, 2000, pp. 3–32.

[2] Garland, D. J., et al., *Theoretical Underpinnings of Situational Awareness: Towards an Objective Measure*, Technical Report No. CAAR-15498-91-1, Embry-Riddle Aeronautical University, Center for Aviation/Aerospace Research, 1991.

[3] Gilson, R. A., " Special Issue Preface," *Human Factors*, Vol. 37, No. 1, 1995, pp. 3–4.

[4] Carretta, T. R., D. C. Perry, and M. J. Ree, "The Ubiquitous Three in the Prediction of Situation Awareness: Round Up the Usual Suspects," in R. D. Gilson, D. J. Garland, and J. M. Koonce, (eds.), *Situational Awareness in Complex Settings*, Daytona Beach, FL: Embry-Riddle Aeronautical University Press, 1994, pp. 125–137.

[5] Hartel, C. E., K. Smith, and C. Prince, "Defining Aircrew Coordination: Searching Mishaps for Meaning," *6th International Symposium on Aviation Psychology*, Columbus, OH, April 1991.

[6] Redding, R. E., "Analysis of Operational Errors and Workload in Air Traffic Control," *Proceedings of the Human Factors Society 36th Annual Meeting*, Santa Monica, CA: Human Factors Society, 1992, pp. 1321–1325.

[7] Endsley, M. R., "Measurements of Situation Awareness in Dynamic Systems," *Human Factors*, Vol. 37, No. 1, 1995, pp. 65–84.

[8] Jones, D. G., and M. R. Endsley, "Sources of Situation Awareness Errors in Aviation," *Aviation, Space and Environmental Medicine*, Vol. 67, No. 6, 1996, pp. 507–512.

[9] Klein, G., "Analysis of Situation Awareness from Critical Incident Reports," in M. R. Endsley and D. J. Garland, (eds.), *Situation Awareness Analysis and Measurement*, Mahwah, NJ: Lawrence Erlbaum, 2000, pp. 51–71.

[10] Gaba, D. M., and S. K. Howard, "Situation Awareness in Anesthesiology," *Human Factors*, Vol. 37, No. 1, 1995, pp. 20–31.

[11] Cook, R. I., et al., "Evaluating the Human Engineering of Microprocessor-Controlled Operating Room Devices," *Journal of Clinical Monitoring*, Vol. 7, 1991, pp. 217–226.

[12] Moll van Charante, E., et al., "Human-Computer Interaction in Context: Physician Interaction with Automated Intravenous Controllers in the Heart Room," in H. G. Stassen, (ed.), *Analysis, Design, and Evaluation of Man-Machine Systems*, New York: Pergamon Press, 1993, pp. 263–274.

[13] Woods, D. D., J. F. O'Brien, and L. F. Hanes, "Human Factors Challenges in Process Control: The Case of Nuclear Power Plants," in G. Salvendy, (ed.), *Handbook of Human Factors*, New York: Wiley, 1987, pp. 1724–1770.

[14] Andersen, H. H. K., and G. Hauland, "Measuring Team Situation Awareness of Reactor Operators during Normal Operation: A Technical Pilot Study," *Proceedings of the 1st Human Performance, Situation Awareness, and Automation Conference*, Savannah, GA, October 15–19, 2000.

[15] Gugerty, L. J., and W. C. Tirre, "Individual Differences in Situation Awareness," in M. R. Endsley and D. J. Garland, (eds.), *Situation Awareness Analysis and Measurement*, Mahwah, NJ: Lawrence Erlbaum, 2000, pp. 249–276.

[16] Jenner, S. M., et al., "Automation and Situation Awareness in Transportation Accidents," *Proceedings of the 1st Human Performance, Situation Awareness, and Automation Conference*, Savannah, GA, October 15–19, 2000.

[17] Molloy, R. J., "Automation-Induced Loss of Situation Awareness in Accidents in Surface Transportation Accidents," *Proceedings of the First Human Performance, Situation Awareness, and Automation Conference*, Savannah, GA, October 15–19, 2000.

[18] Prince, C., and E. Salas, "Training and Research for Teamwork in the Military Aircrew," in E. L. Weiner, B. G. Kanki and R. L. Helmreich, (eds.), *Cockpit Resource Management*, San Diego, CA: Academic Press, 1993, pp. 337–366.

[19] Endsley, M. R., et al., *Modeling and Measuring Situation Awareness in the Infantry Operational Environment*, U. S. Army Research Institute for the Behavioral and Social Sciences, Infantry Forces Research Unit, 2000.

[20] Trafton, J. G., "Predicting Complex Information Using Qualitative Mental Models," *Proceedings of the First Human Performance, Situation Awareness, and Automation Conference*, Savannah, GA, October 15–19, 2000.

[21] Pew, R. W. "The State of Situation Awareness Measurement: Heading Toward the Next Century," in M. R. Endsley and D. J. Garland, (eds.), *Situation Awareness Analysis and Measurement*, Mahwah, NJ: Lawrence Erlbaum, 2000, pp. 33–47.

[22] Sarter, N. B., and D. D. Woods, "How in the World Did I Ever Get into That Mode: Mode Error and Awareness in Supervisory Control," *Human Factors*, Vol. 37, No. 1, 1995, pp. 5–19.

[23] Nowak, C., "On Ontologies for Higher-Level Information Fusion," *Proceedings of the 6th International Conference on Information Fusion*, Cairns, Australia, July 7–10, 2003.

[24] Jeannot, E., "Situation Awareness: Synthesis of Literature Search," *Eurocontrol*, ECC Note No 16/00 Project ASA-Z-EC, 2000.

[25] Dominguez, C., "Can SA Be Defined?" in M. Vidulich, et al., (eds.), *Situation Awareness: Papers and Annotated Bibliography* (AL/CF-TR-1994-0085), Wright-Patterson AFB, OH: Armstrong Laboratory, 1994, pp. 5–15.

[26] Adam, E. C., "Fighter Cockpits of the Future," *Proceedings of the 12th DASC, The 1993 IEEE/AIAA Digital Avionics Systems Conference*, Fort Worth, TX, October 1993, pp. 318–323.

[27] Gibson, J. J., *The Ecological Approach to Visual Perception*, Hillsdale, NJ: Lawrence Erlbaum, 1979.

[28] Marr, D., *Vision*, San Francisco, CA: W. F. Freeman, 1982.

[29] Dawson, M. R. W., *Understanding Cognitive Science*, Malden, MA: Blackwell, 1998.

[30] Adams, M. J., Y. J. Tenney, and R. W. Pew, "Situation Awareness and the Cognitive Management of Complex Systems," *Human Factors*, Vol. 37, No. 1, 1995, pp. 85–104.

[31] Endsley, M. R., "Toward a Theory of Situation Awareness in Dynamic Systems," *Human Factors*, Vol. 37, No. 1, 1995, pp. 32–64.

[32] Smith, K., and P. A. Hancock, "Situation Awareness Is Adaptive, Externally Directed Consciousness," *Human Factors*, Vol. 37, No. 1, 1995, pp. 137–148.

[33] Rosenthal D. M., "Introspection," in R. A. Wilson and F. C. Keil, (eds.), *MIT Encyclopedia of the Cognitive Sciences* (MITECS), Cambridge, MA: MIT Press, 1998.

[34] Prince, C., and E. Salas, "Team Situation Awareness, Errors, and Crew Resource Management: Research Integration for Training Guidance," in M. R. Endsley and D. J. Garland, (eds.), *Situation Awareness Analysis and Measurement*, Mahwah, NJ: Lawrence Erlbaum, 2000, pp. 325–347.

[35] Gibson, C. P., and Garrett, A. J., "Towards a Future Cockpit—The Prototyping and Pilot Integration of the Mission Management Aid (MMA)," in *AGARD-CP-478, Situational Awareness in Aerospace Operations*, Neuilly sur Seine, FR: Advisory Group for Aerospace Research and Development. (AD-A223939), 1990, pp. 7-1–7-9.

[36] Breton, R., and R. Rousseau, *Situation Awareness: A Review of the Concept and Its Measurement*, DREV-TR-2001-220, Valcartier, Canada, 2002.

[37] Endsley, M. R., "Design and Evaluation for Situation Awareness Enhancement," *Proceedings of the Human Factors Society 32nd Annual Meeting*, Vol. 1, 1988, pp. 97–101.

[38] Maggart, L. E., and R. Hubal, "A Situation Awareness Model," in S. E. Graham and M. D. Matthews, (eds.), *Papers from the 1998 Infantry Situation Awareness Workshop*, Washington, D.C.: U.S. Army Research Institute for the Behavioral and Social Sciences, 1998, pp. 19–28.

[39] Endsley, M. R., and W. M. Jones, *Situation Awareness, Information Dominance, and Information Warfare* (AL/CF-TR-1997-0156), Wright-Patterson Air Force Base, OH: United States Air Force Armstrong Laboratory, 1997.

[40] Salas, E., et al., "Situation Awareness in Team Performance: Implications for Measurement and Training," *Human Factors*, Vol. 37, No. 1, 1995, pp. 123–136.

[41] McGuinness, B., and J. L. Foy, "A Subjective Measure of SA: The Crew Awareness Rating Scale (CARS)," *Proceedings of the 1st Human Performance, Situation Awareness, and Automation Conference*, Savannah, GA, October 15–19, 2000.

[42] Endsley, M. R., and Garland, D. J., *Situation Awareness Analysis and Measurement*, Mahwah, NJ: Lawrence Erlbaum, 2000.

[43] Roy, J., "From Data Fusion to Situation Analysis," *Proceedings of the ISIF 4th International Conference on Information Fusion (Fusion 2001)*, Montreal, Quebec, Canada, August 7–10, 2001.

[44] Roy, J., S. Paradis, and M. Allouche, "Threat Evaluation for Impact Assessment in Situation Analysis Systems," *SPIE Proceedings*, Vol. 4729, *Signal Processing, Sensor Fusion, and Target Recognition XI*, Orlando, FL, April 1–5, 2002.

[45] Steinberg, A. N., C. L. Bowman, and F. E. White, "Revision to the JDL Data Fusion Model," *Proceedings of the Joint NATO/IRIS Conference*, Quebec City, Quebec, Ontario, October 19–23, 1998.

[46] Lambert, D. A., "Assessing Situations," *Proceedings of IEEE 1999 Information, Decision and Control*, Adelaide, SA, Australia, February 8–10, 1999, pp. 503–508.

[47] Lambert, D. A., "An Exegesis of Data Fusion," in L. Reznik and V. Kreinovich, (eds.), *Soft Computing in Measurement and Information Acquisition*, Studies in Fuzziness and Soft Computing, New York: Physica Verlag, 2001.

[48] Lambert, D. A., "Situations for Situation Awareness," *Proceedings of the ISIF 4th International Conference on Information Fusion (FUSION 2001)*, Montreal, Quebec, Canada, August 7–10, 2001.

[49] Karakowski, J. A., "Towards Visual Data Fusion," *NSSDF International Open Session*, Georgia Tech Research Institute, Atlanta, GA, April 1998.

[50] Klein, G., "Naturalistic Models C3 Decision-Making," in S. Johnson and A. Levis, (eds.), *Science of Command and Control: Coping with Uncertainty*, Fairfax, VA: AFCEA International Press, 1988, pp. 86–92.

[51] Waltz, E. L., and J. Llinas, *Multi-Sensor Data Fusion*, Norwood, MA: Artech House, 1990.

[52] Antony, R. T., *Principles of Data Fusion Automation*, Norwood, MA: Artech House, 1995.

[53] Hall, D. L., and J. Llinas, "An Introduction to Multisensor Data Fusion," *Proceedings of the IEEE*, Vol. 85, No. 1, January 1997.

[54] Llinas, J., and R. T. Antony, "Blackboard Concepts for Data Fusion Applications," *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 7, No. 2, 1993.

[55] Roy, J., N. Duclos-Hindié, and D. Dessureault, "Efficient Cluster Management Algorithm for Multiple Hypothesis Tracking," *SPIE Proceedings*, Vol. 3163, *Signal and Data Processing of Small Targets 1997*, San Diego, CA, July 29–31, 1997.

[56] Reid, D. B., "An Algorithm for Tracking Multiple Targets," *IEEE Transactions on Automatic Control*, Vol. AC-24, 1979, pp. 843–854.

[57] Roy, J., N. Duclos-Hindié, and D. Dessureault, "A Depth Control Pruning Mechanism for Multiple Hypothesis Tracking," *2nd International Conference on Information Fusion—Fusion'99*, Sunnyvale, CA, July 6–8, 1999.

[58] Blackman, S. S., *Multiple-Target Tracking with Radar Applications*, Norwood, MA: Artech House, 1986.

[59] Blackman, S. S., and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Norwood, MA: Artech House, 1999.

[60] Wilson, G. B., "Some Aspects of Data Fusion," in C. J. Harris and I. White, (eds.), *Advances in Command, Control and Communication Systems*, London, U.K.: Peter Peregrinus, 1987.

[61] Llinas, J., and D. L. Hall, *Situational Awareness for Intelligence, Theater Command and Control, and Contingency Planning Applications*, Course Notes, Technology Training Corporation, April 1995.

CHAPTER 4

Data- and Information-Fusion Models

Jean Roy, Dale Lambert, Frank White, Joseph Karakowski, John Salerno, and Mike Hinman

4.1 Introduction

Data and information fusion (DIF) has already received significant attention for military applications; it is still expected to play a crucial role in the next generation of support systems for aiding decision-makers in military and public-security operations. Indeed, DIF is a key enabler to meeting the demanding requirements of situation analysis and decision-making in command-and-control (C2) support systems. Among the many reasons for interest in this technology, data and information fusion:

- Provides extended spatial and temporal coverage, increased confidence, reduced ambiguity, improved entity detection, and so forth;
- Allows for the management of large volumes of information and the correlation of seemingly unrelated, overlooked, or deceptive information to present a coherent representation of an evolving situation to a decision-maker;
- Enables the commander to cope with the complexity and tempo of operations in modern, dynamic operational theaters.

This chapter reviews the main models that have been developed over the years to better understand and describe data and information fusion. Certainly, each one of these models has value as it provides particular insights into this important field. Hence, our purpose in describing them is not to argue for one or the other but to give the reader a good sense of these various perspectives, mainly to put the other two parts of this book in context.

The four models being considered are the JDL data-fusion model (currently the most widely accepted model of the data-fusion process), the visual data-fusion model, the unified data-fusion (or λJDL) model, and, finally, the situation-awareness reference model. The last three models actually resulted from different attempts to address some perceived deficiencies of the JDL model while, at the same time, considering the elements of Endsley's model of situation awareness.

4.2 The JDL Data-Fusion Model

The JDL is a U.S. DoD government committee overseeing U.S. defense-technology research and development. The data-fusion model developed and maintained by the JDL DFG) is the most widely used method for categorizing data-fusion-related functions [1].

The JDL fusion model is a functional model, motivated by confusion in the community over the many elements of fusion processes [2], was developed to provide a common frame of reference for fusion discussions, to facilitate the understanding and recognition of the types of problems for which data fusion is applicable, and to aid in recognizing commonality among problems and the relevance of candidate solutions.

Much of its value derives from the fact that identified fusion functions have been recognizable to human beings as a "model" of functions they were performing in their own minds when organizing and fusing data and information. It is important to keep this "humancentric" sense of fusion functionality since it allows the model to bridge the operational fusion community, the theoreticians, and the system developers [2]. The framework of the model has been useful in categorizing investment in automation and highlighting the difficulty of building automated processes that provide functionality in support of human decision processes, particularly at higher levels requiring reasoning and inference.

4.2.1 History of the JDL Data-Fusion Model

In early 1985, the Data Fusion Subpanel (DFS), now referred to as the Data-and-Information Fusion Group (DIFG), sensed the need for coordination and communication within the data-fusion community. To bring order to the community, the DFS focused its efforts on establishing a common language and frame of reference for the data-fusion process. This included the development of a data-fusion taxonomy, lexicon [3], and model. The initial lexicon defined *data fusion* as follows:

> Data fusion is a process dealing with the association, correlation, and combination of data and information from single and multiple sources to achieve refined position and identity estimates, and complete and timely assessments of situations and threats, as well as their significance.

The first model developed by the subpanel used the taxonomy to derive the data-fusion domain. It was a product-oriented model that used sensor data and information from other sources to perform analysis at the three "states," or stages, of data-fusion processing to develop assessments for the commander. Sensor-gathered data was the primary source, or input, for the correlation or tracker algorithm. The result of this process (i.e., state 1) was combined with a priori knowledge and other data to accomplish situation assessment (state 2). The results of the state 2 processing were analyzed in context with the threat database to derive state 3 and develop the assessment product for the commander.

This model was further refined to introduce the concepts of levels of processing in lieu of data-fusion states. The levels of processing addressed by the subpanel progressed from a heavy dependence on processing (level 1) to higher-level inference processes (levels 2 and 3). This revised model labeled situation assessment and threat assessment as the intelligence-analysis portion of the data-fusion process. It was clear that the composite track file and situation database, included in the first model, did not sufficiently portray the databases necessary to support the fusion process. Therefore, additional products and supporting databases were incorporated in the revised model.

In this model, tracking and correlation algorithms are used with other analytical statistical routines for object identification and position estimation. The results of this analysis are then combined with information on enemy patterns, structures, and other furnished intelligence using event-recognition and expert-systems techniques. The situation assessment is combined with indications and warning information to develop the threat assessment.

Another way of portraying this process, referred to as the preliminary data-fusion model, stresses an interactive process focusing on product and report generation. Once again the correlation and tracking segment of the process is stand-alone and results in the development of a tactical picture. This is combined with other tactical data to generate fusion products used for the development of situation assessments to support the decision-maker.

The subpanel's view, in 1985, of the data-fusion research-and-development process shows the use of the data-fusion model and ongoing research efforts as the foundation for the generation of a data-fusion research model. The research model is then used to develop a research-and-development plan for the services. Data-fusion research problems were identified through the decomposition of the data-fusion model and a survey of research efforts. This analysis was then used to develop recommendations for joint or coordinated efforts to identify new high-payoff programs and to further research and development in areas of technological gaps.

Work continued in 1985 and 1986, within the subpanel, to further refine the data-fusion model. A template approach was introduced to aid in the situation assessment process. Here, incoming data from multiple sources could be matched against a knowledge base containing information on known enemy operations, activities, and courses of action. The correct association of a template with an ongoing hostile operation would constitute a successful situation assessment.

In November 1986, Dick Baer, a DFS member, proposed a functional representation of the data-fusion process that further defined the level 1 processing as either single- or multiple-event oriented and as processing data either from single or multiple sources. Baer also introduced in this model the concept of intermodel and intramodel connectivity.

In November 1987, Richard Anthony introduced some additional thoughts and concepts for inclusion in the model. The need was identified for an expanded model that would incorporate all elements of the existing model but be flexible enough to accommodate other paradigms and models. The revised model would

- Support centralized or distributed control, including hierarchical;
- Support centralized or highly distributed fusion processing (physically and spatially);
- Support a centralized or highly distributed data or knowledge base;
- Support a centralized situation representation or distributed "picture of the battlefield;"
- Be hierarchically organized and recursively expandable;

- Provide a uniform framework for fusion of all intelligence;
- Allow the use of widely disparate data sources (radio frequency energy, finished image intelligence), local processing paradigms, specialized procedures, hardware, and so forth.

The Kramer/White paradigm for data fusion was subsequently introduced. This paradigm is similar in many ways to the original model. Inputs come from multiple sensors and sources, and the outputs support the decision-making process. It also illustrates the two-way flow of information between the current situation representation and sensor processing, situation interpretation, and threat assessment.

Through subpanel discussions with, and feedback from, the data-fusion community, refinement of the data-fusion model has continued. Weaknesses were identified in the model's strong focus on intelligence products and the difficulty of mapping the data-fusion domain to C2 models. In addition, the model previously portrayed data fusion as a sequential process; however, in some cases, threat assessment may be performed with outputs from level 1 processing, as well as with the level 2 situation-assessment inputs.

The DFS then discussed how the data-fusion domain fits within the C2 and intelligence models. As applied to the intelligence model, the data-fusion domain includes part of the collection cycle, the processing and analysis steps, and part of the production process. This paradigm of intelligence processing is valid for each of the intelligence nodes within the system. The results of the production process are disseminated to the C2 nodes. In the C2 node, the data-fusion domain is primarily oriented toward the understanding and evaluation of the data and the environment to develop products for the decision-making process. As in the intelligence model, data fusion occurs at each node within the C2 structure. Information resulting from data-fusion processing at one node may serve as input for processing at another node. At that time, the C2 and intelligence models were based on the stimulus, hypothesis, option and response (SHOR) paradigm developed by Dr. Joel Lawson of Naval Electronic Systems (NES).

Figure 4.1 shows the JDL DFS data-fusion model that was circulating around 1990. Around that time, the data-fusion domain still consisted of three levels of fusion.

Level 1 consists of single and multisource processing. This level involves tracking and attribute refinement achieved through sampling the external environment of interest. It deals primarily with the location and identification of enemy forces. In level 1, tracks and reports are fused into a tactical picture. If tracks cannot be developed through methods such as parametric characterization and correlation, then only the reports are used. Sensor reports may originate from the entity of interest, clutter, false alarms, or the background. In addition to sensor reports' inaccuracy, there is uncertainty associated with the vicinity, and one cannot associate the observed detections to the associated entities. Numerous processing concepts have been developed to address this uncertainty, including event recognition, tracking, identification, classification, association, and alignment.

Event recognition is research in the areas of information processing, entity recognition, and extraction techniques. *Tracking*, as defined in the DFS data-fusion lexicon, is the computational process dealing with the estimation of an object's

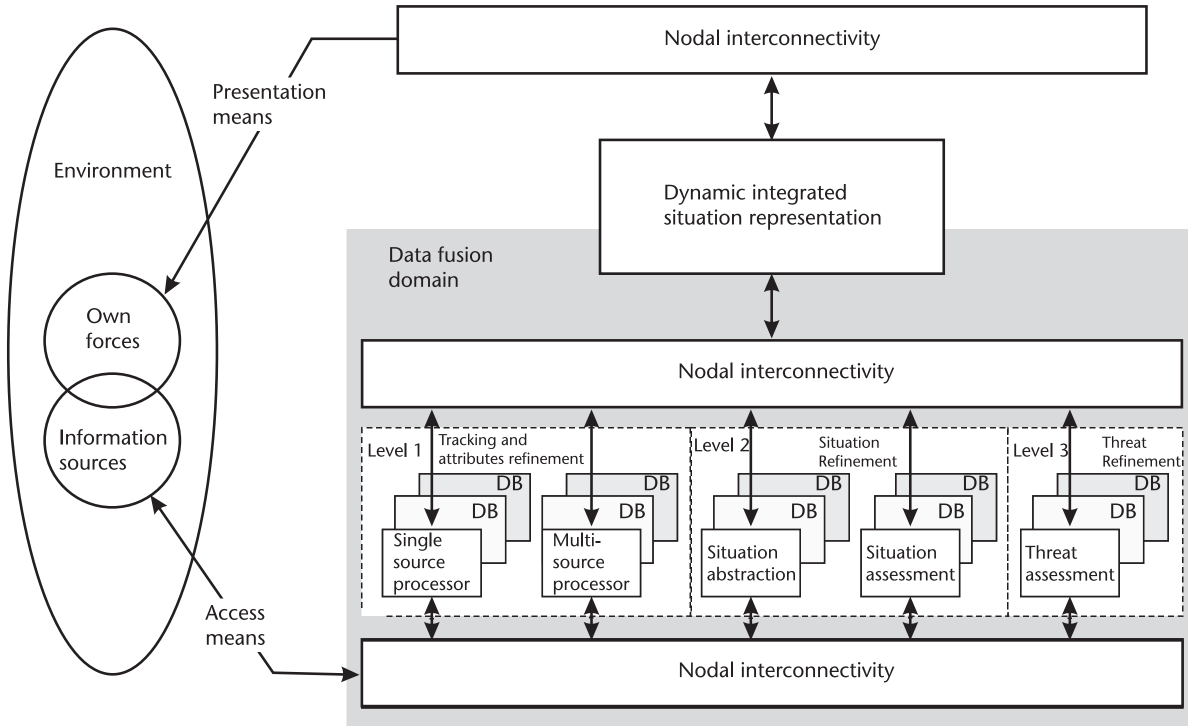


Figure 4.1 JDL data-fusion model circa 1990.

true position based on noisy observations (measurements) of it. Tracking may consist of *filtering* (estimating the position at the time of the latest observation), *smoothing* (estimating the position at a point in the past), and *prediction* (estimating the position at a point in the future). *Identification* techniques provide the true individuality of a person, object, or phenomena. *Classification* uses techniques for indexing and cataloging. *Association* is the process of generating, scoring, and deciding on hypotheses of which detections or measurements under consideration refer to the object (e.g., come from the same entity or should be associated with the same entity) and which refer to different objects. The alternative hypotheses may or may not be explicitly represented if, for example, decisions are made according to heuristic rules. Lastly, *alignment* is the processing of entity reports form sensors to achieve a common time base and spatial reference.

The tactical picture is then fused with referential and narrative data in an assessment and validation structure to arrive at information products. The referential data includes technical characteristics of the operating environment, such as force status, weapons characteristics, and geographical data. The narrative data fused with the tactical picture includes intelligence-scenario assessments and doctrine.

Level 2 processing involves *situation abstraction* and *situation assessment*. Situation abstraction is the process of constructing a tactical picture based on incomplete observations. For example, by abstracting bits and pieces of observations a more complete picture can be generated. Situation assessment provides a context-specific interpretation of the evolving situation. It is the process of interpreting the tactical environment in terms of the blue force's ability to engage the enemy effectively and includes indications and warnings of enemy intentions.

Level 3 processing provides threat assessment and other higher-level intelligence functions. It is a multiperspective (red, white, and blue force) process of developing estimates of the vulnerability of own forces based on enemy capability and intent.

The model is fed by information from all available sources, including organic sensor systems and national assets. Nodal interconnectivity allows information products to be passed from node to node on the same organizational level. Nodal intraconnectivity supports the passage of information or intelligence products between processing levels. This allows threat assessment to be accomplished using level 2 products from another node or, more directly, by using inputs from the information sources in the external environment. In fact, the model permits each of the functions in each processing level to be performed using the products of the other functions in the same type or from other types of nodes. The functions can be supported more directly using inputs from the external environment. Each function is also a processing level that accesses its own supporting database or other databases through interconnectivity and intraconnectivity.

The product of the data-fusion process is a dynamic, integrated situation representation drawn from the blackboard concept often used in artificial intelligence (AI). It shows the products or essential elements of information required by the C2 decision-maker. This is a key interface in the real world and the conceptual model avoids restricting the decision-maker to only certain data-fusion products. It supports using C2 needs or requirements as the driver of the fusion process.

Thus, the C2 decision-maker drives the problem and has full spectrum access to information products vital to the C2 process.

In 1991, the subpanel added a fourth level, process refinement, to the data-fusion model. This new level addresses the evaluation and control of the fusion process and provides guidance for acquiring new data. The concept of this level was introduced by the Office of Naval Technology (ONT) Data Fusion Strategy Panel in a report titled *Functional Description of the Data Fusion Process* in November 1991. In this document, the ONT panel stated that process refinement refers to the monitoring and evaluation of the ongoing fusion process to refine the process itself and to guide the acquisition of data to optimize results. Key functions of this process include:

- *Evaluations:* Evaluation of the performance and effectiveness of the fusion process to establish real-time control and long-term process improvements;
- *Fusion control:* Identification of changes or adjustments to the processing function within the data-forum domain, which many result in improved performance;
- *Source requirement processing:* Determination of the source-specific data requirement (i.e., identification of specific sensors or sensor data, qualified data, reference data) needed to improve the multilevel fusion products;
- *Mission management:* Recommendations for allocation and direction of resources (e.g., sensors, platforms, communications) to achieve overall mission goals.

Figure 4.2 shows the JDL DFS data-fusion model that was circulating in 1991.

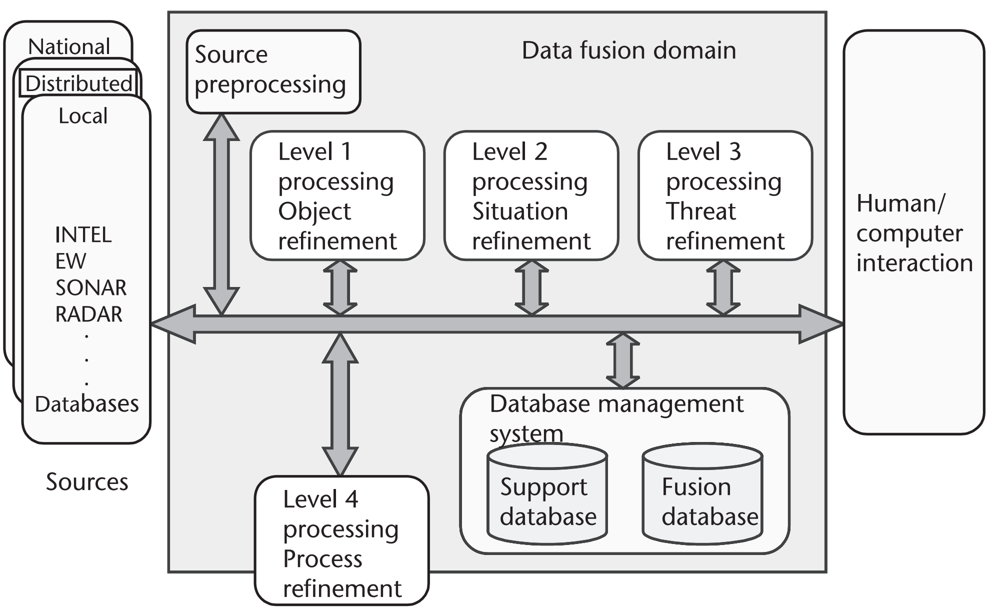


Figure 4.2 JDL data-fusion model circa 1991.

We consider Figure 4.3 from Llinas and Antony [4] a good illustration of the overall data-fusion process as per the JDL conception of Figure 4.2. It reflects the fact that in most defense applications, data-fusion processing tends to be hierarchical in nature due to the inherent hierarchies built into defense organizations and operations. As a result, the fusion process also progresses through a hierarchical series of inferences at varying levels of abstraction.

Figure 4.3 also suggests the iterative, continuous nature of these inference processes driven by the temporal character of the usual defense problem. Figure 4.4 from Antony [5] is also highly representative of this JDL conception of data fusion.

One may have noted that process refinement, that is, level 4 processing, is not entirely included in the data-fusion domain in Figure 4.2. This has been done on purpose. The term *resource management* (RM) can be used to imply the management of both system resources, which are used to provide input or support for processing functionality, and tactical resources, which are used to affect the environment to achieve some tactical or strategic goal. System resources include base systems (e.g., CPU, memory, and bandwidth) and software processes (e.g., algorithm choices). Tactical resources include weapons (e.g., missiles, guns, tracking and illuminating radars) and navigational mechanisms (e.g., control of vessel speed and direction). In this general sense, therefore, RM extends the level 4 processing implied by a strict adherence to the JDL data-fusion model of Figure 4.2.

The version of the model shown in Figure 4.2 has been used extensively for a while, until Steinberg, Bowman, and White [1] presented their effort to revise and expand this model once again, to facilitate, as they say, the cost-effective development, acquisition, integration, and operation of multisensor and multisource systems. This effort involved broadening the functional model and related

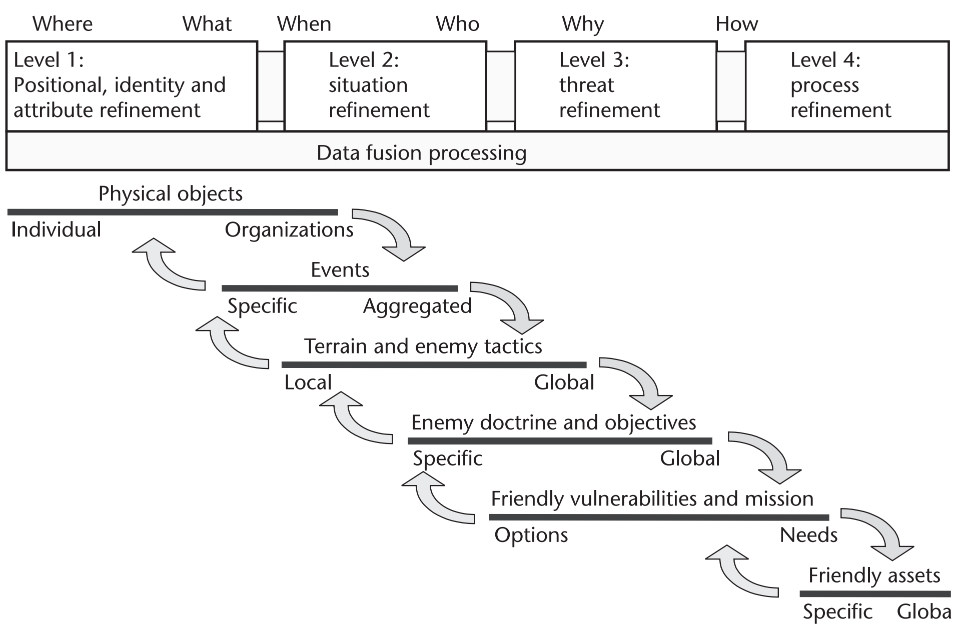


Figure 4.3 Multilevel/multiperspective inferencing. (*After:* [5].)

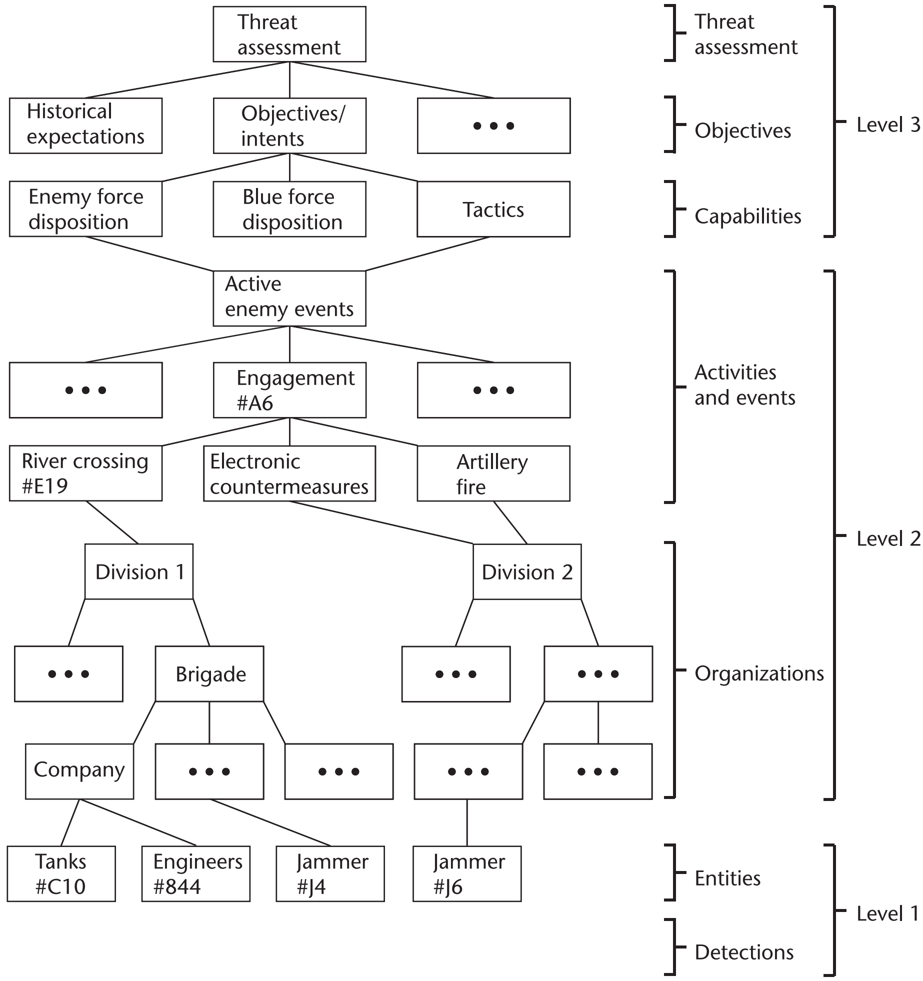


Figure 4.4 A view with multiple levels of abstraction. (*From:* [5]. © 1995 Artech House, Inc. Reprinted with permission.)

taxonomy beyond the original military focus and integrating the data-fusion tree architecture model for system description, design, and development. They introduce a level 0 (subobject data assessment) into the model and also include considerations of informational and perceptual states, in addition to the traditional physical state, that are of interest and can be useful if the job is to estimate the state of a human being (or any other sentient being). The last major part of that paper described the need for an approach to standardizing an engineering design methodology for data-fusion processes, citing the prior works of Bowman [6], Steinberg and Bowman [7], and Llinas et al. [8], which elaborated engineering guidelines for data-fusion processes. This version of the JDL model is potentially the most quoted version in the contemporary literature on data and information fusion.

Around 1999, Erik Blasch proposed the addition of a sixth level to the JDL model, level 5, built around the notion of user refinement. This level is discussed in Blasch and Plano [9], and their definition is as follows:

Level 5—User Refinement (an element of Knowledge Management): adaptive determination of who queries information and who has access to information (e.g., information operations) and adaptive data retrieved and displayed to support cognitive decision-making and actions (e.g., altering a sensor display).

Figure 4.5 shows their proposed DIFG fusion model, a "JDL-User" model.

Blasch and Plano's view is that process refinement, level 4 of the JDL data-fusion model, covers a broad spectrum of actions, such as sensor management and control; a limitation of level 4 is the purpose of control, be it for user needs or system operation. Level 5, user refinement, is a modification of the JDL model that distinguishes between machine-process refinement and user refinement [9] of either human control actions or the user's cognitive model. In many cases, fusion research concentrates on the machine and fails to take full advantage of the human not only as a qualified expert to refine the fusion process but as a customer for whom the fusion system is designed. Without user refinement, fusion is incomplete and inadequate, and the user neglects its worthiness.

To capture user capabilities, Blasch and Plano [9] explore the concept of user refinement through decision and action based on situational leadership models. They develop a Fuse-Act Situational User Refinement (FASUR) model that details four refinement behaviors (neglect, consult, rely, and interact) and five refinement functions (planning, organizing, coordinating, directing, and controlling). Process refinement varies for different systems and different user-information needs. By designing a fusion system with a specific user in mind, vis-à-vis level 5, a fusion architecture can meet user information needs for varying situations, extend user sensing capabilities for action, and increase the human-machine interaction.

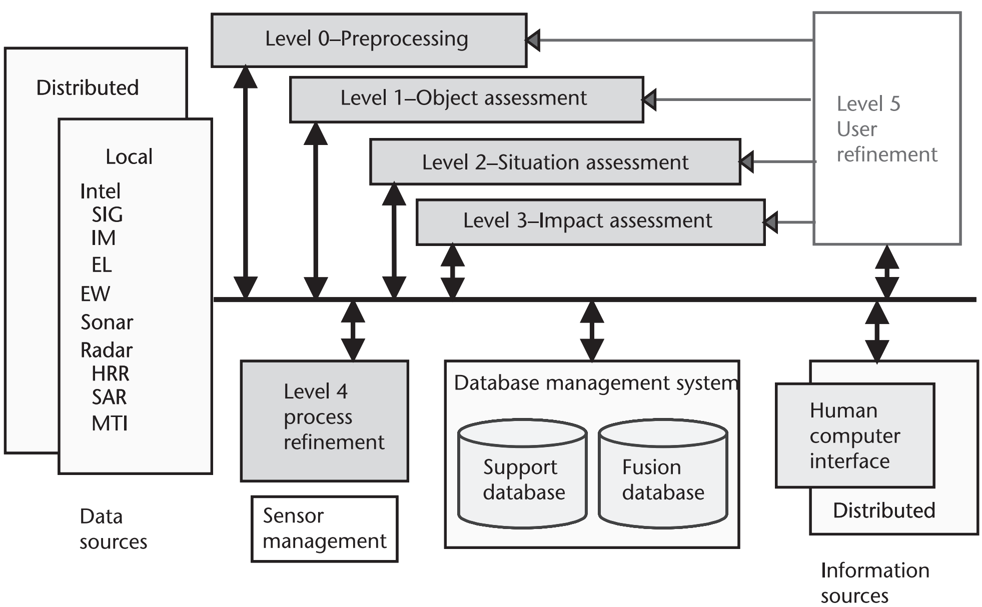


Figure 4.5 Proposed DIFG fusion model. (*From:* [9]. © 2003 SPIE. Reprinted with permission.)

Finally, another recent effort to continue the refinement and extension of the JDL data-fusion model is that of Llinas et al. [2]. They first state in their paper that the JDL model has not been reviewed in accordance with (1) the dynamics of world events, and (2) the changes, discoveries, and new methods in both the data-fusion research-and-development community and related information technologies. They thus make proposals, for community discussion, regarding improvements to the understanding of internal processing within a fusion node, extending the model to include:

- Remarks on issues related to quality control, reliability, and consistency in data-fusion processing;
- Assertions about the need for coprocessing of abductive, or inductive, and deductive inferencing processes;
- Remarks about the need for, and exploitation of, an ontologically based approach to data-fusion process design;
- Extensions to account for the case of distributed data fusion (DDF).

Along this line of thought, they discuss, among other things:

- Nodal and fusion-level processing (including remarks on the fusion "levels" in the current model);
- Interlevel information exchange;
- Adjudication, conflict resolution, and belief change (regarding within-level and interlevel processing);
- Fusion-node or system-level output processing;
- Architectural issues in DDF;
- Local and network fusion algorithms.

They consider their paper an offering about issues and functions considered important for any generalized data-fusion model description for modern-day applications, as well as a possible input to what they hope will be a communitywide effort to establish and control a community-standard model.

Certainly, the JDL model will continue to evolve, especially through technical interchange with members of the data-and-information-fusion community at annual symposiums, special interest group meetings, and other technical forums.

4.2.2 Description of the Contemporary JDL Data-Fusion Model

The history provided in Section 4.2.1 shows that Steinberg, Bowman, and White's version [1] of the JDL model is potentially the most quoted version in the contemporary literature on data and information fusion. It is briefly described here.

In their paper, Steinberg, Bowman, and White begin by revisiting the basic definitions of data fusion. They then propose the following concise definition of data fusion: "Data fusion is the process of combining data to refine state estimates and predictions."

They also discuss the JDL distinction between fusion levels, illustrated in Figure 4.6, which provides a way of differentiating between data-fusion processes that relate to the refinement of "objects," "situations," "threats," and "processes."

The levels in the JDL model were originally the result of a partitioning scheme based on the combined and interdependent effects of changing levels of abstraction and changing levels of problem-space complexity [2].

In their paper, Steinberg, Bowman, and White differentiate the levels first on the basis of types of estimation processes that typically relate to the type of entity for which state is estimated. If a process involves explicit association in performing state estimates (usually, but not necessarily the case), there is a corresponding distinction among types of association processes. They show the sorts of assignment matrices typically formed in each of these processing levels.

The level definitions in Steinberg, Bowman, and White [1] are as follows:

- *Level 0, subobject data assessment:* Estimation and prediction of signal/object observable states on the basis of pixel-/signal-level data association and characterization. Level 0 assignment involves hypothesizing the presence of a signal (i.e., of a common source of sensed energy) and estimating its state.
- *Level 1, object assessment:* Estimation and prediction of entity states on the basis of observation-to-track association, continuous state estimation (e.g., kinematics) and discrete state estimation (e.g., target type and ID). Level 1 assignments involve associating reports (or tracks from prior fusion nodes) into association hypotheses, for which we use the convenient shorthand "tracks." Each such track represents the hypothesis that the given set of

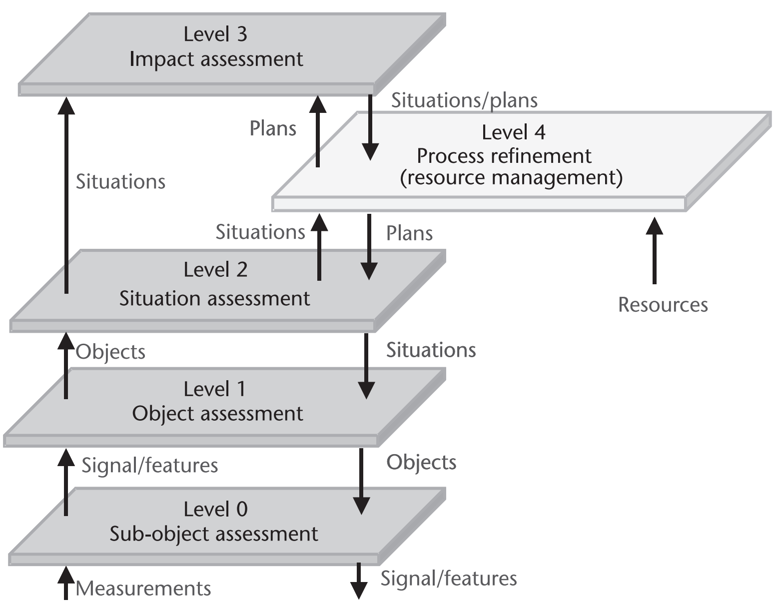


Figure 4.6 The JDL data-fusion model [1].

reports is the total set of reports available to the system referencing some individual entity.

- *Level 2, situation assessment:* Estimation and prediction of relations among entities to include force structure and cross-force relations, communications and perceptual influences, physical context, and so forth. Level 2 assignment involves associating tracks (i.e., hypothesized entities) into aggregations. The state of the aggregate is represented as a network of relations among its elements. Any variety of relations is considered—physical, informational, perceptual, organizational—given that it is appropriate to the given system's mission. As the class of relationships estimated and the numbers of interrelated entities broaden, Steinberg, Bowman, and White tend to use the term *situation* for an aggregate object of estimation.
- *Level 3, impact assessment:* Estimation and prediction of effects on situations of planned, estimated, or predicted actions by the participants, to include interactions between action plans of multiple players (e.g., assessing susceptibilities and vulnerabilities to estimated or predicted threat actions, given one's own planned actions). Level 3 assignment is usually implemented as a prediction function, drawing particular kinds of inferences from Level 2 associations. Level 3 fusion estimates the "impact" of an assessed situation (i.e., the outcome of various plans as they interact with one another and with the environment). The impact estimate can include likelihood and cost or utility measures associated with potential outcomes of a player's planned actions.
- *Level 4, process refinement:* Adaptive data acquisition and processing to support mission objectives. Level 4 processing involves planning and control, not estimation. Level 4 assignment involves assigning tasks to resources.

An extensive description of the version of the JDL model discussed in this section, along with discussions of many peripheral issues that concerns this perspective of data and information fusion, can be found in [10]. Moreover, as previously mentioned, many other aspects have also been discussed more recently in [2], around an effort to continue the refinement and extension of the JDL data-fusion model.

4.2.3 Fusion Versus Reasoning or Inference

Not all of the situation elements of interest to a given decision-maker are directly observable with the typical data and information sources currently available. This is especially true of highly abstract types of situation elements (e.g., enemy intent) and also of the relationships between situation elements. Those aspects of interest that cannot be directly observed must be inferred, that is, derived as a conclusion from facts or premises or by reasoning from evidence. This is an essential aspect of information fusion and situation analysis that will need a lot more attention in the future. Figure 4.7 shows both notions, that is, that of a fusion processing node and that of an inference processing node. This figure also illustrates the distinction between redundancy fusion and complementary fusion.

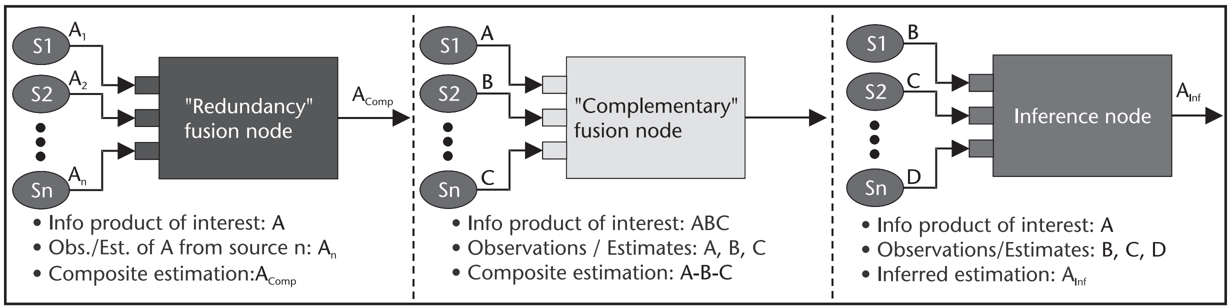


Figure 4.7 Fusion (redundancy and complementary) and reasoning or inference nodes.

4.2.4 Data-and-Information-Fusion Node

The concept of a processing node used in the previous section has been introduced in the framework of the JDL fusion model. Steinberg, Bowman, and White [1] presented this concept as shown in Figure 4.8. According to this fusion-node paradigm, the node processes the data and information provided by the sources (or other prior fusion nodes) at the input to produce a composite, high-quality version of information products of interest to the users (or to other, subsequent fusion nodes) at the output.

Any data-and-information-fusion node, whatever the fusion level, contains three main subprocesses: fusion, association, and alignment. The means for implementing these functions, and the data and control flow among them, will vary from node to node and from system to system. Nonetheless, this node paradigm has proven to be a useful model for characterizing, developing, and evaluating fusion systems.

The fusion per se actually happens in the "State estimation and prediction" box. The word "state" here refers to the state (actual or estimated) of any situation elements of interest to the users (e.g., entity position, velocity, identification, behavior, intent, threat value). However, although one can know very well how to combine (or fuse) input elements from different sources to obtain a composite product, data and information alignment and association have to be achieved first before the fusion can be performed.

Data and information related to an entity, a battlefield event, a group, and so forth, will often be reported independently via a multiplicity of sensors or sources, each differing in coverage area, spectrum, resolution, response time, and observable sensed. Alignment, or common referencing, is the processing of input reports to achieve, among other things, a common time base and a common spatial reference [11]. The alignment subprocess must remove any positional or sensing geometry and timing effects from the data and information [12]. The subprocess also transforms source data into a consistent set of units and coordinates for further processing [11].

Association is a basic subprocess necessary to determine which data and information elements at the input of the process associate to which elements currently being maintained in the situation representation (i.e., the situation model) being maintained by the processing node. Association is necessary to deal with the uncertainty attached to the situation elements. A classic example is determining whether entity data, which have been reported by different sources, represent the same

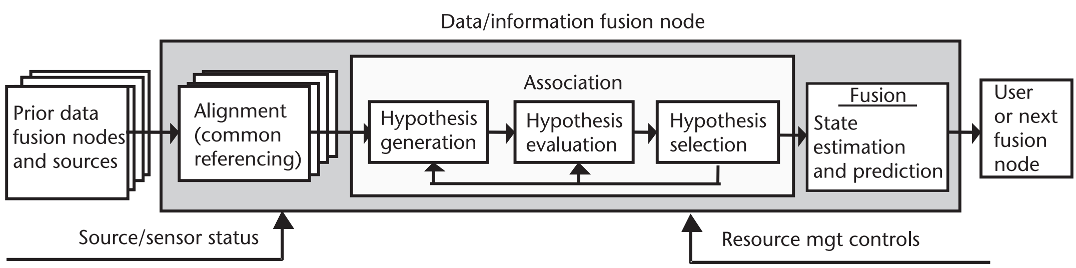


Figure 4.8 Any data-and-information-fusion node. (*After:* [1].)

entity or different entities; in this case, one talks about data-origin uncertainty management.

The diagram in Figure 4.9 generalizes the concept of a fusion node to that of a situation-analysis node. It also represents a more precise version of such a processing node as it includes the notion of a stored representation of the current knowledge of the world (and its associated management), as well as the notion of ancillary knowledge/information/data (KID) sources that are necessary to the execution of the fusion and inference processes.

In particular, Figure 4.9 is a better representation of a situation-analysis node as it allows for the easy illustration of the steps and timing for the processing of a user request for information on the current situation (or for a projection of it), as in Figure 4.10, or for the processing of new input elements (i.e., a situation update), as in Figure 4.11.

4.2.5 KID Processing Tree

A very important step in the development of any situation analysis or information-fusion system is the establishment of the KID processing tree, which makes explicit

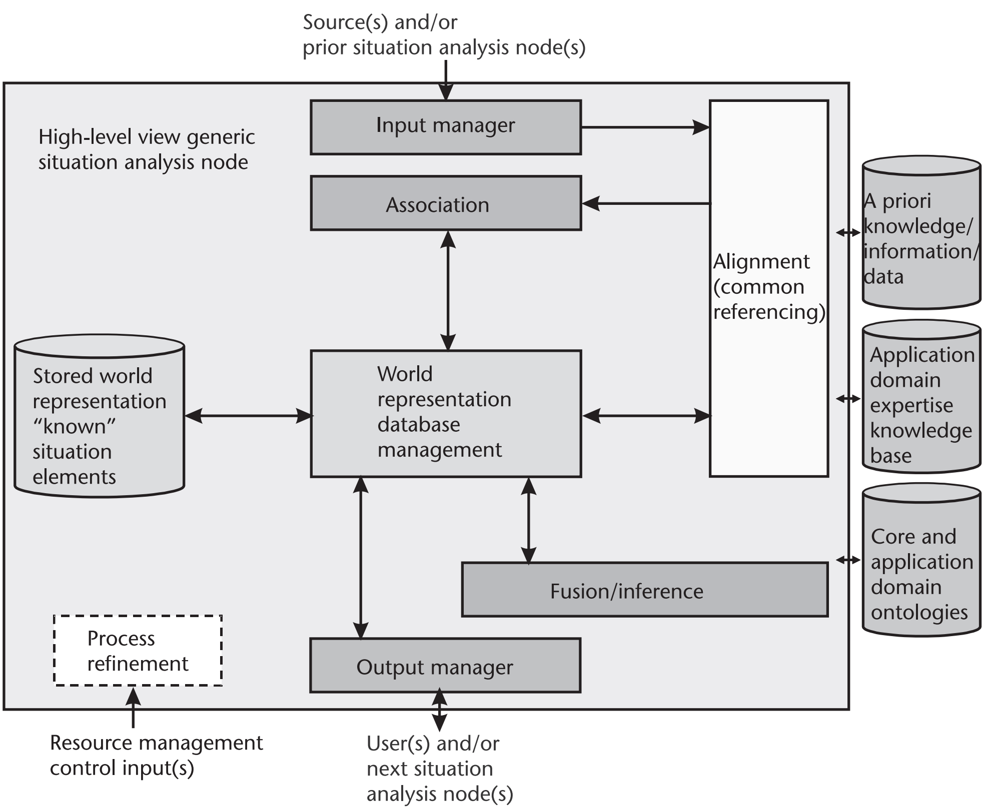


Figure 4.9 Any situation-analysis node.

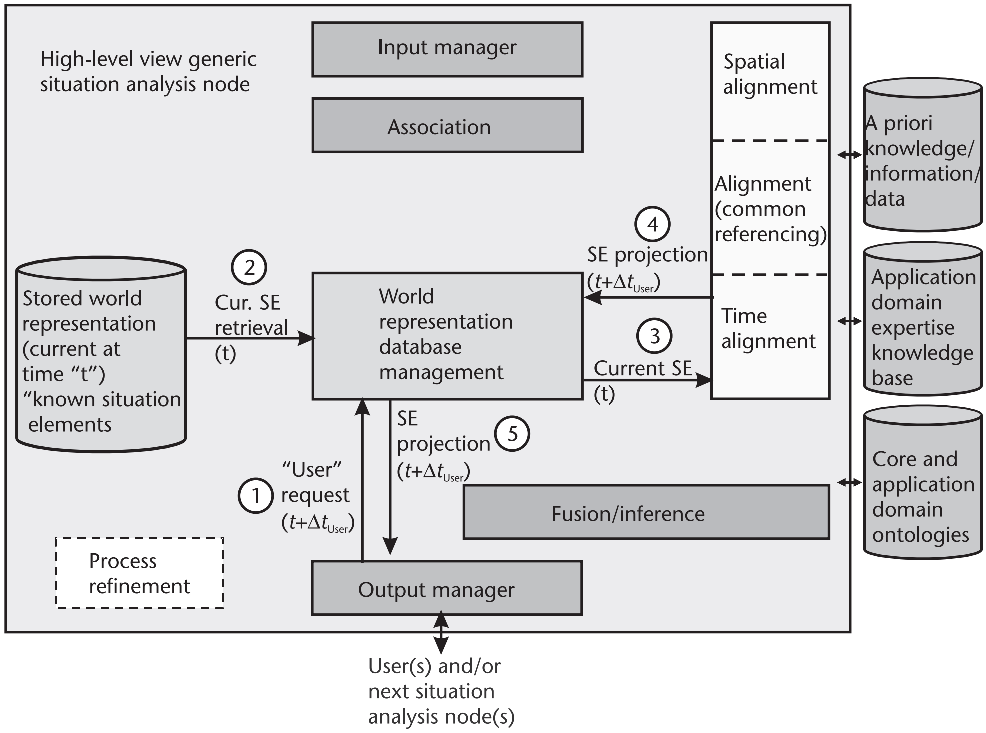


Figure 4.10 "User" request for "known" situation elements.

and illustrates the decomposition (or partitioning) of the situation-analysis processing into a network of multiple interconnected nodes, going from the sources to the different products relevant and useful to (i.e., required by) the decision-makers. Examples of such processing trees are shown in Figure 4.12. There are single-source processing (SSP) nodes and multiple-source processing (MSP) nodes. An important aspect is the notion of specific interfaces (I) to the sources. The challenge here is to encapsulate most of the specificities of an application domain into interfaces that perform the necessary processing to accept the data and information from the very specific sources on one side and to provide standardized inputs to generic situation-analysis and information-fusion engines on the other side.

The duality between data fusion and resource management has often been highlighted by the JDL community [1]. This duality can be extended to include the architectures and functionality of data fusion and resource management, leading to the notion of a resource-management node. Figure 4.13 from Steinberg, Bowman and White [1] shows some highly integrated fusion or management systems as part of a multifaceted, spatially distributed, sensor or response system. Such solutions are facilitated by the formal duality between data fusion and resource management, resulting in the analogous processing node paradigms for the two functions. Note that as one moves to the right of interlaced fusion or management trees, as depicted in Figure 4.13, the fusion or management node pairs generally operate with broader perspectives and slower response times.

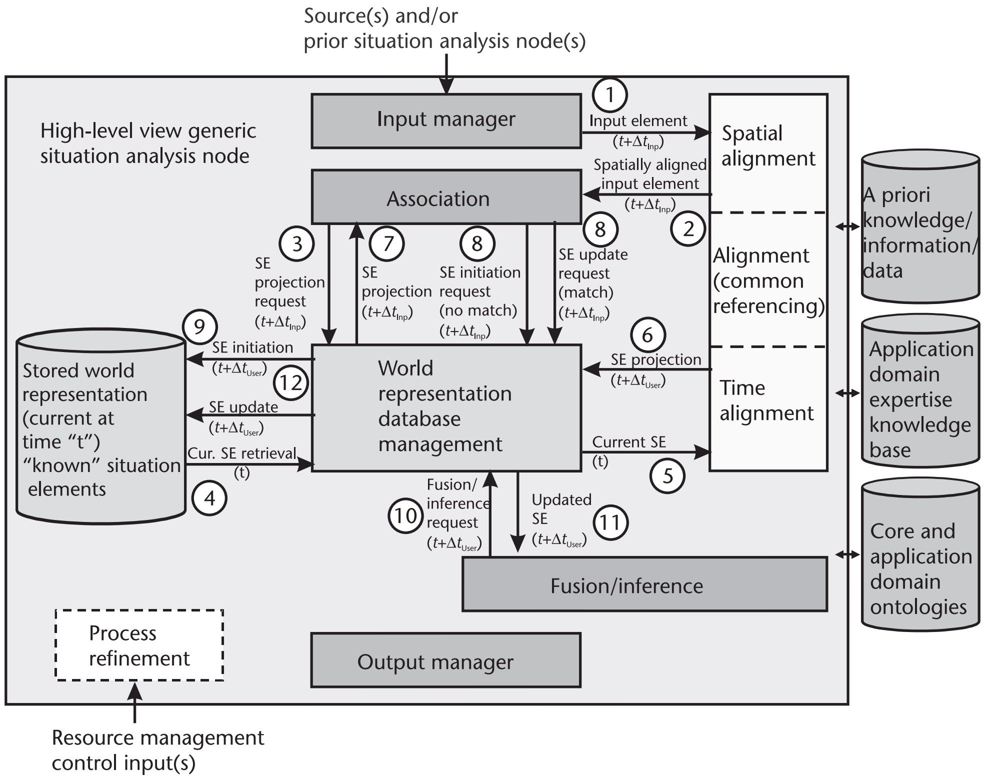


Figure 4.11 Processing of new input elements (situation update).

4.3 The Visual Data-Fusion Model

The basic visual data-fusion (VDF) model is an extension of the JDL data-fusion model as proposed by Karakowski [13] from the U.S. Army RDECOM CERDEC I2WD. It addresses several shortcomings of prior models:

- It maximizes relevant information with minimal information displayed.
- It tailors information-fusion system capabilities to be used by all skill levels of users, yet able to provide increasingly sophisticated problem queries.
- It directly relates to user's needs by responding to his or her personal perception of the problem situation; that is, it is a problem-driven system.

The basic VDF model embodies several premises, including the following:

- Information fusion is a creative problem-solving process, and the human is its central participant.
- The primary value of information results visualized by the human as a result of the fusion process is to assist the human in fuller perception of the problem and possible avenues toward a solution.
- Information presented to the human should be maximally relevant to the problem, while simultaneously minimal in number and dimensionality; in

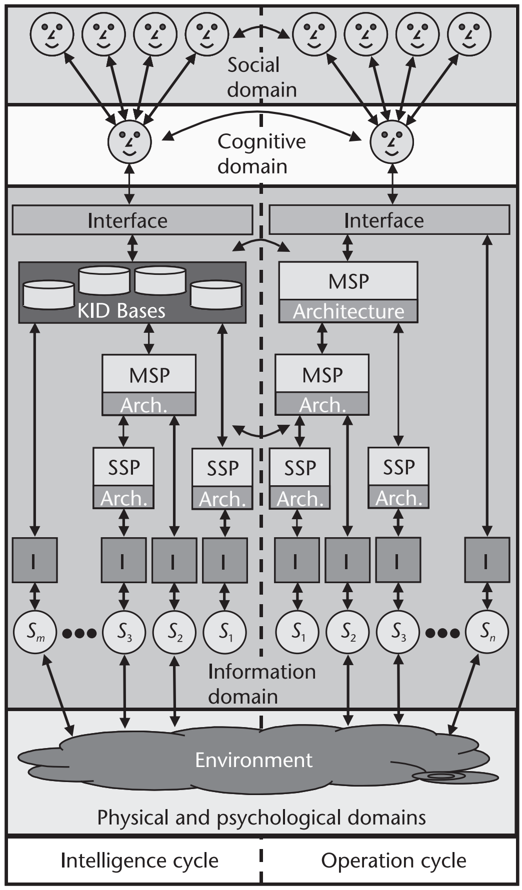


Figure 4.12 KID fusion- or inference-processing trees.

short, it should provide the least information needed to solve the problem and not overwhelm the user with too much data. As part of this, the model exclusively uses imagery as the perceptual transport for user visualization.

The basic VDF model, shown in Figure 4.14, structurally is an extension to the JDL model, but adds a human participant integrally to it. This provides for definition of associated verbal, conceptual, experiential, and visual interfaces. In addition, there is a specific data-control interface for representing the relevancy of the information requests to the fusion processes. Here, human problem input is filtered using a complex relational representation of the associated concepts, such as an ontology. Basic VDF model fusion levels are depicted as increasingly higher levels of generalization of the problem situation, and learning or problem-solving experience is a visual feedback into the model.

4.3.1 Visual Situation Awareness

Visual situation awareness (VSAW) is a functional engineering model of SAW that utilizes individual basic VDF models as building-block elements. It extends the

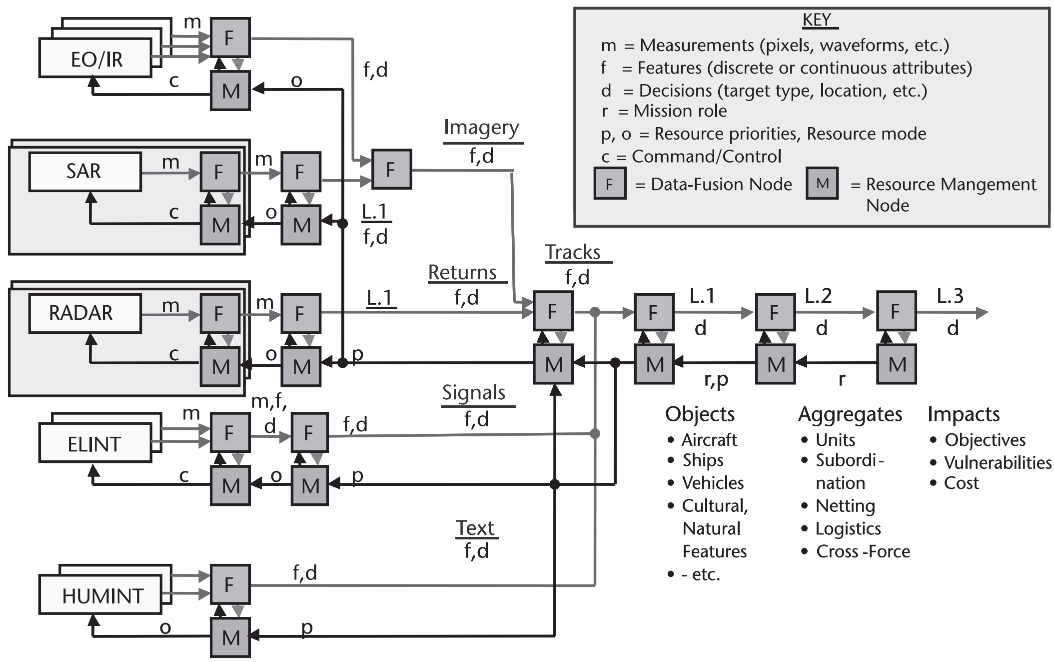


Figure 4.13 Integrated data-fusion or resource-management trees [1].

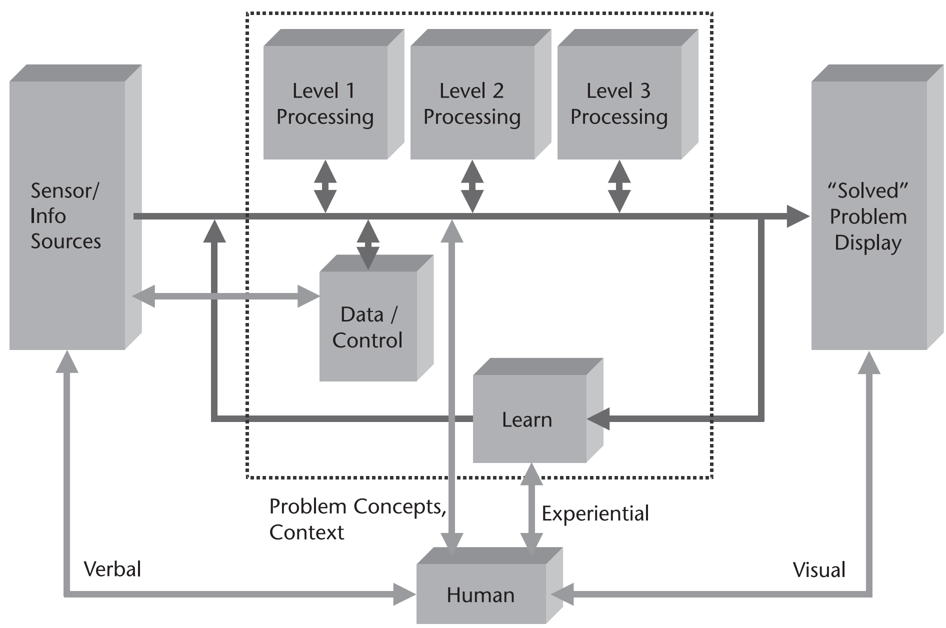


Figure 4.14 Visual data-fusion model.

basic functionality of a single VDF model by networking these blocks into a larger conceptual or contextual information interconnect for real work problem understanding and solution. The building block is shown in Figure 4.15.

How is SAW generated? The VSAW model provides one approach using a perceptual visualization of a continuously time-updated answer to specific problem requests. The problem requests are formulated within specific physical contexts, using individual perceptual biases, experiences, and other differences to make a decision, understand, solve, or otherwise better interpret various information sets as potential solutions to the stated problems. SAW then becomes a time-varying, complex visualization of multiple concepts within multiple contexts, with varying data and information sources.

The most important informational components of the model are: (1) problem requests as *concepts*, and (2) their underlying *context*. A concept is loosely defined as the generic problem to be solved. The VDF model can be considered the facilitator of this process, assisting the human-machine complex in creative problem solving. The concept or problem request typically takes the form of a question that represents a statement of a problem to be solved, for instance, What and where are the current threats near our position? or What can threaten our rear supply or reserves? Results of linguistic queries provide single-time snapshots or a continuous or time-limited answer to the problem or query. In Figure 4.16 illustrates, within a basic geographical context, a single problem or VDF system query for the question, What or where is any activity which threatens my position? This is the format of a single VDF model within a single context. It illustrates one of the simplest SAW architectures of the VSAW model, that which solves a single-query, single-context problem in a single instance or over time.

The framework of Figure 4.16 can be implemented by a single VDF model as shown in Figure 4.17. The model breaks out the information-control functions of "relevant information requests," interprets each specific concept and context, and

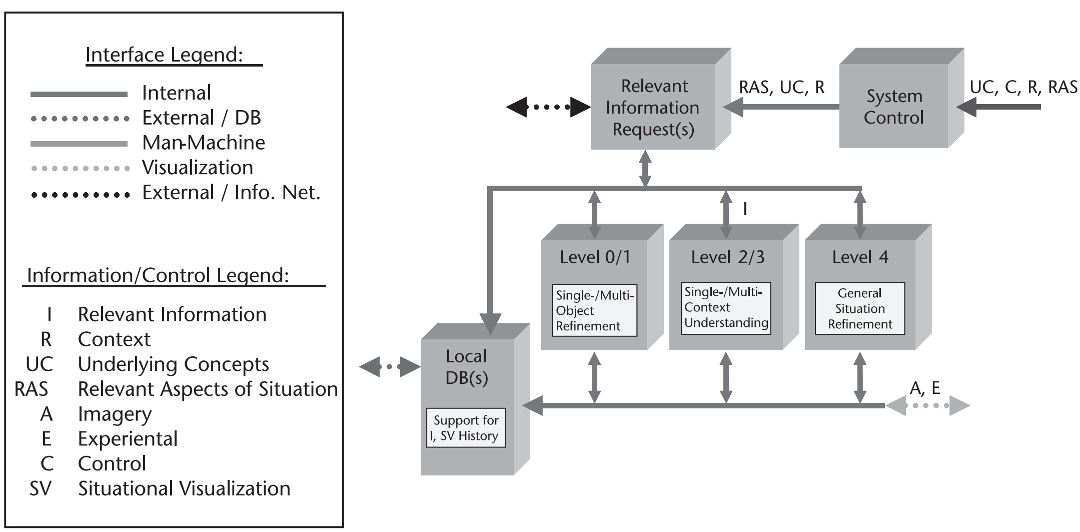


Figure 4.15 VSAW building block.

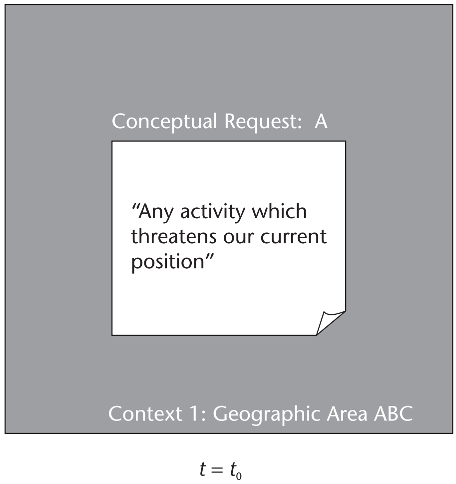


Figure 4.16 Situation-understanding requests—single concept within a single context at a single time.

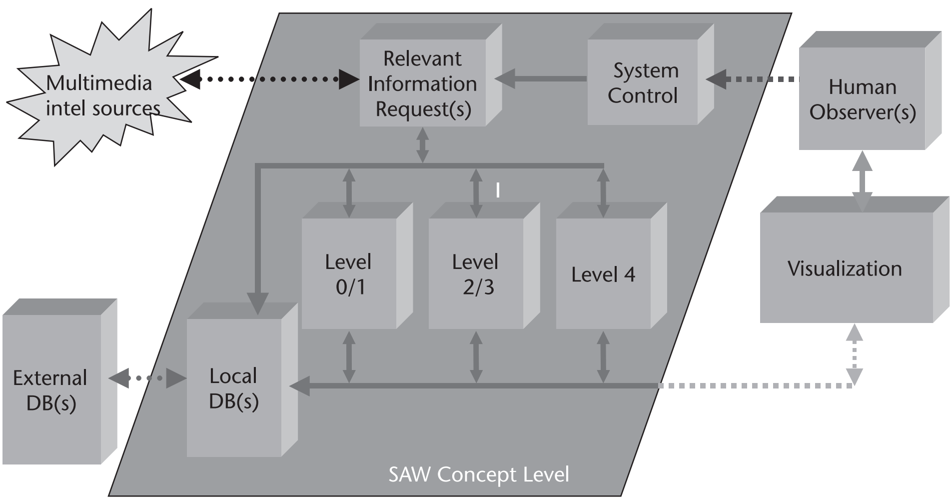


Figure 4.17 Single VDF element as a single-concept, single-context, situation-understanding subsystem at a single time.

continuously returns relevant information to the various fusion levels. Additionally, the local and external databases are broken out. The local database supports "learning" storage, and the external database provides a playback option. System control includes the standard data-processing functions, as well as interactivity for context or concept formulation and feedback.

4.3.2 Distributed Visual Data-Fusion Processes

The single-context SAW can initially be extended to multiple concepts by adding additional concepts within the current context. For example, Figure 4.18 illustrates four concepts, each with a single (geographical) context.

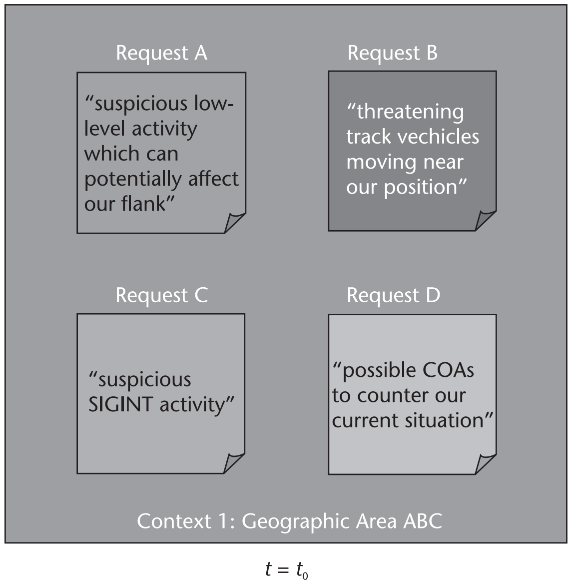


Figure 4.18 Situation-understanding requests—multiple concepts within a single context at a single time.

In this example, within a geographical context, the following "problems" are stated:

- Any suspicious enemy activity that can affect our flank;
- Threatening track vehicles near our position;
- Suspicious signal intelligence activity;
- Possible courses of action to counter our current situation.

Note that not all of these requests are "well formed," but all make sense to the problem formulator within his or her context, and hopefully he or she is in the best position to make use of the answers. Figure 4.19 shows the implementation using the four interconnected building blocks of Figure 4.15. Each of the concepts within the common context is loosely connected through the external database and provides parallel "answers" to each of the stated concepts.

The next step in generalizing the SAW model is to add multiple contexts, each with one or more concepts. This is illustrated conceptually in Figure 4.20, while its implementation using the VDF model is shown in Figure 4.21. Clearly, the complexity has grown, both in number of interfaces and number of concepts. The external interfaces, however, remain the same; only the internal interfaces have become more complex.

Finally, we can illustrate the multicontext, multiconcept structure to show the development of the problem over time (see Figures 4.22 and 4.23). This concept or context environment illustrates the dynamic nature of SAW, which can evolve over time.

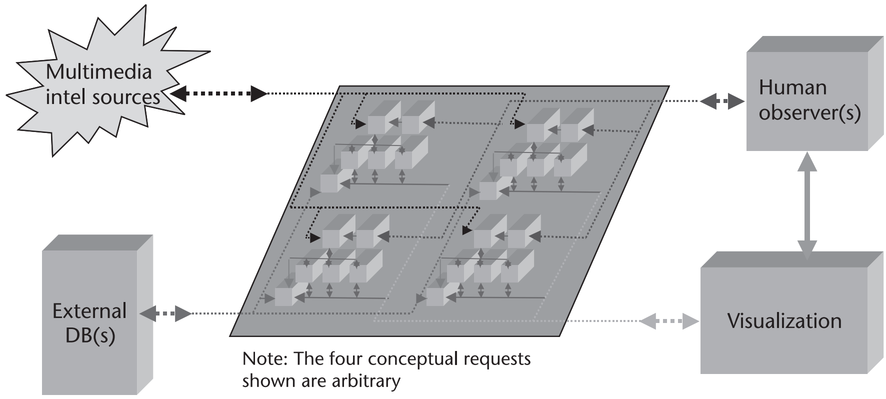


Figure 4.19 Multiple VDF elements as a multiple-concept, single-context, situation-understanding subsystem at a single time.

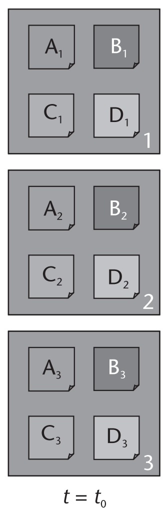


Figure 4.20 Situation-understanding requests—multiple concepts within multiple contexts at a single time.

4.4 The Unified Data-Fusion (λJDL) Model

The terms *situation awareness* (SAW), *common operating picture* (COP), and *data fusion* (DF) are often conflated. Working for the Defence Science and Technology Organisation (DSTO) in Australia, Lambert [14–16] has proposed a unifying conceptualization as a basis for a more mature, strategic foundation for understanding data fusion. Lambert initially called this the unified data-fusion (UDF) model. More recently, he changed the name of this unifying conceptualization to the λJDL model of data fusion [17]. It is described in detail in this section.

4.4.1 A View of Situation Awareness

Endsley's account of SAW is probably dominant in academic military circles. As the perception, comprehension, and projection components of SAW characterize

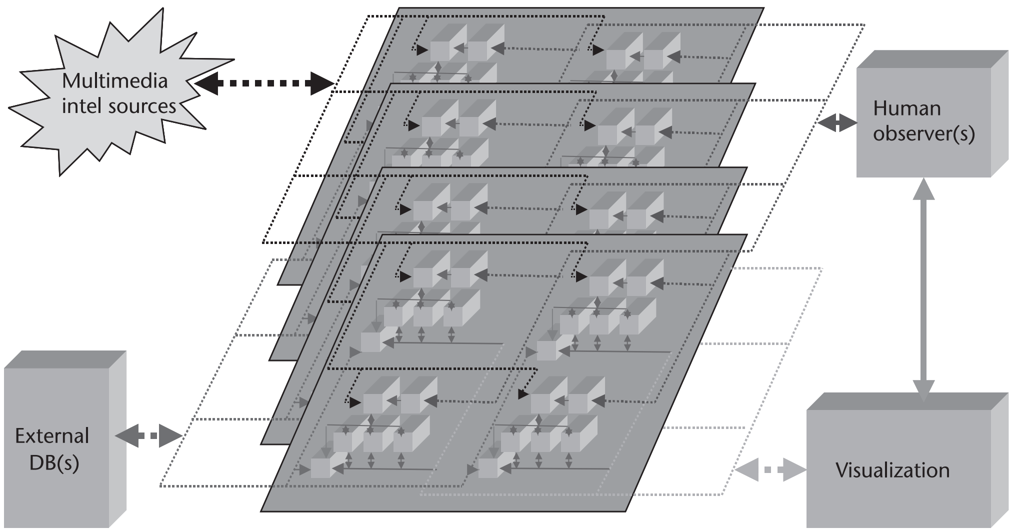


Figure 4.21 Multiple VDF elements as a multiple-concept, multiple-context, situation-understanding subsystem at a single time.

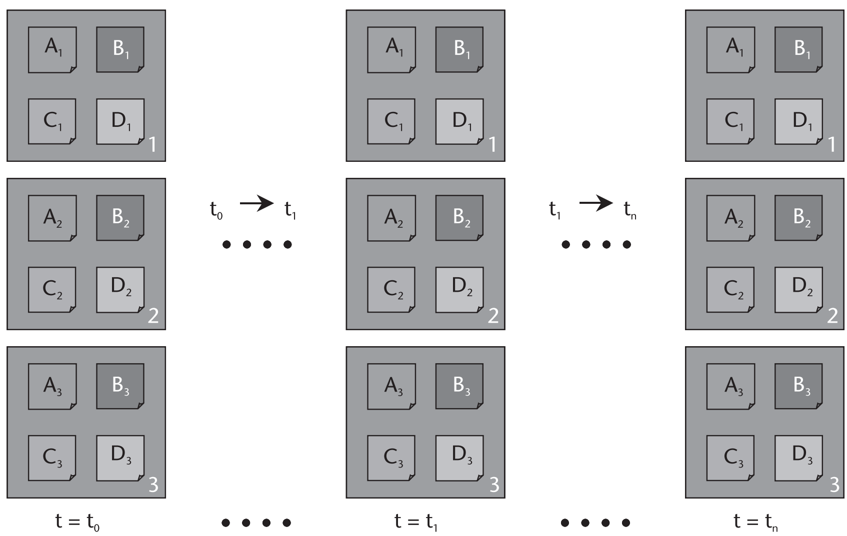


Figure 4.22 Situation-understanding requests—multiple concepts within multiple contexts at discrete, multiple times.

mental attributes, SAW is understood as a mental phenomenon and, in the absence of anthropomorphism, is understood to be about human minds. So viewed, SAW is not a computer system or a screen display; it is a state of human awareness. Equation (4.1) succinctly characterizes its composition and indicates that SAW is the combined product of perception, comprehension, and projection.

$$\text{SAW} = \text{Perception} \cup \text{Comprehension} \cup \text{Projection} \tag{4.1}$$

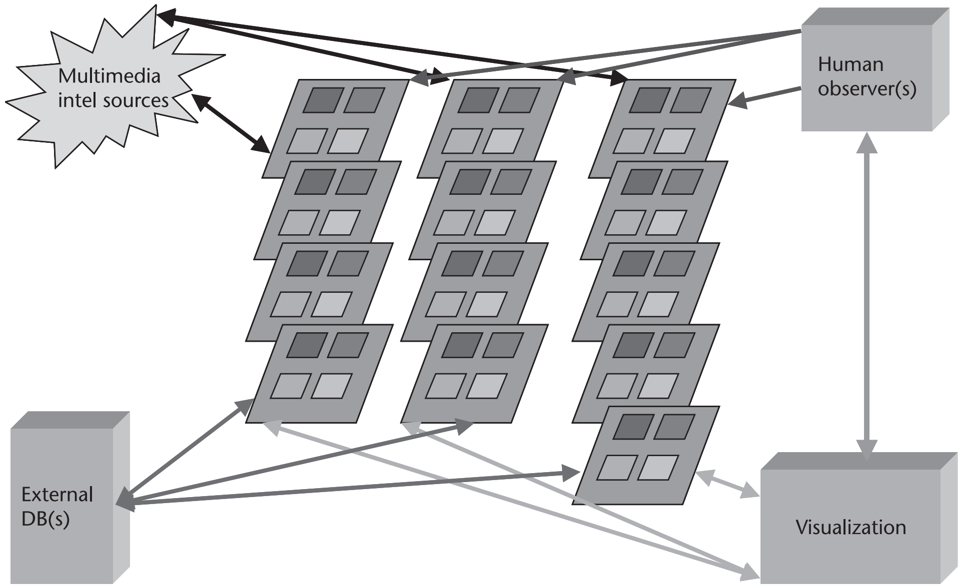


Figure 4.23 Multiple VDF elements as a multiple-concept, multiple-context, situation-understanding subsystem at discrete, multiple times.

Though Endsley's account of SAW originates from observations of U.S. military pilots, it should be emphasized that her account of SAW applies ubiquitously.

- SAW requirements exist across the strategic, operational, and tactical echelons of command. For example, the prime minister; the commander, Australian theatre (COMAST); and the director of northern operations (DNO) within the Royal Australian Air Force (RAAF) all have their own perception, comprehension, and projection requirements.
- SAW requirements exist for enemy (red), own force (blue), neural (white), environmental (green), and political (gray) aspects of conflict. SAW is not solely about enemy intelligence gathering. The threat imposed by an enemy's intent and capability in part depends upon a perception, comprehension, and projection of blue, white, green, and gray interests.
- SAW requirements exist across roles within a center of command. For example, while being a cognizant contributor to J5's own force-planning endeavors (J5 is Military Planning and Policy), the J4 (Military, Logistics, Engineering, and Security) retains his or her own perception, comprehension, and projection needs for understanding the particular logistical aspects of a campaign.

4.4.2 Common Operating Picture

Technologies and technological aids are often introduced to enhance the state of human awareness, and so the advancement of SAW is:

- Partly about psychology;
- Partly about technology;
- Partly about the integration of the two.

The development of a superior SAW capability is therefore not merely a matter of innovative information technology, but rather a question of how innovative information and interface technology can be applied with appropriate personal and organizational structures to yield the requisite psychological awareness. An innovative information capability is more than innovative information technology.

In that context, the concept of a common operating picture is often promoted as a vehicle for facilitating SAW. The U.K. Ministry of Defense has provided the following very broad notion of the COP [P. Houghton, private communication, 2001]: "The Joint Operational Picture (Common Operating Picture) is the total set of information, in whatever form, which is a managed and validated view of the history, current situation and future plans for all components of an operation."

The currently practiced view of the COP, however, is far less encompassing, so some means of characterizing alternative notions of COP is advantageous. As the COP exists to create SAW, and the creation of SAW is partly about psychology, technology, and the integration of the two, it is possible to characterize COP conceptions in terms of information deriving from

- Psychological processes (e.g., direct observation and cultural interpretation);
- Technological processes (such as sensors and databases);
- Integration processes (including telephones and computer interfaces).

Equation (4.2) succinctly characterizes the COP in terms of information deriving from psychological, technological, and integration processes. It states that the COP is the combined product of the products of psychological, technological, and integration processes.

$$\text{COP} = \text{Psychology} \cup \text{Technology} \cup \text{Integration} \tag{4.2}$$

4.4.2.1 Assessing COPs

Equations (4.1) and (4.2) allow us to assess coarsely how successfully a given set of COP processes meets our SAW needs. The SAW we want should be delivered by the COP information provided. Ideally, SAW = COP. The portion of SAW that we actually obtain from a COP is represented by SAW ∪ COP. By (4.1) and (4.2):

$$\begin{aligned}
\text{SAW} \cap \text{COP} = {} & (\text{Perception} \cup \text{Comprehension} \cup \text{Projection}) \\
& \cap (\text{Psychology} \cup \text{Technology} \cup \text{Integration}) \\
= {} & (\text{Perception} \cap \text{Psychology}) \cup (\text{Comprehension} \cap \text{Psychology}) \\
& \cup (\text{Projection} \cap \text{Psychology}) \cup (\text{Perception} \cap \text{Technology}) \\
& \cup (\text{Comprehension} \cap \text{Technology}) \cup (\text{Projection} \cap \text{Technology}) \\
& \cup (\text{Perception} \cap \text{Integration}) \cup (\text{Comprehension} \cap \text{Integration}) \\
& \cup (\text{Projection} \cap \text{Integration})
\end{aligned}$$

We can therefore view the portion of SAW that we obtain from a COP by considering how well the COP addresses the nine parts of SAW ∩ COP listed above. To visualize this contribution, it is convenient to represent it by the nine matrix elements of Figure 4.24.

The perception, comprehension, and projection aspects occupy the rows. They specify the SAW aspects that we want. The psychology, technology, and integration processes occupy the columns and identify the effect of the processes provided by the COP. We can then assess the value of a particular conceptualization of COP by the contribution that it makes within this matrix. Grey-scale shading can be used to provide a visual appreciation of the contribution. Figure 4.25(a) illustrates the case in which the COP processes contribute no SAW benefit (i.e., SAW ∩ COP = ∅). Figure 4.25(b) shows the case where the COP processes fully meet our SAW requirement; that is, SAW = COP, which ensures SAW ∩ COP = SAW.

4.4.2.2 Current COPs

The currently practiced conceptualization of COP tends to:

- View "picture" as some form of "dots on maps" integration display;

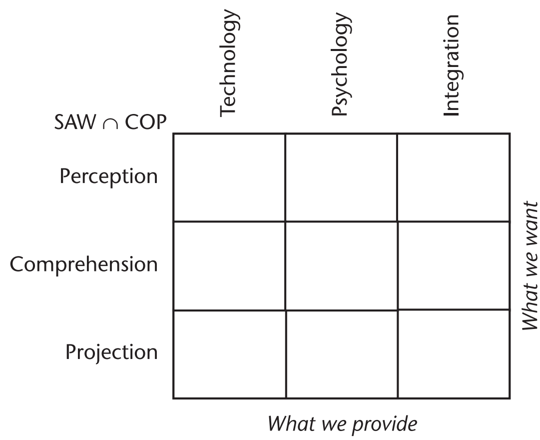


Figure 4.24 Assessing the COP contribution to situation awareness.

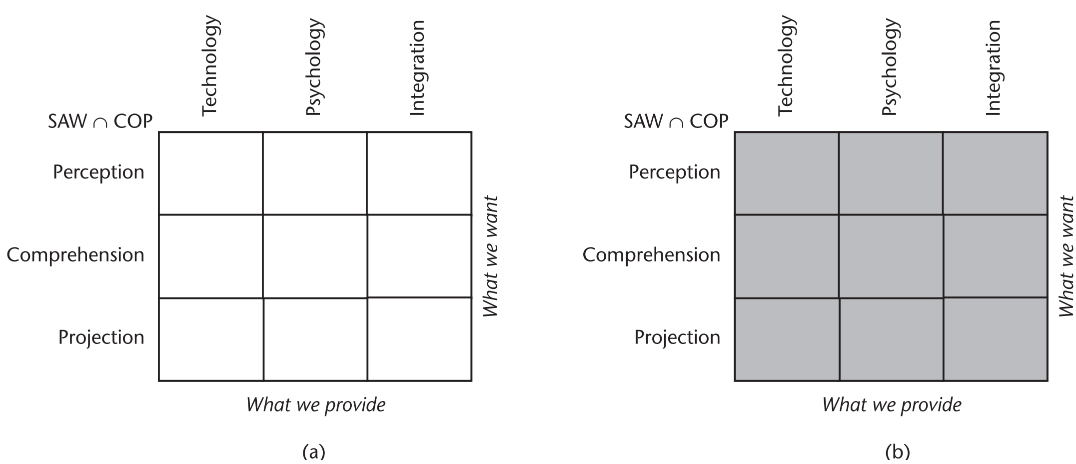


Figure 4.25 (a) COP delivers no SAW, and (b) COP delivers complete SAW.

- Promote "operating" as involving fusion technology for "joining dots" to recognize objects;
- Seek psychological unity by regarding "common" as the dissemination of hierarchical fusion products to all of the distributed contributing elements.

Figure 4.26 provides a gray-scale shading for this conception of COP.

The current conceptualization of *picture* is essentially a "dots on maps" display. In Figure 4.26, this is illustrated by a typical dot-annotated map graphic. As a picture, these displays are designed to integrate psychology and technology and only display entities that can be perceived by either humans or technology, such as radars. As a consequence, they primarily address only the top-right corner of the matrix. They tend neither to explain the behavior of the targets being represented (comprehension) nor to present predictive consequences of that behavior (projection). The one element of the matrix that is addressed has gray shading, lighter than that of Figure 4.25(b), because the current notion of *picture* is primarily limited to the presentation of tactical surveillance information.

The current conceptualization of *operating* involves fusion operations being applied to join dots (detections) to recognize targets. In Figure 4.26, this is illustrated by associating two linked plot data structures, each representing the detection of the same target aircraft at different times. Within the matrix, this is primarily about perception information delivered by a back-end technology process. Consequently, only the top-left element of the matrix is shaded. Again, this element of the matrix is partially shaded because only tactical surveillance information is usually considered in the current context.

The current notion of *common* involves the dissemination of hierarchical fusion products to all distributed battle-space elements. Figure 4.26 provides a representa-

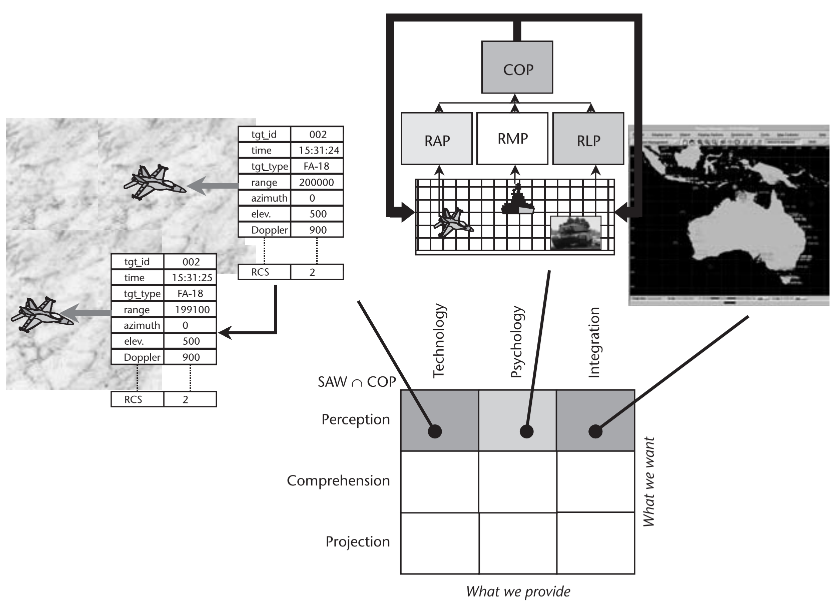


Figure 4.26 Assessing the current conception of COP.

tive diagram. The base of the diagram depicts assets within the battle space. Information from these assets is used to form a recognized air picture (RAP), a recognized maritime picture (RMP), and a recognized land picture (RLP). The outputs of these three pictures are in turn fused into a COP that is then disseminated back to the elements in the battle space. Again, the information being disseminated is primarily perception information, obtained through the operating fusion process and displayed through pictured dots on maps. The purpose of disseminating this information is to produce a common psychological understanding throughout the staff in the battle space. The top center element of the matrix is therefore shaded. The faint shading reflects the extent to which Lambert believes this approach will achieve a common psychology in the battle space.

Under the UDF/λJDL model we see that the current conceptualization of COP produces only a partial shading of the first row of the matrix. The current concept of COP therefore provides poor SAW capability as it:

- Lacks comprehension and projection aspects altogether;
- Treats perception as merely a "dots on maps" product of surveillance assets;
- Presents a simplistic notion of common inadequate for the required sophisticated operational level of command.

4.4.3 A View of Data Fusion

Lambert [15] suggests that the revised definition of data fusion [1] still retains too much of a tracking-literature description of proceedings. The omission of reference to data sources and time is also curious since both seem to be foundational to any real data-fusion process. In response, Lambert [15] has offered the following variation:

> Data fusion is the process of utilizing one or more data sources over time to assemble a representation of aspects of interest in an environment.

Lambert also sought a more unifying account of the levels of data fusion described in [1]. Lambert [15] directed attention toward levels 1, 2, and 3 of the JDL model by temporarily including level 0 within level 1 and by absorbing level 4 within each of the other levels. Lambert [15] provided the following revision of the Steinberg, Bowman and White [1] definitions for levels 1, 2, and 3:

- *Object fusion* is the process of utilizing one or more data sources over time to assemble a representation of objects of interest in an environment. An *object assessment* is a stored representation of objects obtained through object fusion.
- *Situation fusion* is the process of utilizing one or more data sources over time to assemble a representation of relations of interest between objects of interest in an environment. A *situation assessment* is a stored representation of relations between objects obtained through situation fusion.
- *Impact fusion* is the process of utilizing one or more data sources over time to assemble a representation of effects of situations in an environment,

relative to our intentions. An *impact assessment* is a stored representation of effects of situations obtained through impact fusion.

In the spirit of the equations of previous sections, (4.3) succinctly identifies the product DF of the JDL data-fusion process with its component object, situation, and impact assessment outcomes.

$$\text{DF} \approx \text{Obj Ass} \cup \text{Sit Ass} \cup \text{Imp Ass} \tag{4.3}$$

4.4.4 Mental Data Fusion and Situation Awareness

This unified model of data fusion can be applied in at least two ways. One approach is to apply it as a model of mental data fusion (DF_M) to characterize the activity performed by humans when fusing information. Figure 4.27 captures the sentiment. Equation (4.4) expresses the interpreted JDL relationship. It states that the product of mental data fusion is the product of mental object, situation, and impact assessments.

$$\text{DF}_M \approx \text{Obj Ass}_M \cup \text{Sit Ass}_M \cup \text{Imp Ass}_M \tag{4.4}$$

A comparison between (4.4) and (4.1) is instructive. Following Endsley's definition:

- Perception is about "the perception of the elements in the environment within a volume of time and space," while Obj Ass_M involves a stored representation of objects. If "stored representation" means mental representation, then Perception $\approx$ Obj Ass_M.
- Comprehension is about "the comprehension of their meaning," while Sit Ass_M is a stored representation of relations between objects. Attempts to understand meaning are many and varied but generally conceptualize meaning in terms of either reference to the world (e.g., Russell [18]), language

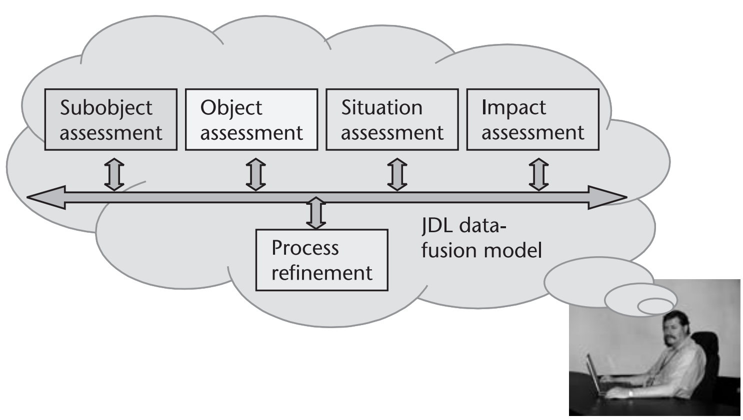


Figure 4.27 JDL model as a model of mental data fusion.

(e.g., Davidson [19]), propositions (e.g., Frege [20]), possible worlds (e.g., Kripke [21]) or psychology (e.g., Fodor [22]). A common thread across all of these approaches is that they involve relations between objects. If "stored representation" means mental representation, then Comprehension ≈ Sit Ass_M.

- Projection is about "the projection of their status in the near future," while Imp Ass_M involves a stored representation of the effects of situations. These are again closely aligned, so if "stored representation" means mental representation, then Projection ≈ Imp Ass_M.

From these three observations. it is reasonable to conclude

$$\text{Perception} \cup \text{Comprehension} \cup \text{Projection} \approx \text{Obj Ass}_M \cup \text{Sit Ass}_M \cup \text{Imp Ass}_M$$

and therefore

$$\mathrm{SAW} \approx \mathrm{DF}_M \tag{4.5}$$

by (4.1), (4.4), and the limited transitivity of ≈.

Equation (4.5) exposes an important unifying relationship between situation awareness and mental data fusion, namely that *mental data fusion is (essentially) situation awareness*.

The UDF/λJDL model delivers a model for SAW, with object assessments resembling perception, situation assessments resembling comprehension, and impact assessments resembling projection. Figure 4.28 highlights the unifying relationship between DF_M, the COP, and SAW.

4.4.5 Machine Data Fusion and Situation Awareness

As a part of the COP, we can also apply the unified data-fusion model as a framework for machine data fusion (DF_m). Figure 4.29 characterizes that applica-

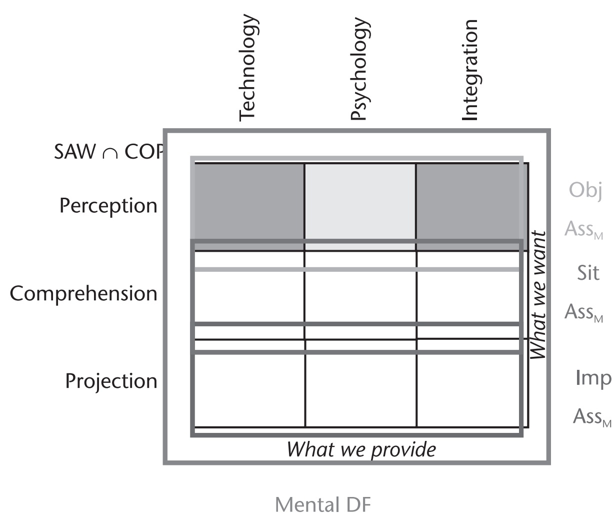


Figure 4.28 Relationship between SAW, the COP, and mental DF.

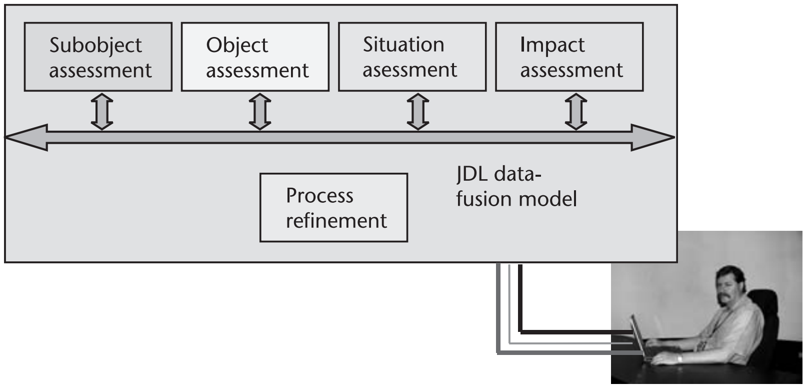


Figure 4.29 JDL model as a model of machine data fusion.

tion. Equation (4.6) expresses the machine interpretation of (4.3). It states that the product of machine data fusion is the product of machine object, situation, and impact assessments.

$$\text{DF}_m \approx \text{Obj Ass}_m \cup \text{Sit Ass}_m \cup \text{Imp Ass}_m \tag{4.6}$$

Machine data fusion is about automating within a machine the mechanical, error-prone, tedious, or prevalent aspects of human perception, comprehension, and projection. If we confine data fusion to mean machine data fusion, and so constrain "stored representation" to mean machine representation, then the JDL model is about the technological aspects of perception, comprehension, and projection. Thus,

$$(\text{Perception} \cap \text{Technology}) \approx \text{Obj Ass}_m \tag{4.7}$$

$$(\text{Comprehension} \cap \text{Technology}) \approx \text{Sit Ass}_m \tag{4.8}$$

$$(\text{Projection} \cap \text{Technology}) \approx \text{Imp Ass}_m \tag{4.9}$$

From this, we conclude,

$$\begin{aligned}&(\text{Perception} \cap \text{Technology}) \cup (\text{Comprehension} \cap \text{Technology})\\ &\quad \cup (\text{Projection} \cap \text{Technology}) \approx \text{Obj Ass}_m \cup \text{Sit Ass}_m \cup \text{Imp Ass}_m\end{aligned}$$

By distributivity, it follows that

$$((\text{Perception} \cup \text{Comprehension} \cup \text{Projection}) \cap \text{Technology}) \approx \text{DF}_m$$

and then, by (4.1), we obtain

$$(\text{SAW} \cap \text{Technology}) \approx \text{DF}_m \tag{4.10}$$

Equation (4.10) is significant. It tells us that within Figure 4.30, our SAW requirements will be met if we provide DF_m as the COP technology column of the matrix. In essence, *machine data fusion delivers the required technological basis for situation awareness*.

Furthermore, (4.7) to (4.9) indicate that *machine situation and impact assessments, respectively, provide the technological comprehension and projection aspects of SAW otherwise absent within the current conceptualization of the COP*.

Figure 4.30 highlights the unifying relationship between machine DF_m, the COP, and SAW. Figure 4.30 and the foregoing equations also highlight the fact that machine DF still requires integration with psychology before we can secure SAW. The UDF/λJDL model involves all nine matrix elements of Figure 4.24, and so addresses psychological and integration issues too.

Lambert [17] provides interpretations and documents the broader endeavors of the UDF/λJDL model, some aspects of which appear in Figure 4.31. This figure shows the Future Operations Centre Analysis Laboratory (FOCAL) being developed at the DSTO in Australia [17]. The FOCAL data-fusion system extends well beyond the traditional *machine sensor-fusion* emphasis of the data-fusion community by including higher-level information-fusion considerations involving both humans and machines.

4.4.6 A View of Sensor and Information Fusion

The machine data-fusion community has tended to draw a distinction between sensor fusion and information fusion. The term *sensor fusion* typically applies to levels 0, 1, and related parts of level 4, while the term *information fusion* is often used to refer to levels 2, 3, and related parts of level 4. Note that the term *information fusion* has sometimes been used synonymously with the term *data fusion*. Within this book, the narrower interpretation applies.

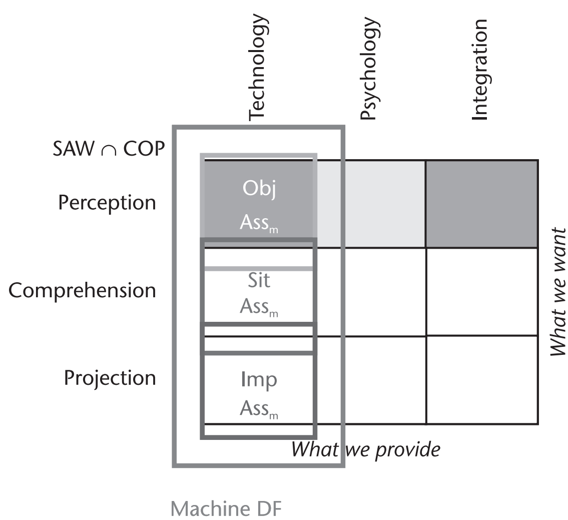


Figure 4.30 Relationship between SAW, the COP, and machine DF.

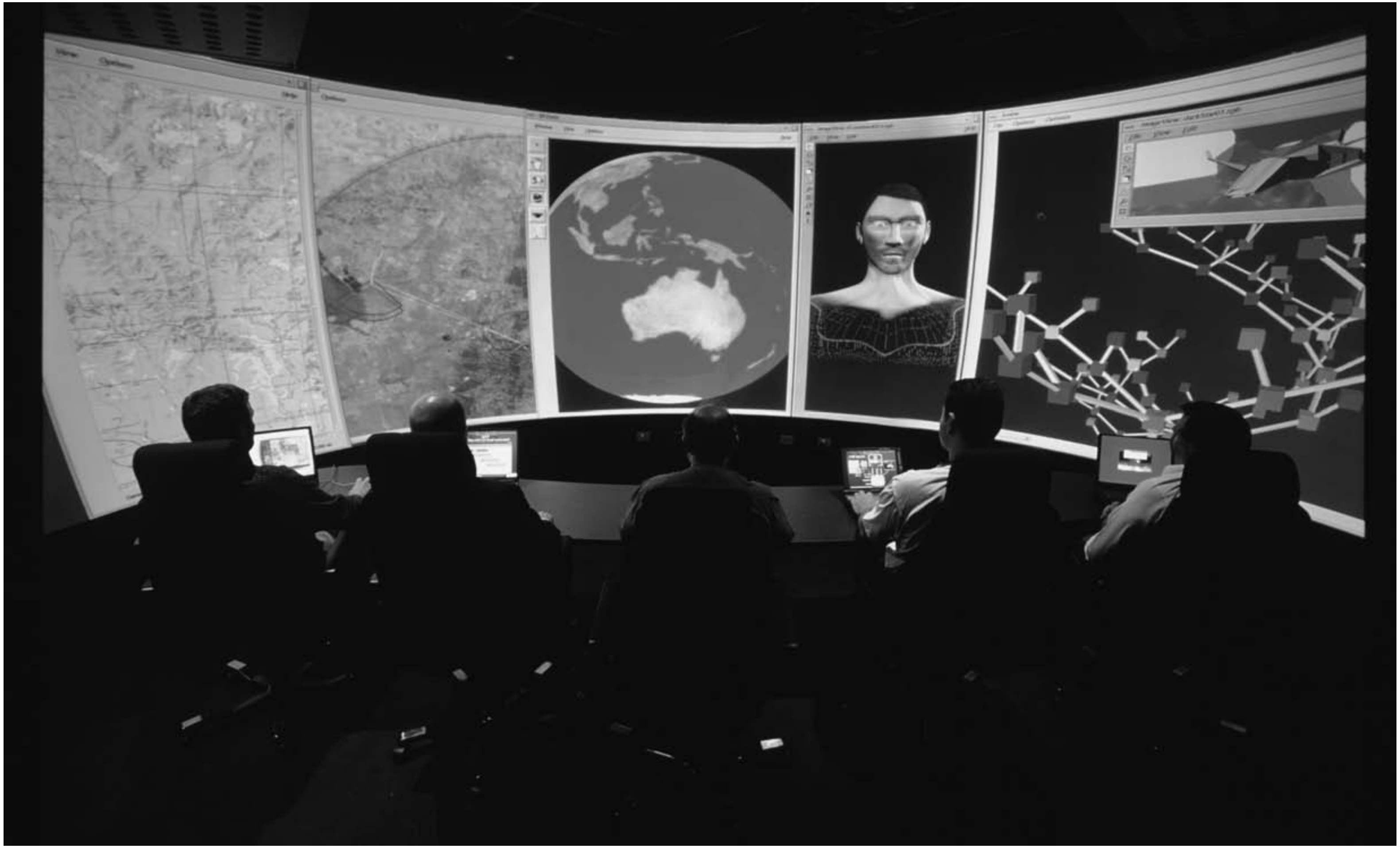


Figure 4.31 FOCAL integration of humans and machines [17].

We can express this division mentioned above by

$$DF_m = SF \cup IF \tag{4.11}$$

It is a purposeful distinction. Lambert [14, 15] argues that there is a paradigm disparity as we move from object assessments to situation assessments. Of object assessments Lambert [15] states,

> Object fusion exists because in our interaction with the world, we are inclined to associate bundles of near-coincident observable properties with objects, and to associate the persistence of those objects with the observed existence of those properties under periodic review. Object assessments allegedly document persistent objects having properties, and in a machine fusion context, these properties are usually measurable. The level-1 fusion literature therefore tends to be numerically based. In a radar environment, for example, signal and track processing is often used to conclude the existence of objects associated with, *inter alia*, measured range, azimuth, elevation, Doppler, radar cross section, target type and target identity properties.

Of situation assessments, Lambert states [15]:

> The emergence of the idea of a relation culminated with Ludwig Wittgenstein, who first explicitly proposed a *world of facts* as the fundamental substrate, where facts are subsequently understood as the application of relations to objects. In his cryptic, unapologetic style, Wittgenstein launched his 1922 publication of *Tractatus Logico-Philosophicus* with the words, "The world is all that is the case," and then "The world is the totality of facts, not of things." Wittgenstein supplanted a view that had persisted for over 2000 years. This fundamental shift in human conceptualization underpins the difference between "level 1" and "level 2" fusion.

Table 4.1 tabulates the conceptual disparities:

- Sensor fusion takes the world to be a world of objects with measurable properties. It represents the world by associating vector states with objects. Each element of the vector designates a property, while the numerical value of each element purports to be the numerical value of that measurable property.
- Information fusion takes the world to be a world of facts, where facts involve the application of relations between objects. Information fusion represents the world by using symbols to make claims about the world, which purport to express facts.

Table 4.1 Paradigm Disparity Between Sensor and Information Fusion

Fusion	*Primary Concept*	*Conceptual Origin*	*Primary Representation*
Sensor	Object	Aristotle	Numeric
Information	Fact	Wittgenstein	Symbolic

To give a simple illustration, at level 1 we might observe that object ship003 has a range property and that the value of that range property is 135 km, while object fighter002 also has the range property, but with a value of 200 km. At level 2, these observations might be fused into the symbolic sentence ship003 is closer than fighter002. The sentence applies the binary "is closer than" relation to the ship003 and fighter002 objects to claim that ship003's being closer than fighter002 is a fact about the world.

Figure 4.30 shows that the absent technological aspects of comprehension and projection in Figure 4.26 are provided under the JDL model by machine situation and impact assessments. In Figure 4.32, these absent components are collectively labeled "information fusion." This suggests that automated symbolic reasoning about facts provides the basis for machine comprehension and projection.

4.4.7 A View of Information Fusion

As shown in Figure 4.33, the practice of information fusion requires:

- Information sources;
- An information-fusion architecture;
- Domain knowledge through knowledge capture and representation.

The domain knowledge is captured and represented within the machine so that it can provide situation and impact assessments from information sources.

The UDF/λJDL model attempts to maximize compliance between human and machine, both in terms of ease of interaction between human and machine and in facilitating knowledge capture and transfer between human and machine. Consequently the UDF/λJDL model conceptualizes the fusion architecture as a multiagent system of human and software agents, and it conceptualizes the behavior of the software agents in a manner compliant with how we conceptualize the human agents. In the case of the latter, Lambert [23] explains:

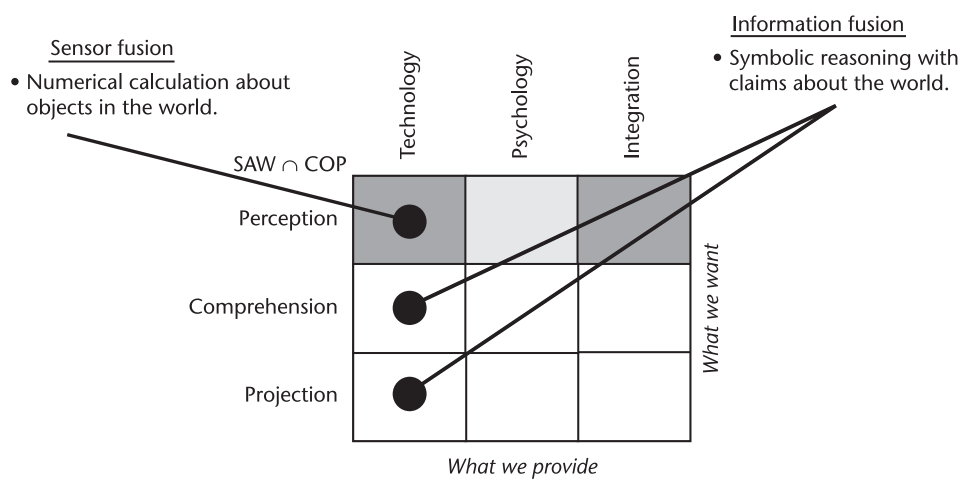


Figure 4.32 Relationship between SAW and information fusion.

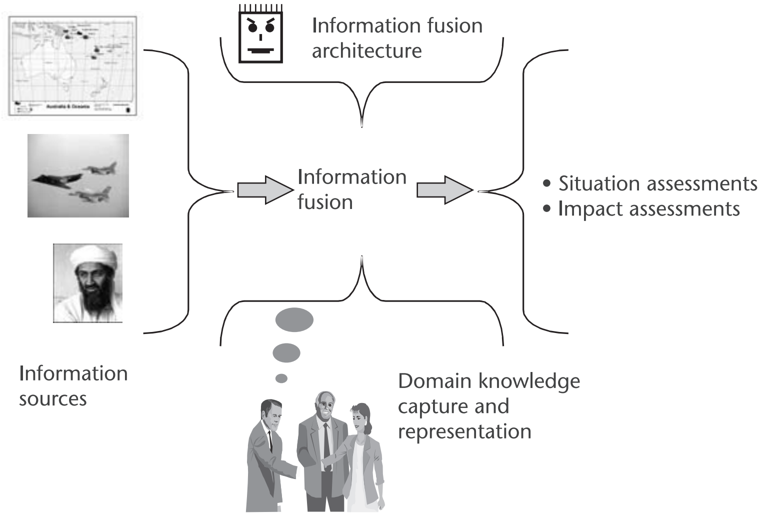


Figure 4.33 The practice of information fusion.

Humans predict and explain the behavior of other humans by ascribing mental attitudes to them, such as beliefs, desires, expectations, fears, hopes, *et cetera*, and when expressing these and other mental attitudes, the syntax of the expression always assumes the form

<subject> <attitude> that <propositional expression>

The following examples illustrate

Fred believes that the sky is blue
Tom expects that Mary will win lotto
Mary hopes that Tom is insightful

Expressions having this syntactic form are called *propositional attitude expressions* and the beliefs *et cetera* that they denote are technically termed *propositional attitudes*. In a propositional attitude expression: the subject, e.g., Fred, expresses which individual has the propositional attitude; the propositional expression, e.g. the sky is blue, expresses some assertion about the world; and the attitude, e.g. believes, expresses the kind of response the subject has toward the proposition.

From the standpoint of propositional attitudes, *situation assessment is about belief* [15]:

Three kinds of belief must be considered when engineering automated situation fusion. Environmental beliefs arise from direct observation of the world at a given time, such as when I believe that there is a can on the desk in front of me. In ATTITUDE, a sensor can be used to sense object assessments and create the corresponding beliefs in an event associated with that sensor. Definition beliefs derive from the meaning of terms, such as the belief that a son is a male child. When using the above relation previously, we will usually implicitly take it to be

> a transitive relation.[1] Such beliefs need to be explicitly included within the machine before we turn it on. Domain beliefs, express how we presume the world to be independently of direct observation, such as my belief that dark clouds usually produce rain. The domain beliefs can be expressed procedurally in terms of how something can be achieved (routines) or declaratively in terms of what is the case (inference). Situation assessments are situations comprising beliefs which are generated from the interaction of environmental, definition and domain beliefs.

Moreover, the beliefs about the world [24] are not oriented toward isolated facts [15]:

> When engaging the world, we rarely attend to individual facts in isolation. In assessing a typical mental snapshot picture of the world over a limited time frame and region, we are inclined to represent it as a *collection of facts*. In the early 1980s, Barwise and Perry suggested that situations were the fundamental building blocks of our assessment of the world.
>
> Reality consists of situations—individuals having properties and standing in relations at various spatiotemporal locations.
>
> Situations are essentially collections of related spatiotemporal facts, where facts consist of relations between objects. This is a step up from Wittgenstein's world of facts. Here the world is a world of situations, and assessing the world involves individuating situations. Situation assessment involves assessing situations, not facts or objects *per se*.

According to Lambert [14], situations are composed of events, events are composed of facts, and facts involve the application of relations to objects. Properties are unary relations.

From the standpoint of propositional attitudes, *impact assessment is about the effect of belief on desire* [15]:

> Via object assessments, situation assessments express beliefs (situations) about how the world might be. The consequences or effects of those beliefs are important to us, but only in as much as they impact upon what we want to be the case. This is the essence of "level-3" fusion—it is about how our beliefs impact upon our will. "Level-3" fusion is about the effect of situations on our intentions, and thus interprets the world in terms of opportunities and threats, with a view toward maintaining the satisfaction of our intent.

Mental behavior is cast in terms of routines that seek to satisfy desires given beliefs. Lambert [25] is developing an approach to the knowledge capture of mental routines based upon avowed propositional attitudes while problem solving. Working with Lambert at DSTO, Nowak [26] is developing a process metaphysics to support ontologies for representing the propositional content within these propositional attitudes.

1. $\forall x \forall y \forall z(((\text{above } zy) \;\&\; (\text{above } yx)) \Rightarrow (\text{above } zx)$.

4.4.8 The Grand Challenges of Information Fusion

In the framework of the UDF/λJDL model described earlier, Lambert [17] identifies a number of grand challenges of information fusion and illustrates how the FOCAL program at DSTO is addressing them. These challenges are:

- *Semantic:* What symbols should be used, and how do those symbols acquire meaning?
- *Epistemic:* What information should we represent, and how should it be represented and processed within the machine?
- *Paradigm:* How should the interdependency between the sensor-fusion and information-fusion paradigms be managed?
- *Interface:* How do we interface people with complex symbolic information stored within machines?
- *System:* How should we manage data-fusion systems formed from combinations of people and machines?

The epistemic challenge has been further discussed in Lambert [25], while the semantic challenge is the focus of Nowak and Lambert [27].

Nowak and Lambert [27] show how meaning can be assigned to symbols by formal theories with logics because they constrain the number of possible interpretations of those symbols. Implemented examples are discussed involving ontologies, inference engines, and agents. Concerning what symbols to use, their paper describes experimentation with a conceptually large military scenario. Two implementations are discussed. The first concerns an ontology and agents for semantically fusing legacy databases. The second concerns agents that receive queries through an agent grid and semantically fuse information to reply to the queries.

4.4.9 A State-Transition Data-Fusion Model

Recently, a state-transition data-fusion (STDF) model has been introduced by Lambert [28] as an extension of the dominant sensor-fusion paradigm to provide a unification of both sensor and higher-level fusion.

Both the JDL model and its deconstructed form (i.e., the UDF/λJDL model), segregate object, situation, and impact fusion into amorphous blocks, without explaining their internal mechanisms. In doing so, those models celebrate the difference between object, situation, and impact fusion, but at the expense of demonstrating their unity. Similarly, Lambert [14] notes that sensor-fusion representations of the world fail to scale up to higher-level fusion. A situation assessment of a missile targeting a communications tower will more likely resemble a symbolic expression than a set of state vectors. Nonetheless, there is a unifying framework for data fusion, which the STDF model introduced in Lambert [28] aims to expose.

4.5 The Situation-Awareness Reference Model

Around 2001, John Salerno from the U.S. Air Force Research Laboratory (AFRL) initiated another effort to develop a framework combining elements from the JDL data-fusion model with elements from Endsley's model of situation awareness. Based on these two models, Salerno [29] provides his initial discussion of a conceptual framework for situation awareness and assessment. His intent is to provide a roadmap for building a fusion system for situation awareness and assessment. He states that much work has already been accomplished in this area and that he believes it is a matter of bringing the components together and integrating them into an overall system architecture.

He begins the discussion of his concept from the bottom up, that is, from the data. The data is what restricts or confines our comprehension, thus projection. From there, he presents various components that he feels are necessary in building a situation-awareness framework, that is, acquiring perception; obtaining comprehension, information extraction, model-generation and learning algorithms (including data-mining techniques), model-analysis tools, and alert notification; and providing projection. To put things back in prospective, he finally presents a simple process flow of the proposed components for the concept. Actually, there are two major flows in the concept presented—a background and a "real-time" process.

The concept presented by Salerno is model driven; the problem with many such concepts is the existence and construction of such models. In this perspective, the primary focus of the background process in Salerno's concept is to build and nominate potential models that can be activated. Salerno believes collection, information extraction, and data mining to be key technologies in this portion of the concept, for if one cannot build such models, the concept will quickly fall apart.

The "real-time" portion of Salerno's concept is triggered as new data or information is provided. As this new information enters the system, it is examined for relevancy based on standing profiles. Information that passes this stage is then parsed to extract relevant attributes, which are sent on to determine whether the new information is of interest. Models are compared and prioritized based on the probability of activation. Based on this prioritization, a list of possible predictions can then be provided.

Salerno's initial conceptual framework is presented again in Salerno, Hinman, and Boulware [30] with minor differences, especially in the diagram used to discuss the process flow of the proposed components and the inclusion of some additional discussion of knowledge-discovery issues for building long-term memory.

Salerno's initial ideas were continuously refined between 2001 and 2004 [29–32], and a mature version of the conceptual framework was ultimately presented in [30], where, as a step in developing the goal of SAW, they presented a detailed discussion of the JDL and Endsley's models, as well as the motivation for combining the two by using Endsley's work to further define level 2 (situation assessment) of the JDL model.

Using Endsley's definition of SAW (slightly adapted "in order to enable decision superiority"), they drew further on the JDL model and Endsley's model to bridge the gap and develop a common framework for situation awareness. Essentially, they sought to use the strengths of both models and augmented these strengths to address shortfalls and provide more detail, relying on the JDL model for levels 0, 1, and 4, but using Endsley's notions of perception, comprehension, and projection (which they refer to as anticipation) as the overarching framework. In addition, they added an initial data-requirements component to enable top-down control over the entire process. The common framework, called the situation-awareness reference model, is shown in Figure 4.34 with three flows highlighted:

1. Process flow;
2. Offline processing;
3. Feedback.

The solid line displays the process flow, while the dashed line portrays the offline processing, and the dotted line shows the feedback portion of the model.

The process commences with the analyst's defining the problem of interest. In many areas (e.g., indications and warning), much experience and knowledge has been obtained through history, and various models have been developed that document this previous experience. The analyst begins with the adaptation of an existing model based on the specific concerns and parties involved (in terms of

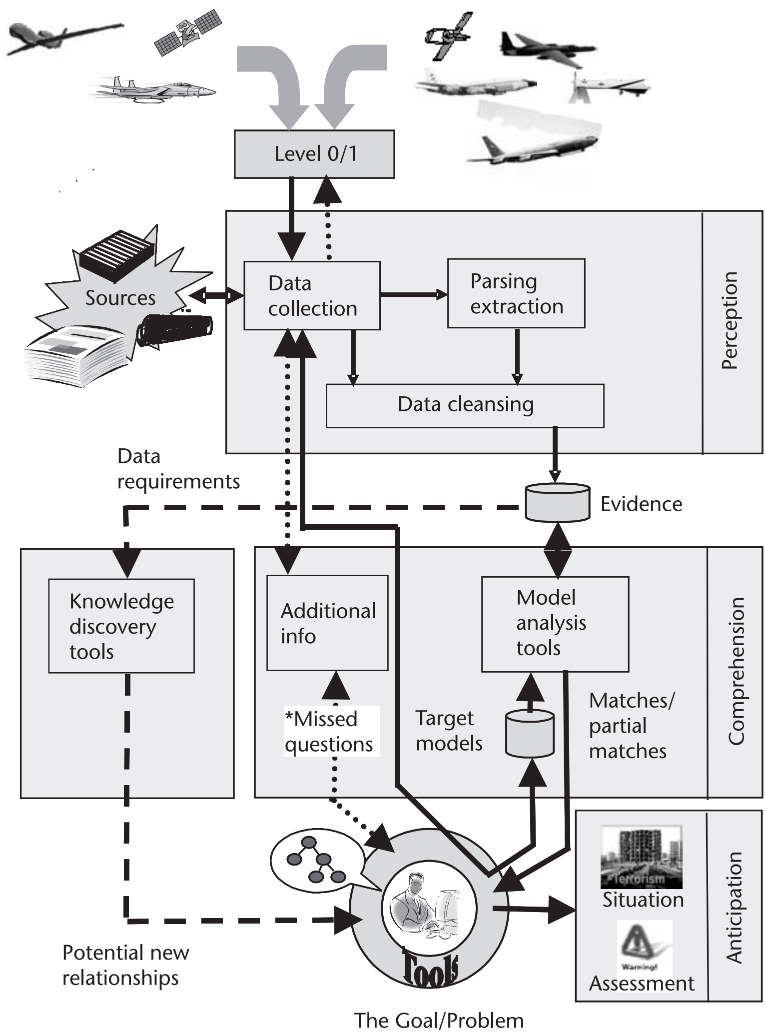


Figure 4.34 Situation-awareness reference model.

possible scenarios). This model defines the pattern(s) of interest and indirectly what data and information must be collected in order to develop an understanding of the situation.

The data-collection component receives the data requirements based on the model of interest and has the intelligence to determine how and where to gather the data, and when to request updates. It then gathers this data, wraps it in a common document structure, and publishes it along with metadata capturing various details, such as when the information was collected, what source the information came from, and the format of the data. Based on the format of the data, it may be necessary to parse it (e.g., formatted messages) or to extract relevant entities, relationships, and events through the use of natural language extractors. In any event, once events and relationships are obtained, a cleansing process must remove redundant, incomplete, and "dirty" data. This process must also deal with data transformations and aliases. The goal of this process is to provide evidence that is free from errors and contains perishability and confidence estimates. This evidence forms the perception. It should also be noted here that the collector is continuously gathering new data based on the problem(s) at hand.

Inherently, an accurate perception depends on various types of data. For example, information from various sensors may be vital. For this part, Salerno, Hinman, and Boulware rely on the JDL model (levels 0 and 1) to provide them with an interface between real-time sensor data and observable objects or events. Because of the many limitations of computers in "understanding" multimedia data, they must rely on many of the existing manual, human processes of exploitation. Here, they rely on the disciplines of information exploitation (IE). Simply put, IE can be considered a process to transform raw signals or data into formatted textual reports.

An example here might provide better insight into the applicability and value of IE. Systems that automatically process imagery are rare and provide minimal capabilities. For example, in imagery exploitation, imagery is collected, and imagery analysts (IA) or photo interpreters (PI) exploit the imagery by analyzing current images in the context of previous reports and other current and previous images. One output of this process is a textual report or message describing any significant events in the image. These reports are then disseminated throughout the intelligence community through message-handling systems. Most of these reports are structured for computer use. Based on this process and the state of the foreseeable future, we focus our attention on textual input. Because of the highly formatted structure of the reports, simple message parsers can extract useful information and insert it into the evidence database. Meanwhile, any free-text information may be processed by an information-extraction module that will extract the named entities, events, and relationships.

As previously discussed, level 0/1 identifies objects and tracks. At the SAW level, we may have to aggregate individual objects (e.g., tanks, trucks, artillery) into clusters with a label such as "unit" or "division." To accomplish this clustering, experimental methods group the objects together, and templates (some call these models) are used to label the group. In this case, the aggregated object provides more additional knowledge about the entity than tracking individual objects themselves. The fact that a division, rather than a number of objects, is moving can increase the significance of the overall concern. Another example is the aggregation of a number of pings in a computer network. In this case, such evidence could indicate that an Internet Protocol sweep is occurring and that the first part of an attack—the reconnaissance or data-gathering phase—is potentially underway. There are many other examples where observations are derived from a collection of evidence and must be aggregated together to produce a single overall observation or indicator.

As new evidence is gathered, model-analysis tools are used to determine if any parts of the target models appear. One way to accomplish this is to build a graph from the evidence (which Salerno, Hinman, and Boulware refer to as the input graph) and search for isomorphic instances of a particular model, or target graph. Based on the analysis, any portions of the input graph that match the target graph at or above a specified threshold are identified and provided to the analysts as alerts. This portion of the process defines the comprehension portion of the model. That is, the analyst captures knowledge gained in the past by defining the target graph and uses this knowledge to analyze what he or she has perceived thus far.

However, in many cases, the evidence coming in is incomplete, inconsistent, or incorrect. Before altering the status of a particular warning problem or issuing an alert, an analyst may want to find additional information to further clarify any inconsistencies. This is where the feedback loop comes into play. Models provide the user with knowledge that can be used to locate additional information. An analyst may wish to search for information that is missing from a particular model or use the model to project or anticipate what new data might exist and where. In addition, associated models or events may be used to locate information that collection initially missed as a result of its configuration. Analysts could also use current events as seeds for asking for more details pertaining to the event. Finally, a model can also be used for input to the sensor-management system in order to provide some intelligence as to where to look and what type of sensor to use to gather additional data that may be helpful in resolving the nature of the current situation.

The final thread is the offline processing portion of the system. Because the world is constantly changing, the system must also be able to learn. While analysts capture their experiences with manually generated models, it is often desirable to learn models automatically from data. These models have the potential to indicate activities, capabilities, and group memberships, and in some cases, they have outperformed humans. Besides learning models, such techniques can also identify additional relationships or entities that can augment existing models. This area is depicted in Figure 4.34 as knowledge-discovery tools. This is further discussed next.

4.5.1 Knowledge-Discovery Tools

Predictive analysis requires information about past events and their outcomes. Much of the work in this area requires a predefined model built by subject-matter experts or substantial amounts of data to train model-generation software to recognize patterns of activity. To date, these models are manually intensive to construct, validate, and interpret. Algorithms are needed to provide efficient inferencing, reasoning, and machine-learning procedures. Learning applications range from data-mining programs that can discover general rules in large datasets to "knowledge-assisted," hybrid approaches aimed at accomplishing deeper levels of reasoning and pattern identification.

As reported in [30], Witten, Frank, and Gray [33] defined data mining as the extraction of implicit, previously unknown, and potentially useful information from data. The idea is to build computer programs that sift through databases automatically, seeking regularities or patterns. They go on to state that strong patterns, if found, will likely generalize to make accurate predictions about future data. Data-mining activities can be divided into two types: (1) identifying patterns based on event associations, referred to as pattern learning, and (2) identifying groups based on similar activities, referred to as community generation.

It is crucial that we thoroughly sift through archived data to look for the associations between entities at multiple levels of resolution. Pattern-learning technologies serve to address this task by providing techniques that mine relational data. Pattern learning can be roughly described as the process of examining the relationships between entities in a database, the end-products of which are predictive models (statistical extrapolations) capable of describing what has been examined in terms of an abstract mathematical formalism (usually, a graph-theoretic construct). Relational data presents several interesting challenges:

- Relational learning must consider the neighborhood of a particular entity, not just a singular record.
- Most learning is predicated on (usually false) assumptions of independent samples. Relational data does not meet this criterion.
- Data must be semistructured to make learning possible. A query language must be developed to support the retrieval of data.

As also reported in [30], Jensen [34] states that the biggest concern in developing a pattern learner for situation awareness is the relatively low number of so-called positive instances, turning the pattern-learning process into an anomaly-detection process. Problems such as these are often considered "ill posed" in the computational learning community, and more often than not, partially invalid assumptions about the data must be made to correct for these conditions. If the learning process is improperly handled, low rates of positive instances will completely confound it, resulting in low-fidelity models, which produce high numbers of false positives and negatives. While the challenges are significant, so too is the potential payoff. Relational learning allows systems to exploit multiple tables in a database without the loss of information that occurs in a join or an aggregation [35]. The resulting discoveries may include predictive patterns that more accurately describe the world by utilizing entities' attributes, as well as the relationships between entities in the learning process.

Missing and corrupted data are also prime sources of error. Numerical data is naturally a bit easier to work with, given the fact that we can interpolate. The lack of numerical descriptors for the type of archived data with which we often deal exacerbates the issue of missing items. Luckily, a recent surge of research activity in the domain of relational learning has been addressing all of these issues.

Community generation and the class of problems it is trying to solve can be categorized as a matter of discerning group membership and structure. Under this topic, two types of paradigms are being investigated: one where two parties and the activity type are given, and one where only one party and one associated event is given. Zhang et al. [36] describe the first class as biparty and the later as uniparty.

Community-generation algorithms will typically take events and relationships between individuals (whether implicit or explicit) and develop some correlation between them. This correlation value defines the strength of the link. Why are these models important to us? The models derived provide us with insights into organizational structure and people of interest. Let us consider the first instance—organizational structure. Suppose that we have identified two groups whose structures are shown in Figure 4.35.

We can easily see from the models shown in Figure 4.35 that there is a key node in the model, which if removed or identified could have major impacts on the community. In this case, it could be a key individual within an organization. A second use of this information is the development of a behavioral model for the group. Knowing the individuals in charge of the group and "understanding" their behaviors could facilitate more advanced modeling and simulation capabilities, as well as direct surveillance efforts.

4.5.2 Situation-Awareness Reference Model Applied to the Cyberdomain

In addition to presenting their situation-awareness reference model, Salerno, Hinman, and Boulware [30] also demonstrate in their paper how this framework can be applied to a sample, well-known "monitoring" problem. Subsequently, they have been researching the application of their new model to other domains, including the asymmetric threat, tactical, and homeland security domains. In each of these domains, the primary objective is to aid an analyst in making sense out of a glut of raw data, that is, to aid in the understanding and awareness of a current or an unfolding or evolving situation. Although the same basic model can be applied to all of these domains with the same objective, each domain also has unique problems, including the volume and format of available data and the time available to identify the evolving situation. Salerno, Hinman, and Boulware have found that in the strategic domain, the amount of available data is so vast, exceeding hundreds of terabytes per day and consisting of countless pages of open-source data, sensor data, and analysts reports, that they can't adequately assess whether they have accurately identified a situation or measure improvements realized by SA systems.

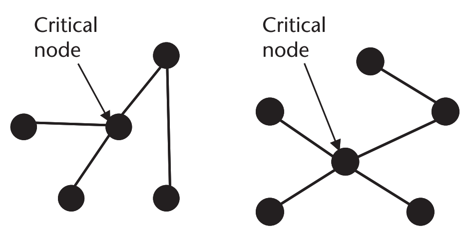


Figure 4.35 Community-generated models [30].

Although the situation-awareness reference model and a lot of their conceptual development originated in the strategic domain, Salerno, Hinman, and Boulware have discovered over the last few years that in the cyberdomain, particularly in the detection of network attacks, the model provides them with many distinct advantages [31]. While by no means a simple problem, the cyberdomain is proving to be a much more bounded problem then the strategic domain. The cyberdomain benefits from data sources that are more structured (TCP packets, Snort or sensor alerts, and even system log files have distinct formats or data fields), data volume that can be restricted without losing the overall context or making the problem trivial, and the ability to establish reliable ground truth for learning, testing, and evaluation. From that perspective, as illustrated in Figure 4.36, they've adapted the generalized situation-awareness reference model specifically to the cyberdomain. Details about this recent work can be found in Tadda et al. [37].

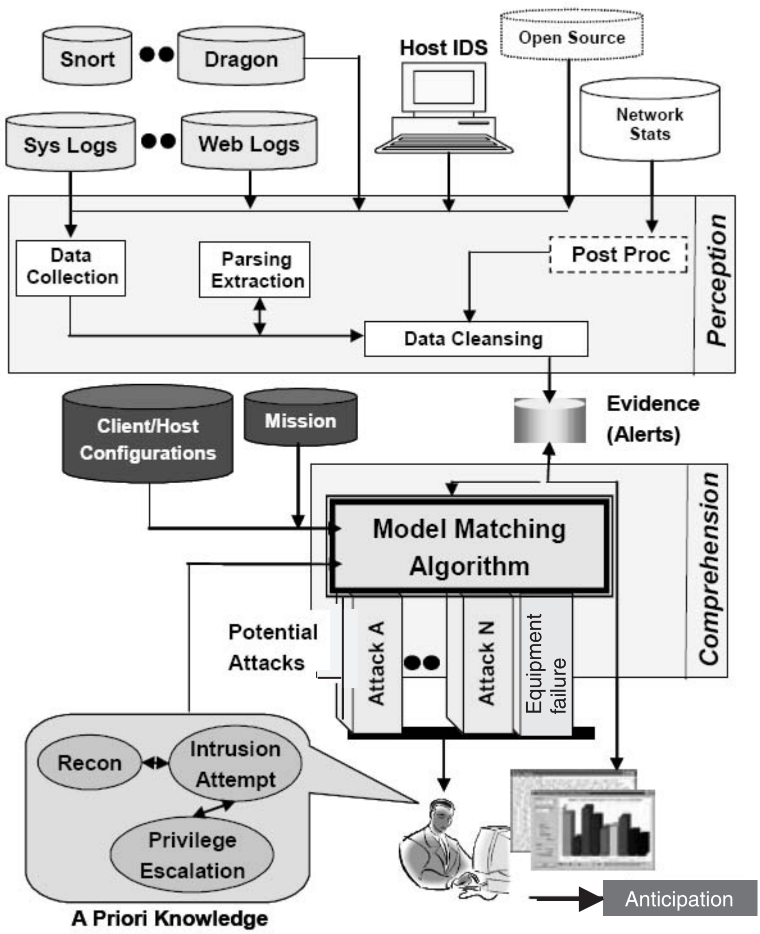


Figure 4.36 Situation-awareness reference model applied to the cyberdomain [37].

References

[1] Steinberg, A. N., C. L. Bowman, and F. E. White, "Revision to the JDL Data Fusion Model," *Joint NATO/IRIS Conference*, Quebec City, Quebec, Canada, October 1998.

[2] Llinas, J., et al., "Revisiting the JDL Data Fusion Model II," *Proceedings of the ISIF 7th International Society on Information Fusion Conference (Fusion 2004)*, Stockholm, Sweden, June 28–July 1, 2004.

[3] White, F. E., Jr., *Data Fusion Lexicon*, Joint Directors of Laboratories, Technical Panel for C3, Data Fusion Sub-Panel, Naval Ocean Systems Center, San Diego, CA, 1987.

[4] Llinas, J., and R. T. Antony, "Blackboard Concepts for Data Fusion Applications," *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 7, No. 2, 1993.

[5] Antony, R. T., *Principles of Data Fusion Automation*, Norwood, MA: Artech House, 1995.

[6] Bowman, C. L., "The Data Fusion Tree Paradigm and Its Dual," *Proc. 7th National Symposium on Sensor Fusion*, Albuquerque, NM, March 1994.

[7] Steinberg, A. N., and C. L. Bowman, "Development and Application of Data Fusion Engineering Guidelines," *Proc. 10th National Symposium on Sensor Fusion*, 1997.

[8] Llinas, J., et al., *Data Fusion System Engineering Guidelines*, Technical Report 96-11/4, USAF Space Warfare, 1996.

[9] Blasch, E. P., and S. Plano, "Level 5: User Refinement to Aid the Fusion Process," *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2003, Proc of the SPIE*, Vol. 5099, 2003.

[10] Hall, D. L., and J. Llinas, *Handbook of Multisensor Data Fusion*, Boca Raton, FL: CRC Press, 2001.

[11] Hall, D. L., and J. Llinas, "An Introduction to Multisensor Data Fusion," *Proceedings of the IEEE*, Vol. 85, No. 1, January 1997.

[12] Waltz, E. L., and J. Llinas, *Multi-Sensor Data Fusion*, Norwood, MA: Artech House, 1990.

[13] Karakowski, J. A., "Towards Visual Data Fusion," *NSSDF International Open Session*, Georgia Tech Research Institute, Atlanta, GA, April 1998.

[14] Lambert, D. A., "Assessing Situations," *Proceedings of IEEE 1999 Information, Decision and Control*, Adelaide, SA, Australia, February 8–10, 1999, pp. 503–508.

[15] Lambert, D. A., "An Exegesis of Data Fusion," in L. Reznik and V. Kreinovich, (eds.), *Soft Computing in Measurement and Information Acquisition*, Studies in Fuzziness and Soft Computing, New York: Physica Verlag, 2001.

[16] Lambert, D. A., "Situations for Situation Awareness," *Proceedings of the ISIF Fourth International Conference on Information Fusion (FUSION 2001),* Montreal, Canada, August 7–10, 2001.

[17] Lambert, D. A., "Grand Challenges of Information Fusion," *Proceedings of the ISIF Fusion 2003 Conference*, Cairns, Australia, July 7–10, 2003, pp. 213–220.

[18] Russell, B., "On Denoting," in I. M. Copi and J. A. Gould, (eds.), *Contemporary Philosophical Logic*, New York: St. Martin's Press, 1905, pp. 84–96.

[19] Davidson, D., "Truth and Meaning," in D. Davidson, (ed.), *Inquiries into Truth and Interpretation*, Oxford, U.K.: Clarendon Press, 1967, pp. 17–36.

[20] Frege, G., "Sense and Reference," in I. M. Copi and J. A. Gould, (eds.), *Contemporary Philosophical Logic*, New York: St. Martin's Press, 1892, pp. 65–83.

[21] Kripke, S. A., *Naming and Necessity*. Oxford, U.K.: Basil Blackwell Publisher, 1980.

[22] Fodor, J. A., *Psychosemantics: The Problem of Meaning in the Philosophy of Mind*, Cambridge, MA: MIT Press, 1987.

[23] Lambert, D. A., "Advisers with Attitude for Situation Awareness," *Proceedings of the 1999 Workshop on Defence Applications of Signal Processing*, La Salle, IL, 1999, pp. 113–118.

[24] Barwise, J., and J. Perry, *Situations and Attitudes*, Cambridge, MA: MIT Press, a Bradford Book, 1983.

[25] Lambert, D. A., "Automating Cognitive Routines," *Proceedings of the ISIF 6th International Conference on Information Fusion*, Cairns, Australia, July 7–10, 2003.

[26] Nowak, C., "On Ontologies for Higher-Level Information Fusion," *Proceedings of the ISIF 6th International Conference on Information Fusion*, Cairns, Australia, July 7–10, 2003.

[27] Nowak, C., and D. A. Lambert, "The Semantic Challenge for Situation Assessments," *Proceedings of the ISIF 8th International Conference on Information Fusion*. Philadelphia, PA, July 25–29, 2005.

[28] Lambert, D. A., "A Unification of Sensor and Higher-Level Fusion," *The ISIF 9th International Conference on Information Fusion (Fusion 2006)*, Florence, Italy, July 10–13, 2006.

[29] Salerno, J., "Information Fusion: A High-Level Architecture Overview," *Proceedings of the ISIF International Conference on Information Fusion (Fusion 2002)*, Annapolis, MD, July 8–11, 2002.

[30] Salerno, J. J., M. Hinman, and D. Boulware, "Building a Framework for Situation Awareness," *Proceedings of the ISIF International Society on Information Fusion Conference (Fusion 2004)*, Stockholm, Sweden, June 28–July 1, 2004.

[31] Salerno, J. J., M. Hinman, and D. Boulware, "A Situation Awareness Model Applied to Multiple Domains," *Proc of the Defense and Security Conference*, Orlando, FL, March 2005.

[32] Salerno, J. J., et al., "Information Fusion for Situational Awareness," *Proceedings of the ISIF 6th International Conference on Information Fusion (Fusion 2003)*, Cairns, Australia, July 7–10, 2003.

[33] Witten, I., E. Frank, and J. Gray, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, San Francisco, CA: Morgan Kaufman Publishers, 1999.

[34] Jensen, D., "Statistical Challenges to Inductive Inference in Linked Data," *Proceedings of the 7th International Workshop on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, January 3–6, 1999.

[35] Dzeroski, S., "Multi-Relational Data Mining: An Introduction," *Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining*, Vol. 5, No. 1, July 2003.

[36] Zhang, Z., et al., "Using Data Mining Techniques for Building Fusion Models," *Proceedings of SPIE: Data Mining and Knowledge Discovery: Theory, Tools, and Technology V*, Orlando, FL, April 2003, pp. 174–180.

[37] Tadda, G., et al., "Realizing Situation Awareness in a Cyber Environment," *Proceedings of SPIE's Symposium on Defense and Security*, Orlando, FL, April 17–21, 2006.

CHAPTER 5

Situation Analysis and Decision-Support Systems

Jean Roy, Richard Breton, and Robert Rousseau

5.1 Introduction

The characteristics inherent in the command-and-control (C2) domain in military and public-security operations pose significant challenges to the C2 process, to the design of future C2 systems, and to the personnel responsible for conducting this process using these systems to fulfill their objectives. C2 is characterized by ill-structured problems, changing and stressful conditions, technological advances in threat technology, the increasing tempo and diversity of scenarios, and the volume, rate, imperfect nature, and complexity of the information, among other things. Most likely, the latter will be processed under time-critical conditions; as a consequence, the risks of saturation in acquiring and maintaining situation awareness and of making the wrong decision increase. Although human qualities such as initiative and creativity and the notions of responsibility and accountability remain essential, the support of the technology is clearly required to cope with such characteristics in order to complement human capabilities and address human limitations [1].

Such a technological perspective of C2 has led system designers to propose solutions to overcome many of the domain problems by fitting operational platforms with support systems for data and information fusion, situation analysis, and decision-making. The main role of such systems is to help the operational personnel to acquire and maintain the appropriate situation awareness for their decision-making activities and to support the execution of the resulting actions.

This chapter discusses various issues related to the design and insertion of such technological tools in the decision-making process. In the past, the lack of knowledge in cognitive engineering has often jeopardized the design of helpful computerized aids aimed at complementing and supporting human cognitive tasks. Moreover, this lack of knowledge has, most of the time, created new problems regarding trust in the designed tools and human-in-the-loop concerns. Supporting decision-making in complex military and public-security operations indeed requires balancing the human-factors perspective with the that of the system designer and coordinating efforts to design a cognitively fitted system. In this regard, this chapter presents a triad model establishing the relationship between the elements required for the design of a system that support humans: the task, the human, and the technology.

The model allows for the design of systems taking into account the human role in a dynamic decision-making process like command and control.

5.2 Human Limitations

Providing as much data and information as possible about the situation(s) and the environment is not necessarily an adequate way to support the decision-maker's performance. As illustrated in Figure 5.1, all this data and information may exceed human information-processing capabilities. The human only has limited attentional and memory resources, and only a small fraction of all the data and information available can thus be processed (i.e., perceived and understood). Unfortunately, many situations require that a lot of different pieces of information be considered simultaneously, exceeding the human short-term memory and attentional resources.

The difference between the information required for optimal decision-making and the information actually processed by humans is called the information gap. Some of the factors related to human capabilities and limitations are further discussed next.

5.2.1 Stress

Physical factors like stress and fatigue must be considered when assessing human skills and limitations to perform a task. According to Proctor and Van Zandt [2], stress refers to a physiological response to unpleasant or unusual conditions. These conditions may be imposed by the physical environment, the task performed, one's

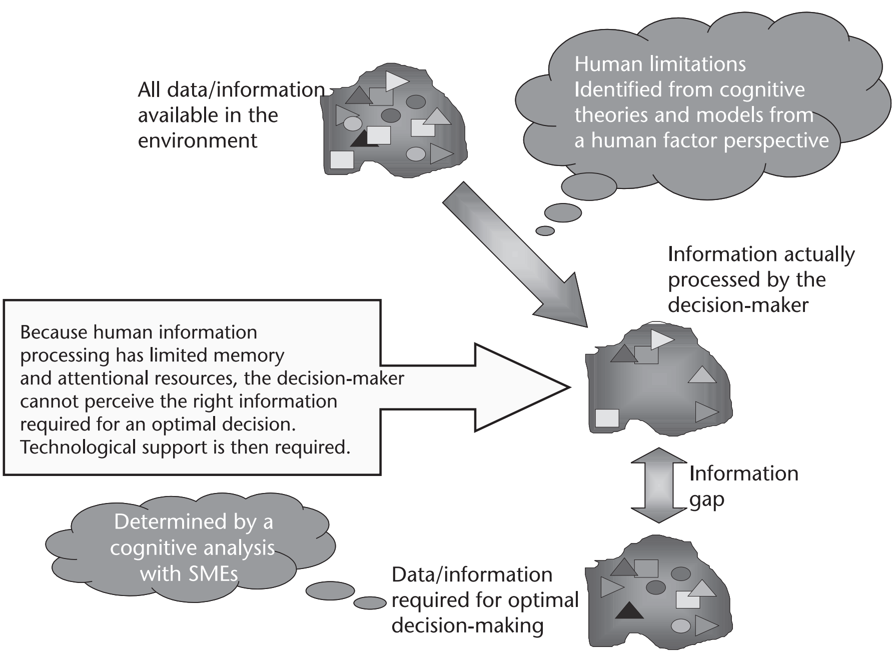


Figure 5.1 Information required and human limitations—the information gap.

personality, or social interactions. Stressful situations are defined by a substantial imbalance between the demands imposed by the environment and the human's capability to handle those demands successfully. Stressful situations are created by overload and also by underload [3]. The influence of physical factors on decision-making abilities has been investigated in the Tactical Decision Making under Stress (TADMUS) project following the Vincennes incident (explained in [4]).

5.2.2 Attention

The human has limited resources, and these resources are generally related to attention capacity. It seems that the attention is divided into limited pools of resources. There is some multiplicity of nonoverlapping reservoirs [5]. The pools are related to each specific sensory modality [6]. Hence, two different tasks can be performed simultaneously if they refer to different pools. For instance, it is possible to drive a car and talk with someone at the same time. However, it is impossible to sing and talk simultaneously. This affirmation brings the concept of serial and parallel processing. Two different tasks that refer to different pools can be processed in parallel. However, they must be processed serially if they refer to the same pool. In the latter situation, the workload related to the two different tasks determines the complexity of the situation. The workload can be defined by the demand required by the execution of a task in function of the resources available in the pools. The workload cannot be solely defined in terms of attentional resources.

5.2.3 Working Memory

The working memory is also involved in any attentive activity. The working memory is the cognitive center responsible for problem solving, retrieval of information, language comprehension, and many other cognitive operations [7]. To encode words in the long-term memory, the human must be attentive to these words, and the flow of the presentation of the words cannot exceed the capacity of the working memory. Unfortunately, the storage and processing capacity of the working memory is limited. However, these limited resources can be expanded through practice.

5.2.4 Workload and Level of Expertise

The workload related to a task is thus defined by the demands imposed by the task in terms of attentional and working-memory resources needed. Moreover, human performance is closely related to the workload of the task. Tasks with high workloads can be seen as more complex than those with low workloads. However, strategies, practice, and training can reduce the workload to a level at which enough resources are available. The idea that mental events operate automatically after a certain amount of practice is a well-entrenched doctrine of folk psychology, and it has a long history in academic psychology [6]. According to Schneider and Shiffrin [8], mental operations that are trained sufficiently are performed more quickly and accurately. They also undergo qualitative changes. Trained operations impose lower-capacity demands, providing more resources for concurrent mental activities. Trained operations are also not subject to voluntary control or conscious awareness and require little or no mental effort.

Rasmussen [9] proposes a skill-rule-knowledge (SRK) framework including three different levels of performance in which the automation is different. At the skill-based level, human performance is governed by stored patterns of knowledge. This knowledge is acquired with practice. With a specific stimulation from the environment, a specific response is given. The link between the stimulation and the response can be seen as a reflex that requires no effort or conscious awareness. The second level is the rule-based level, which is applicable to tackling familiar problems in which solutions are governed by rules (if-then-else). Processes related to this level are mainly automatic. With new situations, the third level described by Rasmussen is involved. The knowledge-based level deals with unfamiliar situations for which actions must be planned online, using conscious analytical processes and stored knowledge. These processes are controlled and impose a high mental workload. However, with practice and training, unfamiliar situations become familiar and can thus be solved at the rule-based level. Moreover, with extended practice, this knowledge can even become a reflex in specific situations (skill-based level). Dreyfus [10] proposes five different stages to becoming an expert (novice, advanced beginner, competent, proficient, and expert).

5.3 Technological Support for Situation Awareness and Decision-Making

In view of the discussion above, technological support is required to cope with human limitations when facing very complex and ill-defined problems within uncooperative C2 environments. Even with extended practice and the use of strategies, the human may require the support of systems; with technological developments, it is indeed highly appealing to tackle C2 problems by providing humans with computer-based information-fusion, situation-analysis, and decision-support systems. However, it is crucial that these systems be designed according to human information-processing requirements. A cognitive analysis should provide an understanding of how the human perceives the C2 task and should define the constraints of the environment. From such an analysis, it is crucial to identify and understand how the human perceives a task, which processes are involved, what the human needs are, which part of the task can and must be automated, and which part of the task can and must be supported. Human shortfalls are thus eventually translated into requirements for the technology community.

In particular, to support the decision-maker adequately and be compatible with his or her information-processing capabilities, the technology must be designed to present only the critical information required by the decision-maker to execute the task. As an example of a problem that can be raised in providing as much information as possible via the technology, let's take a situation in which a human has to detect a piece of specific, rare, and subtle information from his or her environment without the support of any technology. This attentional task can be defined as a vigilance task. In this situation, the human's performance is related to his or her ability to detect that rare and subtle information from the environment. Now, let's imagine this same human having to detect that same information, but with the support of the technology. If such technology provides too much information, the nature of the attentional task may be shifted from a vigilance task to one of recognition, discrimination, and selection. Instead of having to detect the critical information, the human now has to recognize and select it from an impressive pool of information. Thus, the difficulty of the task is no longer related to the detection of rare and subtle information but rather to the recognition and selection of this same information from many others plausible inputs. Both situations are very challenging for the decision-maker, but for different reasons.

Ultimately, the information that enhances the decision-maker's SAW and increases the probability of an accurate decision-making process must be available at the right moment in the situation. It must also be presented in a format compatible with human information processing.

By comparing the mental workload imposed on the human (the critical information to be processed) and human information-processing capabilities, one may identify the human limitations that translate into technological requirements. Such requirements lead to the design of a support system that is compatible with both the information required by the C2 task and human limitations (see Figure 5.2). The role of the support system is to bridge the gap between the demands of the task (the workload) and human capacities [1]. As illustrated in Figure 5.2, not all of the information required for optimal decision-making may ultimately be captured by the technology. However, one may claim that the technological capabilities regarding this issue often exceed the human capabilities.

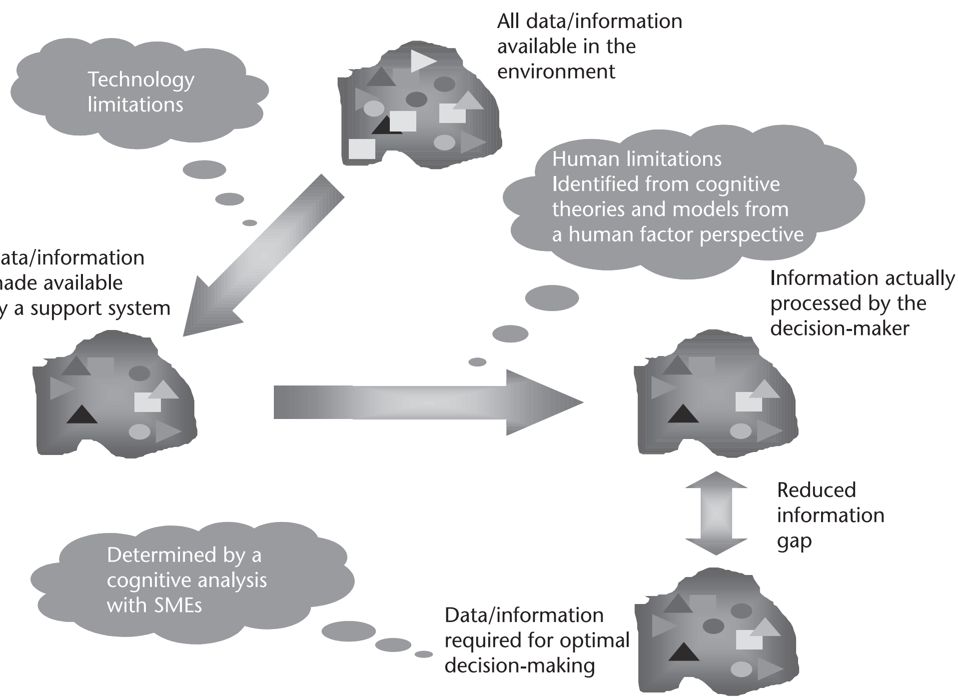


Figure 5.2 Reducing the information gap with a support system.

5.4 Task-Technology Interactions and Technological Automation

Evidently, humans and machines have different capabilities for performing various tasks [11]. On one hand, in addition to number-crunching capabilities, computer-based systems have great deductive capacities. However, they can hardly perform inductive reasoning. On the other hand, the human can hardly deal with several hypotheses at the same time but has the capacity to perform inductive reasoning. According to Ballas et al. [12], inducing hypotheses is better accomplished by humans, and the validation of these hypotheses is efficiently done by computer-based aids.

Typically, technological automation changes the nature of the implication of the human (i.e., it redefines the human contribution). With automated systems, the human role is often mainly related to the supervision of the situation (i.e., the role of the human shifts from a controlling one toward a monitoring one). This new role brings new problems and issues to consider. Indeed, according to Bainbridge [13], the automation of processes may expand rather than eliminate problems with the human operator. Such technological developments may increase the complexity of the environment, thereby imposing higher processing demands to the human. In fact, Bainbridge suggests that the more advanced a system is, the more crucial the contribution of the human may be.

Bainbridge also raises an important point with automated systems. One can only expect the operator to monitor the computer's decisions at some metalevel to decide whether the computer's decisions are acceptable. If the computer is being used to make decisions because human judgment and intuitive reasoning are not adequate in the context, then which of the decisions are to be accepted? The human in a monitoring role cannot handle the information-processing and decision loop anymore. Most likely, the human will not be able to cope with the system and, consequently, will not use it due to a lack of proper understanding or trust.

Technological automation also raises the question about which part (i.e., the human or the system) has the authority. There is no general answer to this question. A proposed approach is to delegate authority according to the situation. Chalmers [14] proposes five modes of operator-system delegation. The human selects the mode, which applies until mode transition is triggered by a new selection. It is obvious that a good understanding of the situation is crucial to select the required mode. Each mode implies a fixed delegation of authority for all the various subprocesses for which automated support is available. Figure 5.3 presents these modes, along with variations in the level of work distribution and the synergy between the automation and the operator in these various modes.

The issues discussed above do not mean that support systems or automated systems are not useful. However, their design, purposes, and interaction with the human are critical. Moreover, given the nature of unpredictable events, it is crucial that the design process start with a complete understanding of the environmental constraints and human information processing. The technological perspective must be seen as the solution to human shortfalls. Hence, the design process must involve systems designers and human-factors specialists.

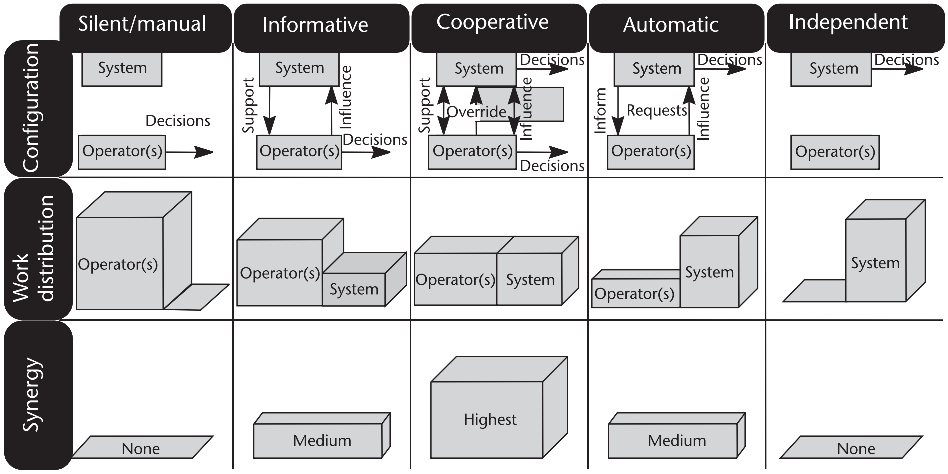


Figure 5.3 Operator-system modes of operation.

5.5 Support-System Requirements: A Task/Human/Technology Triad Model

A triad approach has been proposed by Breton, Rousseau, and Price [15] to represent the collaboration between the support-system designers and the human-factors specialists. As illustrated in Figure 5.4, the three elements that compose the triad are the task, the technology, and the human. In the command-and-control context, the OODA loop represents the task to be accomplished. The design process must start with the identification, by subject-matter experts (SMEs) within the context of a cognitive analysis, of the environmental constraints and possibilities.

Support-systems designers are introduced into the triad via the technology element. Their main axis of interest is the link between the technology and the task. The general question related to this link is, What systems must be designed to accomplish the task? Systems designers are also considering the human. Their secondary axis of interest is thus the link between the technology and the human. The main question of this link is, How must the system be designed to fit with the human? However, systems designers have a hidden axis. The axis between the human and the task is usually not covered by their expertise. From their analyses, technological possibilities and limitations are identified. However, all environmental constraints may not be covered by the technological possibilities. These uncovered constraints, called deficiencies hereafter, are then addressed as statements of requirements to the human-factors community (see Figure 5.5). These requirements lead to better training programs, the reorganization of work, and the need for leadership, team communication, and so forth.

Human-factors specialists are introduced via the human element of the triad. Their main axis is the link between the human and the task, which is the hidden axis of systems designers. Through a cognitive analysis, they seek to understand the interaction between the human and the task. They identify how the humans perceive the task, what they have to do to accomplish the task, how they think,

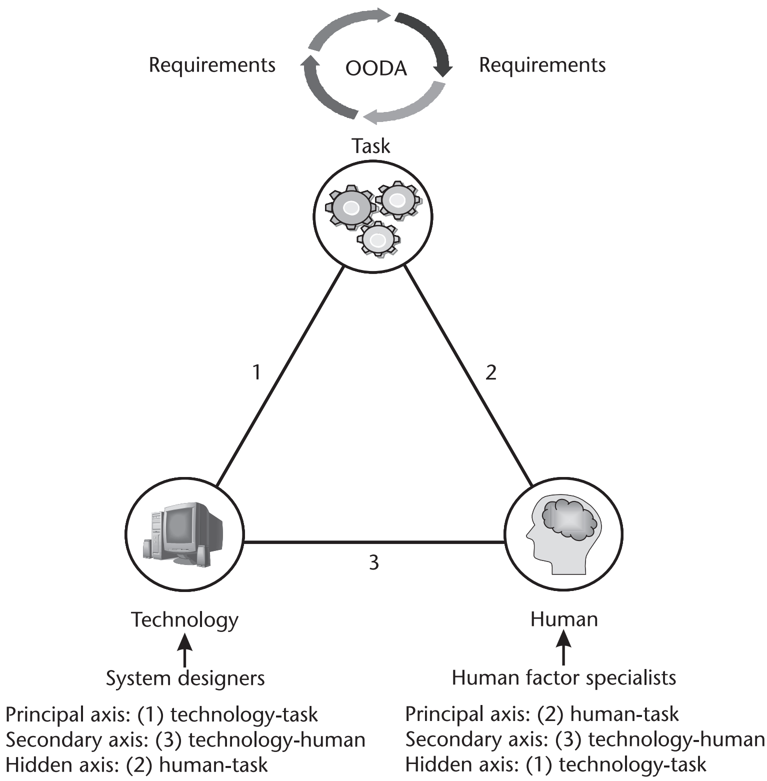


Figure 5.4 Task/human/technology triad model.

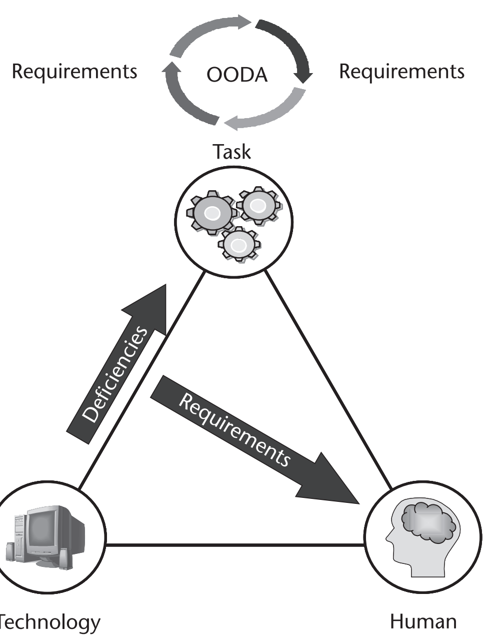


Figure 5.5 Human-factors requirements.

how they apply a skill, what strategies and resources are involved, and what the shortfalls and human limitations are. Their secondary axis of interest is the same as that for the system designers (i.e., the human-technology link), and their hidden axis is the link between the technology and the task, which is the main axis of the system designers. From their analyses, human possibilities and limitations are identified. However, all environmental constraints may not be covered by human possibilities and resources. The uncovered deficiencies are then addressed as statements of requirements to the technological community (see Figure 5.6). These statements become the specification of which part of the task needs support or must be automated, what the system must do, in which conditions, and how the system must interact with the operator.

In this context, everyone involved in the design process has his or her own field of intervention. The weakness of one is the strength of the other. The sets of statements of requirements produced by the system designers and the human-factors specialists are analyzed by a multidisciplinary team involving both communities. This analysis leads to one set of consolidated requirements that determines the nature of the solution (see Figure 5.7). It is very important that both types of specialists work in a close collaboration. Working in isolation will generate requirements formulated by one part that are unrealizable by the other.

Within the context of military or public-security operations, unpredictable events are expected more frequently and are often caused by intelligent sources. The inductive capacity of the human is then required to deal with these events. Some part of the overall system can be automated, but the technological system must mostly be designed to support the human in his or her activities. Hence, the solution cannot be found from a complete technological perspective or a complete human perspective; rather, it must be a mixture of both.

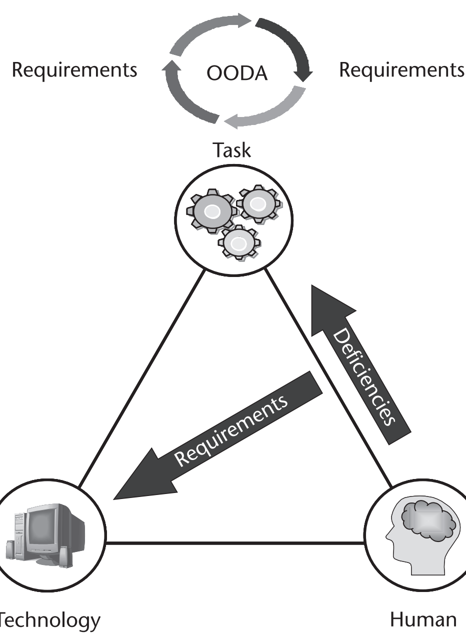


Figure 5.6 Technological requirements.

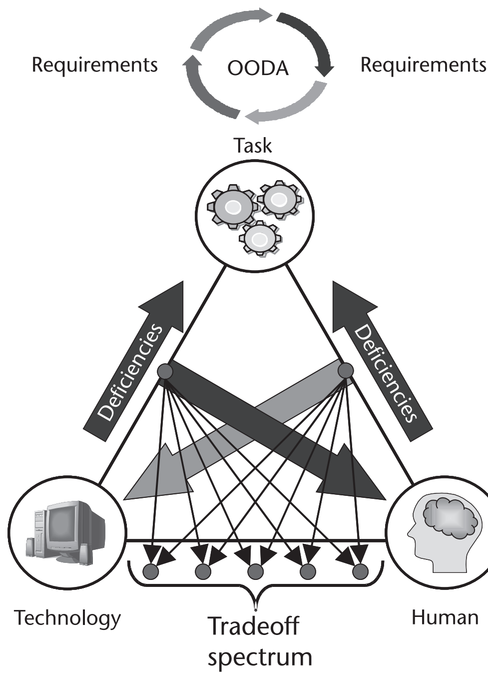


Figure 5.7 Requirements trade-off spectrum.

5.6 Cognitive Fit and Support-System Insertion into the Operational Environment

As one can see from the previous discussion, an important challenge is to develop a support system that, on one hand, takes advantage of all the technological opportunities but, on the other hand, is totally compatible with the way the human executes the decision-making task. Because of the importance of the cognitive fit between the support system and the human, the development of this system must include the participation of system designers and human-factors specialists. The multidisciplinary team must identify which information is required for the optimal execution of the task, as well as when and how it must be presented by the support system to the decision-maker.

Answering these questions and translating them into system concepts is a step toward the development of an effective support system. However, it may not automatically ensure the success of the development process. One must keep in mind that the introduction of any particular new support system in a decision-making environment, for instance, a command-and-control environment, changes the overall dynamic between the human and the decision-making task [15]. This is particularly true when the new system is very different from the old one or when it provides means to execute the task that were not available before. The availability of such a new support system may completely redefine the way the human executes the command-and-control process and, thus, create unpredicted problems that may be more critical than those solved by the system. It is critical to assess the changes brought by the new support system because, from a human-performance perspective, the quality of the overall decision-making process can be reduced, in some circumstances, by the introduction of this new system.

From an operational perspective, the introduction of a new support system may also lead to other important problems. First, it is possible that the training required for the optimal use of the new system is impossible to achieve because of time, money, and or constraints. Second, it is possible that the expertise developed by the decision-maker with the previous systems will become irrelevant in the context of the new procedures brought by the new system. In this particular situation, the introduction of the new support system may bring all of the decision-makers back to the novice level. Third, the changes brought by the new system may not be acceptable from a military-doctrine or rules-of-engagement perspective.

Thus, under some specific circumstances, it may happen that a solution totally compatible with human information-processing capabilities is not acceptable from an operational perspective. Conversely, an optimal solution from an operational point of view may not be adequate to support the decision-maker's performance. Thus, another challenge facing the multidisciplinary development team is to design a support system that is adequate from both the human-performance and the operational perspectives.

To ensure a certain level of adequacy of the support system, the changes brought by this new system must be validated from both perspectives. The validation of the support system can be performed by testing procedures that are integrated into the development process. The goal of these procedures is to answer the following questions:

- Do we create new problems with the insertion of this particular support system?
- Does the system really support and improve human performance?

In the following section, we outline the phases of the development process, using the threat-analysis task as an example. Some potential problems are raised in the execution of this task in the command-and-control environment. We also provide a brief cognitive analysis of these problems, based on known cognitive theories and models, and propose potential technological solutions. Obviously, we are not addressing an exhaustive list of all potential problems, user needs, and solutions related to this task that could be identified. The goal of this specific example is to illustrate some of the phases of the development process. We end the next section by providing reasons explaining the absence of a validation phase in the development process.

5.7 Cognitive Systems Engineering

Within the last decade or so, technological development has raised new issues and challenges regarding the design process of fusion, situation-analysis, and decision-support systems. Actually, the type of issues and challenges has shifted from identifying the technological possibilities and limitations to determining how these systems must be designed to fit with human information processing. This situation has contributed to the emergence of the human-factors community and the development of engineering methods for cognitive systems.

Cognitive engineering is an interdisciplinary approach to designing computerized systems intended to support human performance [16]. It encompasses the fields of human factors, human-computer interaction, cognitive psychology, computer science, artificial intelligence, and other, related fields. The methods of cognitive engineering consider workers and the tasks they perform as the central drivers for system design. These methods are quite diverse [17]. Certain methods aim to get an understanding of users and tasks by constructing quantitative models of expert reasoning. Other methods focus on documenting the key decisions made in the domain and the information required to make those decisions. The aim is to develop systems that support cognitive functions.

Questions that drive the design of fusion, situation analysis, and decision-support systems and that are addressed by methods of cognitive engineering include [17]:

- What are the goals and constraints of the application domain?
- What range of tasks do domain practitioners perform?
- What strategies do they use to perform these tasks today?
- What factors contribute to task complexity?
- What tools can be provided to facilitate the work of domain practitioners and achieve their goals more effectively?

The methods of cognitive engineering have tremendous potential to impact some of the most difficult aspects of system engineering, especially in the command-and-control domain. In fact, it has been suggested that the only way to deal with the increased complexities in future command-and-control systems, including the vast amounts of available data, the pressure to make timely decisions using the totality of that data, and reduced manpower and cost goals, is to follow a human-centered approach to system engineering [18].

Cognitive engineering methods can be organized into five primary categories:

1. Modeling cognitive processes;
2. Modeling behavioral processes;
3. Describing cognitive and behavioral activities;
4. Modeling erroneous actions;
5. Modeling human-machine systems.

While some methods overlap multiple categories, each method is assigned to a "primary" category. Bonaceto and Burns [17] further discuss these methods.

5.7.1 Information Requirements for Optimal Decision-Making

With the technological evolution, it is possible to create more powerful and sophisticated technological systems that make available to decision-makers a huge amount of data and information about situations and the environment in large-scale military and public-security operations. Hence, a typical problem with today's systems is not a lack of information but finding what is needed when it is needed [19]. Providing as much data and information as possible about a situation and its environment is not necessarily an adequate way to support the decision-maker's performance. All of the data and information available is not relevant and useful for reaching an optimal decision.

In fact, in some situations, most of the data can be seen a distraction or as noise by the decision-maker and may thus reduce his or her level of SAW. For instance, when a given system makes available some sources of information that are superfluous and that bring noise or confusion to the situation, the decision-maker still has to devote perceptual and attentional resources, which are limited, to scan and select the relevant information from these extraneous sources of data. The presence of these distracters may also increase the workload imposed on the short-term memory required in the processing of multiple pieces of data.

Typically, only a small fraction of the overall data and information available is relevant and useful for the decision-making process. The identification of the critical information can be done through various procedures or analysis techniques developed to identify cognitive processes.

5.7.2 Mental Models and Cognitive Fit

The situation-awareness processes described by Endsley are often initiated by the presence of an object in the perceiver's environment. However, such processes can also be triggered by a priori knowledge, feelings, or intuitions. In these situations, hypotheses related to the possible presence of an object are formulated. The perceiver then initiates search processes in the environment that confirm or invalidate these hypotheses. Note that this type of SAW is possible only if mental models related to the possible objects are available. In her theory of SAW, Endsley clearly presumes that patterns and higher-level elements are present, according to which the situation can be structured and expressed. SAW can be interpreted as the operator's mental model of all pertinent aspects of the environment (processes, states, and relationships).

There is a tight link between this mental model used to structure and express situation elements and the cognitive processes involved in achieving the levels of awareness. This link is known as the cognitive fit and requires an understanding of how the human perceives a task, what processes are involved, what the human's needs are, and what part of the task can be automated or supported. This understanding is crucial and only achieved via a number of specialized human-factors investigations known as cognitive engineering analyses.

Such analyses are generally conducted by the human-factors engineering community. Human-factors engineering can be seen as the U.S. counterpart of ergonomics. According to Preece et al. [20], cognitive ergonomics is a discipline that focuses particularly on human information processing and computer systems. By definition, it aims at developing knowledge about the interaction between human information-processing capacities and limitations and technological information-processing systems.

The usefulness of an information system is closely related to its compatibility with human information processing. Hence, such a system must be developed according to human needs, especially regarding information processing. A first step is the identification of the cognitive processes involved in the execution of a task. Many procedures have been developed to identify those processes. Jonassen, Hannum, and Tessmer [21] describe task analysis as a process that is performed in many ways, in a variety of situations, and for multiple purposes. Such an analysis determines what the performers do, how they perform the task, how they think, or how they apply a skill.

5.7.3 Cognitive Task Analysis and Cognitive Work Analysis

Among the procedures developed to identify cognitive processes and provide, via interviews with SMEs, the set of critical information that must be made available to reach optimal decisions, cognitive task analysis (CTA) [22] and cognitive work analysis (CWA) are both often put forward.

There are differences between these two procedures. A CTA is concerned with informing the information-system design process through the application of cognitive theories [20]. It is used to elicit and capture the knowledge and processing used by experts in performing their jobs. A CTA often begins with high-level descriptions of the task based on observations and interviews. However, the bulk of the data collection occurs via in-depth interviews with SMEs.

The CWA can be seen as a broader analysis than the CTA. According to Vicente [23], traditional task-analysis methods typically result in a single temporal sequence of overt behavior. Such a description represents the normative way to perform the task. Unfortunately, traditional methods cannot account for factors like: (1) changes in initial conditions, (2) unpredictable disturbances, and (3) the use of multiple strategies. The use of traditional task analysis brings an artifact that will only support one way to perform the task.

Vicente proposes an ecological approach in which the three factors above are considered. The ecological approach, which can be seen as a CWA, takes its roots in psychological theories that were first advanced by Brunswick [24] and Gibson [25, 26]. These researchers raised the importance of studying the interaction between the human organism and its environment. The perception of an object in its environment is a direct process, in which information is simply detected rather than constructed [26]. The human and the environment are coupled and cannot be studied in isolation. A central concept of this approach is the notion of *affordance*, an aspect of an object that makes it obvious how the object is to be used. Examples are a panel on a door to indicate "push," and a vertical handle to indicate "pull" [20]. When the affordance of an object is obvious, it is easy to know how to interact with it. The environment in which a task is performed has a direct influence on overt behavior. Hence, the ecological approach begins by studying the constraints in the environment that are relevant to the operator. These constraints influence the observed behavior [27].

The ecological approach is comparable to, and compatible with, Rasmussen's abstraction hierarchy framework. Rasmussen's framework is used for describing the functional landscape in which behavior takes place in a goal-relevant manner.

This abstraction hierarchy is represented by means-ends relations and is structured in several levels of abstraction that represent functional relationships between the work-domain elements and their purposes. With the ecological approach, Rasmussen has developed a comprehensive methodology for CWA that overcomes the limitations of traditional CTA by taking into account the variability of performance in real-life, complex work domains. For these reasons, the CWA seems to be a good choice to answer questions related to understanding complex tasks like information fusion and situation analysis for command and control in large-scale military and public-security operations.

5.7.4 Applied Cognitive Work Analysis (ACWA)

Unfortunately, although CTA and CWA are well suited to deal with fusion, situation-analysis and decision-support issues, they are in practice very expensive to conduct, time-consuming and, more importantly, generally inefficient from a design-process perspective. With the latter limitations in mind, a pragmatic cognitive systems engineering (CSE) approach, known as the Applied Cognitive Work Analysis (ACWA), has been developed to bridge, in a structured, efficient, and converging way, the gap between cognitive analysis and design [28].

This ACWA modeling method is a pragmatic adaptation of the CWA method in order to cope with the limitations related to applying CWA. As a result, the cost to conduct CSE analyses using the ACWA approach is reduced, and the analysis-design efficiency is significantly improved, making easier the identification of decision-aiding concepts suited to provide effective decision support. Figure 5.8 provides a visual depiction of the sequence of methodological steps and their associated output artifacts, as well as an indication that the process is typically repeated in several expanding spirals, each resulting in an improved support system. Elm et al. [29] describe each step of this approach in detail.

5.8 Development Process of a Support System

As discussed in [1], the development process for a support system can be described as follows (see Figure 5.9):

- Interview of SMEs in the context of a cognitive analysis (e.g., CTA, CWA, ACWA);
- Identification of decision-making requirements through the identification of user needs, problems, and deficiencies from the cognitive analysis;
- Use of current cognitive theories and models to understand the meaning of these requirements from a human-factors perspective;
- Identification of technological solutions to address these requirements;
- Testing procedures to validate the technological solutions from the human-performance and operational perspectives.

To illustrate the support-system development process, we consider, as an example, the development of a system built to support the execution of a threat-analysis

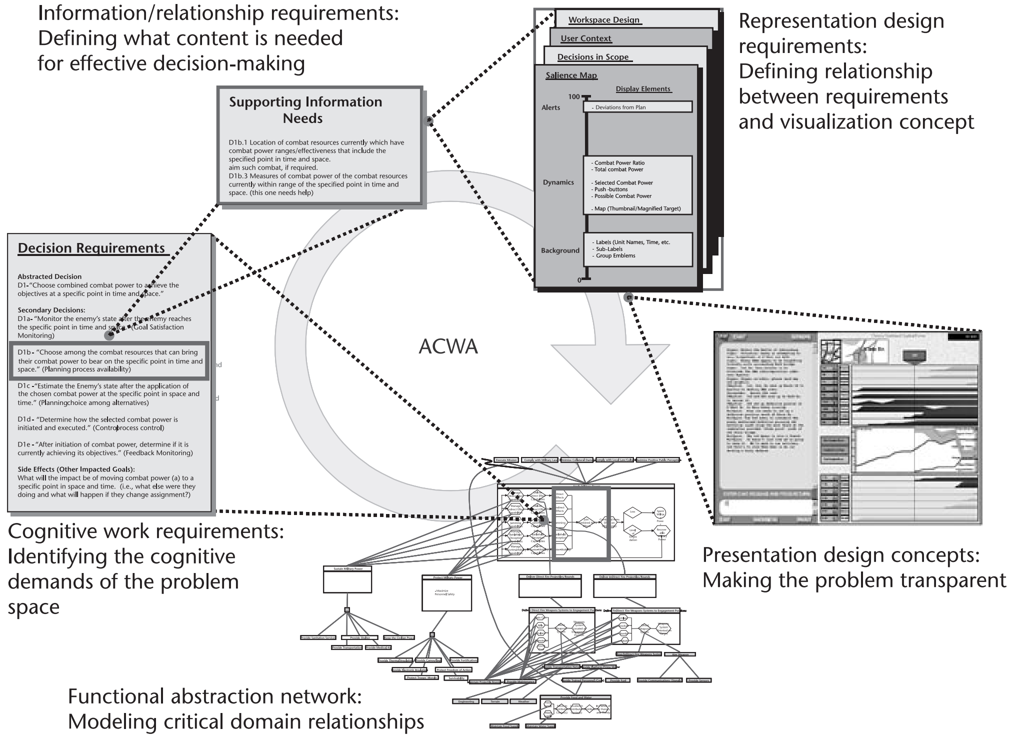


Figure 5.8 ACWA [29].

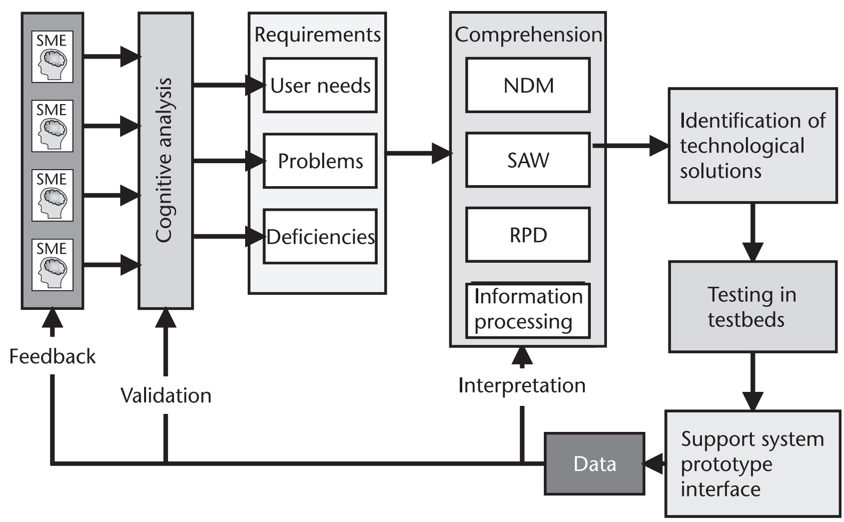


Figure 5.9 The support-system development process.

task by a decision-maker in the naval context. While conducting antiair warfare, threat analysis is particularly difficult because the available information is often incomplete or ambiguous [30]. Moreover, the uncertainty related to the information can also be voluntarily created by an intelligent source (e.g., the enemy). Stress factors, such as the possibility of blue-on-blue (friendly) or blue-on-white (neutral) engagements, social pressure, and political issues are added to make the situation even more complex and stressful.

5.8.1 A Cognitive Analysis of the Problems

Through interviews with SMEs while conducting a cognitive analysis, some problems and deficiencies are identified. For instance, one problem, which is likely to happen, can be reported by the SMEs and described as follows [29]:

> The decision maker is rapidly overloaded by the presence of multiple entities that are potentially threatening. In these complex situations, according to the SMEs interviewed, it can be very difficult to perceive and recognize all the relevant information and rank all these entities properly in function of their lethality, level of threat, their importance, etc. The situation can become rapidly overwhelming for the decision maker.

To design an adequate technological system supporting the SMEs in the execution of the threat-analysis task, the problem cited in the previous subsection must be interpreted in terms of appropriate cognitive theories and models, such as the NDM, the RPD and the SAW models. These theories and models have been developed to represent human behaviors in the execution of complex tasks, such as situation analysis and decision-making, which are critical in a complex command-and-control environment. Examples of such cognitive analyses proposed by the development team could be cited as follows:

- According to the RPD model, the critical phase of the decision-making process is the recognition of relevant features in the environment. In a complex and dynamic environment, the decision-maker does not have time to weigh all of the alternatives and select the optimal one. Instead, the expert rapidly selects an appropriate course of action according to the features immediately perceived and recognized in the environment. The quality of the course-of-action selection process is highly related to the features-recognition process and the level of expertise of the decision-maker. This level of expertise is related to the amount of relevant knowledge stored in the expert's long-term memory. In a complex situation involving multiple potential threats, a large number of critical features related to these potential threats must be perceived and recognized rapidly to select an appropriate course of action. Thus, it is likely that the attentional resources required for the processing of these numerous features will exhaust the attentional resources available at the moment. As a result, it is possible that some critical features required for an optimal execution of the task will be omitted, ignored, misjudged, or simply not perceived.
- From the perspective of the SAW model, the lack of attentional resources results in a low level of SAW. Some critical features are not perceived; consequently, they are not available for subsequent SAW processes, such as the comprehension of the situation and the prediction of future states that

lead to the selection of the course of action. As a result, the decision-maker's SAW level is considered to be low. Unfortunately, there is a strong link between SAW and the quality of decision-making [19].
- The match between the features perceived from the environment and the knowledge stored in the long-term memory of the decision-maker takes place in the working memory. Besides this recognition process, which is well defined by the RPD model, the working memory is also involved in some SAW processes, such as the comprehension of the situation (level 2) and the projection of future states (level 3). The working memory has only limited processing capacities. Thus, this structure can be rapidly overloaded by a situation involving multiple entities.

After an adequate cognitive analysis of the problem, the development team must address a list of potential technological solutions to support the SMEs in the execution of the threat-analysis task.

5.8.2 Technological Solutions

As a result of the cognitive analysis of the problem(s), user requirements are better understood. Here again, we are not providing an exhaustive list of all possible user requirements that could be identified following a cognitive analysis. For the purpose of our example, only a few requirements are provided:

- The recognition process of the features must be supported.
- The workload imposed on the working memory must be reduced.

From the identification and comprehension of these requirements, some technological solutions can be proposed:

- Use different symbols and colors to improve the recognition process of features.
- Use automated warning signals and pop-up windows to draw the expert's attention to relevant critical features in the environment. This will improve the expert's SAW level.
- Use data fusion to gather relevant features related to a given contact. This will reduce the workload imposed on the working memory by reducing the number of features to consider.

5.8.3 Validation of Technological Solutions

In a serious and efficient support-system development process, the technological solutions proposed to address the user requirements are based on the results of an "in-depth" cognitive analysis. Consequently, one may claim that they are probably appropriate to support the human in the execution of the task. This is likely true. However, as mentioned above, the introduction of a new support system redefines the overall dynamic between the human and the task and, as a result, can produce other problems that can be more critical than those solved by the system. It is also possible that the support provided by the system only generates a nonsignificant improvement in performance. In this latter situation, the marginal improvement may not be worth the cost and time spent to insert the system into the environment. Thus, as illustrated in Figure 5.10, before introducing the support system into the SME environment, the development team must determine the level of improvement generated by the technological solution and verify whether this solution creates other problems or is compatible from an operational perspective. In other words, the technological solution must be validated from both the human-performance and the operational perspectives.

To determine the level of improvement to human performance generated by the solution and its compatibility from an operational perspective, an experimental environment must be created in which empirical evidence is obtained. Two characteristics are essential to this environment:

- The experimental environment must be realistic enough to allow for the generalization of the results to the real environment. The experimental setting must reproduce the conditions in which the SME executes the task.
- The experimental environment must allow for a systematic manipulation of the variables of interest. To interpret the results, the development team must be able to establish a clear link between the observed results and the manipulation of the variables of interest. It is important to eliminate as much as possible the influence of any extraneous variables on the results.

For reasons such as funding and time constraints, the validation step is often skipped in the support-system development process. In such constrained circumstances, the validation of the system is simply done through its use in real-world situations. However, it can be particularly risky to bring a new support system into the real-world environment without completely knowing its effects from the human performance and the operational perspectives. The use of an ineffective support system may encourage the decision-maker to bypass or ignore the inputs from the system. In the worst case, the use of an inappropriate system may result in dramatic consequences, such as the loss of lives.

Another important explanation for the absence of the validation process from the support-system development process is the lack of appropriate tools to validate

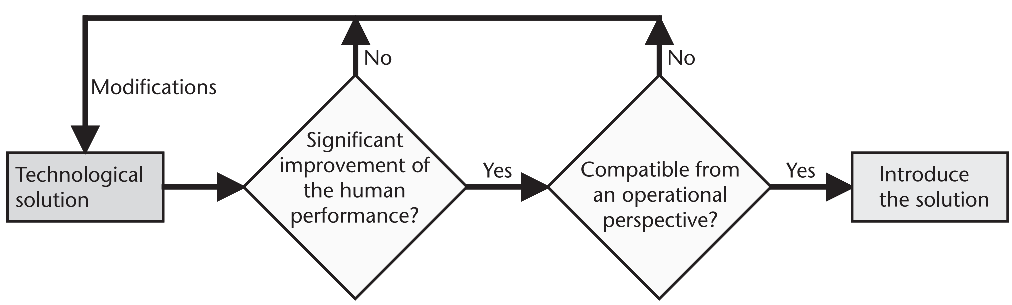


Figure 5.10 The validation of technological solutions.

the system. Field trials and laboratory settings that can be used to validate the support system are both subject to important restrictions. On one hand, there is a frequent complaint related to field trials about the presence of extraneous variables that threaten the validity of the experiment. On the other hand, one can claim that controlled procedures in laboratory settings make the experimental setting too artificial and therefore make the results difficult to generalize to real-world situations. Results obtained in a very controlled environment can hardly be extrapolated to an unstable and uncertain environment. For instance, display concepts developed in laboratory settings may not be appropriate in unstable, uncertain, and ill-defined environments such as command and control in military and public-security operations.

References

[1] Breton, R., J. Roy, and S. Paradis, *A Command Decision Support Interface for Human Factors and Display Concept Validation*, DREV-TR-2001-218, Valcartier, Canada, 2002.

[2] Proctor, R. W., and T. Van Zandt, *Human Factors in Simple and Complex Systems*, Needham Heights, MA: Allyn and Bacon, 1994.

[3] Proctor, R. W., and T. Van Zandt, *Human Factors in Simple and Complex Systems*, New York: John Wiley & Sons, 1983.

[4] Riffenburgh, R. H., *Tactical Decision Making Under Stress Program Study: Initial Design*, San Diego, CA: Naval Ocean Systems Center, 1991.

[5] Wickens, C. D., "Processing Resources in Attention," in R. Parasuraman and R. Davies, (eds.), *Varieties of Attention*, New York: Academic Press, 1984, pp. 63–102.

[6] Pashler, H. E., *The Psychology of Attention*, Cambridge, MA: MIT Press, a Bradford Book, 1998.

[7] Haberlandt, K., *Cognitive Psychology*, 2nd ed., Boston, MA: Allyn and Bacon Publishers, 1997.

[8] Schneider, W., and R. M. Shiffrin, "Controlled and Automatic Human Information Processing: I. Detection, Search, and Attention," *Psychological Review*, Vol. 84, 1977, pp. 1–66.

[9] Rasmussen, J., "Skills, Rules, Knowledge: Signals, Signs, and Symbols and Other Distinctions in Human Performance Models," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-13, No. 3, 1983, pp. 257–267.

[10] Dreyfus, H. L., "Intuitive, Deliberative, and Calculative Models of Expert Performance," in C. E. Zsambok and G. Klein, (eds.), *Naturalistic Decision Making*, Mahwah, NJ: Lawrence Erlbaum, 1997, pp. 17–28.

[11] Schneiderman, B., *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 2nd ed., Reading, MA: Addison-Wesley, 1992.

[12] Ballas, J. A., et al., *Interfaces for Intelligent Control of Data Fusion Processing*, Naval Research Laboratory, NRL/FR/5513-96-9806, April 1996.

[13] Bainbridge, L., "The Ironies of Automations," *Automatica*, Vol. 19, No. 6, 1983, pp. 775–779.

[14] Chalmers, B., "On the Design of a Decision Support System for a Data Fusion and Resource Management in a Modern Frigate," *NATO Symposium on Sensor Data Fusion and Integration of the Human Element*, Ottawa, Canada, September 13–17, 1998.

[15] Breton, R., R. Rousseau, and W. Price, *The Integration of Human Factor in the Design Process: A Triad Approach*, DREV-TM-2001-002, Valcartier, Canada, 2002.

[16] Roth, E. M., E. S. Patterson, and R. J. Mumaw, "Cognitive Engineering: Issues in User-Centered System Design," in J. J. Marciniak, (ed.), *Encyclopedia of Software Engineering*,

2nd ed., New York: John Wiley & Sons, 2001, http://mentalmodels.mitre.org/cog_eng/ce_intro.htm.

[17] Bonaceto, C. A., and K. J. Burns, *A Roadmap for Cognitive Engineering in Systems Engineering*, Bedford, MA: The MITRE Corporation, 2004, http://mentalmodels.mitre.org/Hyperlinks/Methods_and_Uses.htm.

[18] Perry, A., et al., *The Solution for Future Command and Control: Human-Centered Design*, Office of Naval Research Manning and Affordability Initiative, 1999, http://mentalmodels.mitre.org/cog_eng/ce_sys_eng_challenges.htm.

[19] Endsley, M. R., and D. J. Garland, *Situation Awareness Analysis and Measurement*, Mahawah, NJ: Lawrence Erlbaum Associates, 2000.

[20] Preece, J., et al., *Human-Computer Interaction: Concepts and Design,* Reading, MA: Addison-Wesley, 1994.

[21] Jonassen, D. H., W. H. Hannum, and M. Tessmer, *Handbook of Task Analysis Procedures*, New York: Praeger Publishers, 1989.

[22] Matthews, M. L., R. D. G. Webb, and D. J. Bryant, *Cognitive Task Analysis of the Halifax-Class Operations Room Officer*, Human Systems Inc., PWGSC Contract No. W7711-7-7404/001/SV, DCIEM Scientific Authority, Carol McCann, March 10, 1999.

[23] Vicente, K. J., *A Few Implications of an Ecological Approach to Human Factors*, Mahwah, NJ: Lawrence Erlbaum, 1995.

[24] Brunswick, E., *Perception and the Representative Design of Experiments*, 2nd ed., Berkeley, CA: University of California Press, 1956.

[25] Gibson, J. J., "The Problem of Temporal Order in Stimulation and Perception," *Journal of Psychology*, Vol. 62, 1966, pp. 141–149.

[26] Gibson, J. J., *The Ecological Approach to Visual Perception*, Hillsdale, NJ: Lawrence Erlbaum, 1979.

[27] Rasmussen, J., *Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering*, New York: North-Holland Publishing, 1986.

[28] Paradis, S., et al., "A Pragmatic Cognitive System Engineering Approach to Model Dynamic Human Decision-Making Activities in Intelligent and Automated Systems," *Proceedings of NATO HFM Symposium on the Role of Humans in Intelligent and Automated Systems*, Warsaw, Poland, October 7–9, 2002.

[29] Elm, W. C., et al., "Applied Cognitive Work Analysis: A Pragmatic Methodology for Designing Revolutionary Cognitive Affordances," in Hollnagel, E., (ed.), *Handbook of Cognitive Task Design*, Mahwah, NJ: Lawrence Erlbaum, 2002.

[30] Hutchins, S. G., *Principles for Intelligent Decision Aiding*, Naval Command, Control, and Ocean Surveillance Center, RDT&E Division, Technical Report 1718, San Diego, CA, 1996.

CHAPTER 6
Knowledge, Belief, and Uncertainty

Anne-Laure Jousselme, Patrick Maupin, Éloi Bossé, and Dale Lambert

6.1 Introduction

Knowledge, belief, and uncertainty are three key notions of the situation-analysis process (through data and information fusion). As discussed in previous chapters, the two basic elements involved in situation awareness are the situation and the person. The situation can be defined in terms of events, entities, systems, other persons, and the like, as well as their mutual interactions. The person can be defined according to the cognitive processes involved in situation awareness. The cognitive process of the human must be supported to help him or her build his mental model (understanding) of the situation. To this end, the human uses observing devices or agents (sensors or other humans) and computers (processing) to support his or her reasoning about the situation. Figure 6.1 illustrates the main challenge in situation analysis. Belief and knowledge representation is a crucial step in transforming data into knowledge. The data and information coming from the different sources must be converted into a certain language or other information format (e.g., visualization)

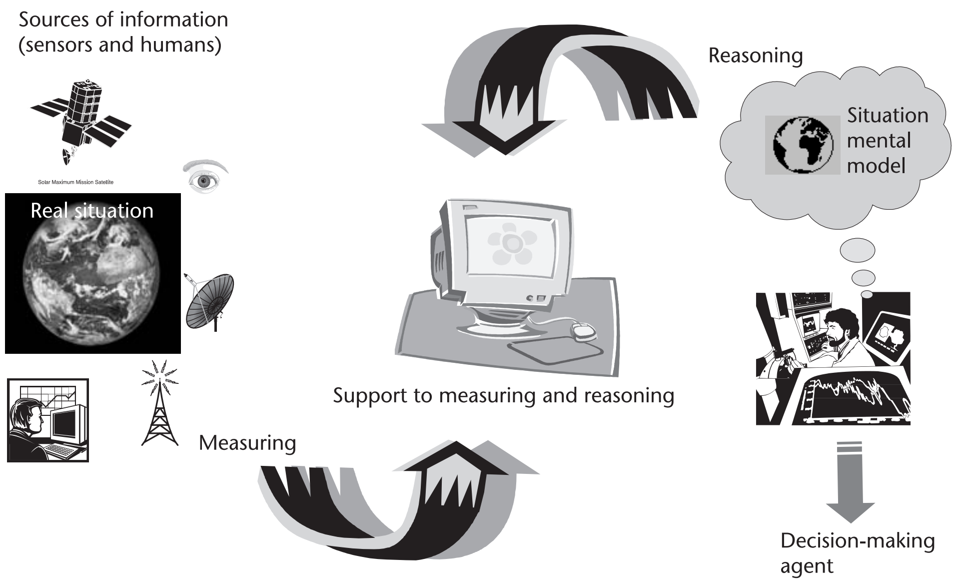


Figure 6.1 Measuring and reasoning about the situation.

so that they can be processed and used by the human to build a mental model in order to decide and act. One great challenge in designing a support system is to make use of the mathematical and logical tools that can allow measuring and reasoning about the situation using a common analysis framework. This chapter discusses the key notions of knowledge, belief, and uncertainty in relation to information fusion.

6.2 Knowledge and Belief

Following [1], we assume that there are real situations in the world. Through interaction with the world, people seek situation awareness by forming perceptions, comprehensions, and projections about these situations. Hereafter, these perceptions, comprehensions, and projections are characterized as *beliefs*, where beliefs can be understood from either a realist [2, 3] or instrumentalist perspective [4, 5]. In the previous chapters, we also proposed automated information-fusion systems as computational systems aiming to form perceptions, comprehensions, and projections within a machine. It is convenient likewise to think of these as beliefs. These machine-based beliefs can then also be understood from the realist or instrumentalist perspectives. A machine that explicitly represents its beliefs symbolically illustrates the realist perspective. Alternatively, a robot engineered through a subsumption architecture [6] might be ascribed beliefs instrumentally on the basis of its behavior, without their being explicitly represented anywhere within the machine.

Whether explicitly or tacitly conceived, beliefs are customarily described through sentences of the form "x believes that σ" (e.g., Fred believes that it is raining), where σ expresses some claim about the world and x identifies an individual having a belief attitude toward that claim. Consequently, we interpret beliefs *propositionally* (without all of its Fregean connotations [7]) as mental states that are ascribed to an individual (human or machine) and that make some truth-functional claim about the world. Propositional expressions (e.g., SAM#1 is targeting F18#7) will be used to express the propositions associated with these mental states, while propositional attitude expressions (e.g., AEW#2 believes that SAM#1 is targeting F18#7) will be used to express the association of a propositional belief state with an individual. To illustrate, for a machine AEW#2 that explicitly represents its beliefs, we might:

- Identify a belief state with a belief data structure (an instance of a particular type of data structure);
- Identify the proposition that surface-to-air missile number 1 is targeting F18 number 7 with that belief state if that data structure contains the string "SAM#1 targeting F18#7";
- Express the proposition that surface-to-air missile number 1 is targeting F18 number 7 through the propositional expression SAM#1 is targeting F18#7;
- Identify the association of the belief that surface-to-air missile number 1 is targeting F18 number 7 with machine AEW#2 if machine AEW#2 has stored the belief data structure containing the string "SAM#1 targeting F18#7";

- Express AEW#2's propositional attitude through the propositional attitude expression AEW#2 believes that SAM#1 is targeting F18#7.

Situation awareness is usually associated with having perception, comprehension, and projection knowledge about the world. Philosophers have struggled to produce a cogent definition of knowledge for centuries. While Plato's "justified true belief" fails to define knowledge adequately, the elements of truth, justification, and belief in some way figure prominently in many accounts of knowledge [8]. For that reason, those elements serve as the cornerstones of the approach presented in this book. We accept belief of σ as a necessary, but insufficient, condition for knowledge of σ.

True beliefs are valued over false beliefs because they offer greater utility in dealing with the world. Truth, like knowledge, remains a philosophically allusive commodity, with the correspondence, coherence, pragmatist, and Tarskian theories being among the more dominant. For our mathematical theory, we will primarily present a Tarskian account of truth, according to which a sentence is true if, and only if, its metalinguistic interpretation is true. Thus, in keeping with conventional model theory [9], we might:

- Interpret the token SAM#1 as the object A(SAM#1) in the world;
- Interpret the token F18#7 as the object A(F18#7) in the world;
- Interpret the token is targeting as a set of object pairs A(is targeting), which is the set of all targeting pairs in the world;
- Then assert that the propositional expression SAM#1 is targeting F18#7 is true if, and only if, $\langle$A(SAM#1), A(F18#7)$\rangle \in$ A(is targeting).

If we assume that the world consist of situations that can be conceptualized in terms of A(SAM#1), A(F18#7), and A(is targeting), then the truth of $\langle$A(SAM#1), A(F18#7)$\rangle \in$ A(is targeting) is decided by whether or not it characterizes a real situation in the world.

Conceptualization figures prominently in this formulation of truth. If we cannot identify the object A(SAM#1), the object A(F18#7), or the set (relation) A(is targeting), or whether $\langle$A(SAM#1), A(F18#7)$\rangle \in$ A(is targeting) is the case, then we have no basis for deciding the truth of the propositional expression SAM#1 is targeting F18#7. Of these, the relation A(is targeting) is the most interesting and difficult to identify, for we would not ordinarily understand the concept of targeting by thinking of all instances of something targeting something else. In practice, we are more inclined to understand the concept of targeting by relating it to other concepts, either by defining targeting in terms of those other concepts or by proposing a collection of constraints that relate the targeting concept to other concepts. Attempts to specify conceptualizations formally in this way have led to the recent research activity in ontologies, and although much of this research was initially concerned with simple conceptual taxonomies, it is now evident that richer conceptual specifications are required. In this way, the concept of targeting can be formally specified through a set of formal sentences (formal theory) formed from terms for other concepts, like time, distance, enemy, intent, and so forth, rather than by pretending to contemplate all instances of targeting between objects.

The ontologies-as-formal-theories approach allows us to specify what we mean by targeting, for example. The value of this maneuver comes when we introduce an inference relation $\vdash$. An inference relation identifies rules for deducing formal expressions from sets of formal expressions, and it is customary to write $\Sigma \vdash \tau$ whenever τ can be deduced from Σ. The choice of an inference relation H is far from arbitrary.

- An inference relation $\vdash$ is *sound* if it is truth preserving; that is, if $\Sigma \vdash \tau$, then whenever every sentence in Σ is true, the sentence τ must also be true.
- An inference relation $\vdash$ is *complete* if it recovers all truthful conclusions; that is, if whenever every sentence in Σ is true, the sentence τ must also be true, then $\Sigma \vdash \tau$.

Similarly, wherever possible it is advantageous for each formal theory Σ to be consistent and complete, where:

- Σ is *consistent* if there is no τ such that $\Sigma \vdash \tau$ and $\Sigma \vdash \neg\, \tau$.
- Σ is *complete* if for every τ, $\Sigma \vdash \tau$ or $\Sigma \vdash \neg\, \tau$.

We are then able to determine the truth of propositional expressions, without appealing to the interpretations of their terms, by reasoning with the inference relation to determine what must be true. As equivalently noted in [10], by combining some domain theory D of beliefs about the world with a theory M expressing the meaning of the terms used in the domain theory, the consequences of D can then be deduced as $\{\tau \mid (D \cup M) \vdash \tau\}$. A consistent and complete theory M will ensure that inconsistencies and ignorance in our data-fusion system derive from the beliefs D about the world. In this way, the inference relation $\vdash$ provides the remaining element of *justification*. If the beliefs in D are all true, then the beliefs in $\{\tau \mid (D \cup M) \vdash \tau\}$ must also be true; moreover, they are justified because the inference relation $\vdash$ can be used to explain why they must be true! The three elements of belief, truth, and justification can be related in this way to provide a concept of knowledge for our automated fusion systems. Ontologies as formal theories will become a significant and fundamental element of the mathematical foundation of information fusion.

6.3 Uncertainty

Uncertainty is a widely used term within the artificial intelligence and engineering communities. However, the authors in these fields of application and research do not always agree on the meaning of the word "uncertainty," its different types, its possible sources, its synonyms, its possible classifications, its representations, and so forth. In this section, we explore the concept of uncertainty and related concepts, such as imperfection, imprecision, vagueness, ambiguity, incompleteness, ignorance, and the like. We start with sociological points of view [11, 12], before describing the artificial intelligence and engineering accounts of uncertainty from the last 10 years [13–16].

6.3.1 Bronner's Sociological Point of View

Sociologist Gérald Bronner [11] distinguishes two kinds of uncertainty: *uncertainty in finality* (or material uncertainty) and *uncertainty of sense*. Uncertainty in finality is "the state of an individual that, wanting to fulfill a desire, is confronted to the open field of the possibles" (e.g., Will my car start? Am I ill?). In contrast, uncertainty of sense is "the state of an individual when a part or the whole of its systems of representation is deteriorated." Uncertainty in finality is uncertainty about goal outcomes, while uncertainty of sense is uncertainty of meaning. In situation analysis, agents are confronted with uncertainty of sense (data driven) from the bottom-up perspective and with uncertainty in finality (goal driven) from the top-down perspective. Bronner further classifies *uncertainty in finality* into three types, according to one's capacity both for uncertainty and for avoiding it:

- *Situation of type I:* Uncertainty does not depend on the agent and cannot be avoided.
- *Situation of type II:* Uncertainty does not depend on the agent but can be avoided.
- *Situation of type III:* Uncertainty is generated by the agent and can be avoided.

6.3.2 Smithson's Taxonomy of Ignorance

Smithson [12] proposes a taxonomy of ignorance where uncertainty appears as a kind of ignorance, "one of the most manageable kinds of ignorance." This taxonomy is reproduced in Figure 6.2.

Smithson interprets ignorance as nonknowledge. He initially separates ignorance into two categories: the *state of ignorance* (error) and the *act of ignoring* (irrelevance). The latter corresponds to a deliberate action to ignore something irrelevant to the problem-solving situation, whereas the former is a state (of ignorance) resulting from different causes (distorted or incomplete knowledge). For Smithson, uncertainty is incompleteness in degree (as compared to absence, which

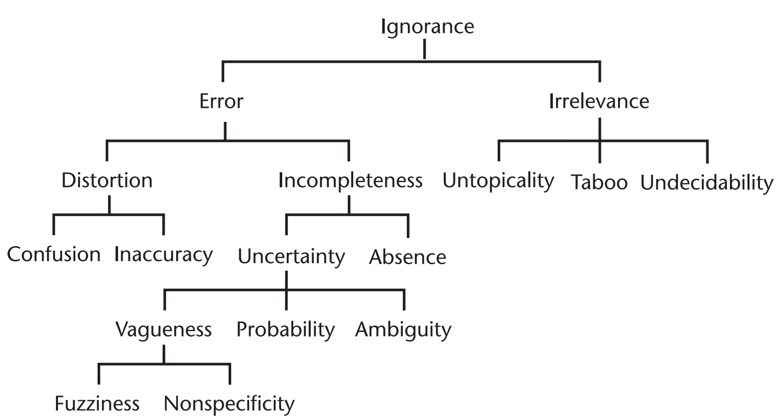


Figure 6.2 Taxonomy of ignorance according to Smithson [12].

is incompleteness in kind) and is subdivided into three types: *probability*, *vagueness* (being either nonspecificity or fuzziness), and *ambiguity*.

6.3.3 Krause and Clark's Uncertainty Classification

Krause and Clark [13] propose an alternative typology of uncertainty to Smithson's, centered on the concept of uncertainty. Krause and Clark distinguish two aspects: *unary* (i.e., uncertainty applied to individual propositions) and *set theoretic* (i.e., uncertainty applied to sets of propositions). Both categories lead either to *conflict* (conflicting knowledge) or *ignorance* (lack of knowledge). As subcategories, we find vagueness, confidence, propensity, equivocation, ambiguity, anomaly, inconsistency, incompleteness, and irrelevance. This model is reproduced in Figure 6.3.

Compared to Smithson's taxonomy, Krause and Clark's taxonomy [13] adds the unary–set theoretic dichotomy in order to introduce the concept of inconsistency and to move the concept of incompleteness in the set theoretic branch.

6.3.4 Bouchon-Meunier and Nguyen's Model

Bouchon-Meunier and Nguyen [14] propose a model for uncertainty (Figure 6.4). They refer to uncertainty as "imperfection on knowledge" and denote three main types of imperfection:

- Probabilistic uncertainty;
- Incompleteness in knowledge (e.g., belief, general laws, imprecision);
- Vague and imprecise description.

The scheme of Figure 6.4 distinguishes between the two general senses of uncertainty. Reading the graph from right to left, uncertainty appears as a final state (of mind) possibly caused by belief, general laws, imprecision, vagueness, or

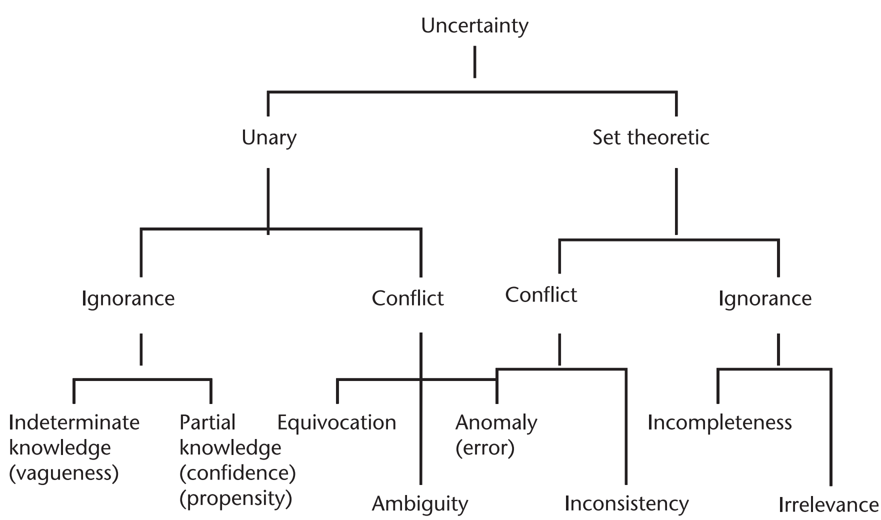


Figure 6.3 Uncertainty model according to Krause and Clark. (*After:* [13].)

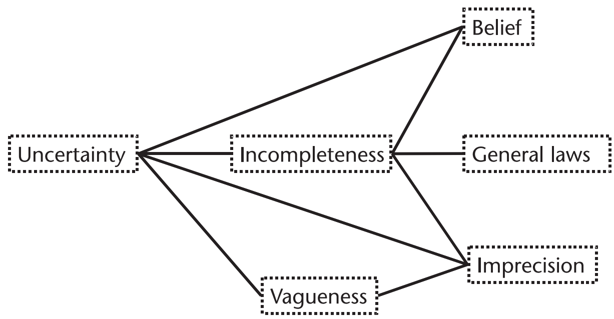


Figure 6.4 Types of uncertainty according to Bouchon-Meunier and Nguyen. (*After:* [14].)

incompleteness. This refers to sense I, with uncertainty as a mental state. Reading the same graph from left to right supposes that incompleteness, vagueness, and so forth, are kinds of uncertainty, which engenders sense II, where uncertainty is a feature of information being of many kinds.

6.3.5 Types of Uncertainty According to Klir and Yuan

The typology of uncertainty proposed by Klir and Yuan [15] is built upon the different existing mathematical theories of uncertainty. After a description of the measures of uncertainty available within the theories, Klir and Yuan propose the typology of uncertainty presented in Figure 6.5.

For Klir, uncertainty can be either *fuzziness* or *ambiguity* (two kinds of uncertainty). Ambiguity can itself be either *nonspecificity* or *discord*. These four terms can be related to some previously used terms in the other classifications: Fuzziness is close to vagueness, discord is a synonym of conflict, and nonspecificity means principally imprecision or generality.

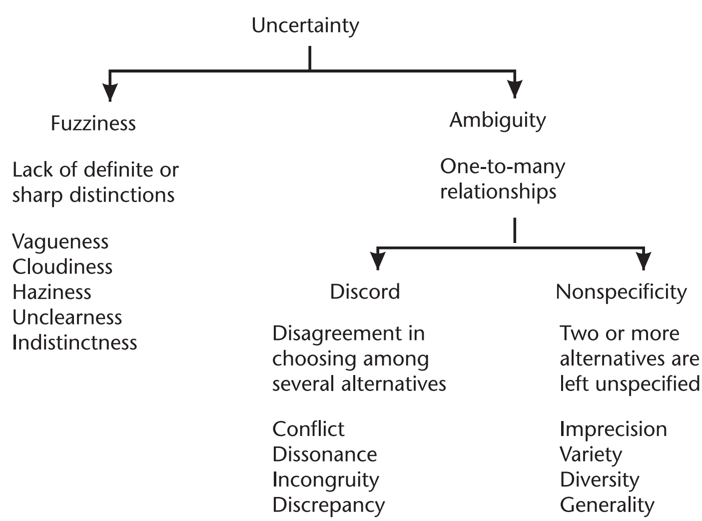


Figure 6.5 Types of uncertainty according to Klir and Yuan. (*After:* [15].)

In their typology, Klir and Yuan integrate the main key terms used by Smithson (fuzziness, nonspecificity, ambiguity), as well as the set theoretical aspect introduced by Krause and Clark (discord). In their discourse, Klir and Yuan do not mention knowledge and thus stay at a lower level of processing (i.e., at the information level). Indeed, they introduce the term *uncertainty-based information* to designate the information obtained from a reduction of uncertainty in a problem-solving situation.

6.3.6 Smets's Structured Thesaurus of Imperfect Information

Instead of a typology of uncertainty, Smets [16] has built a typology of *imperfection of information*. His model classifies imperfect information into three main categories (Figure 6.6):

- *Imprecision:* This is related to the content of the statement (informational property, external world, negligence). Several worlds satisfy the statement.
- *Inconsistency:* No world satisfies the statement.
- *Uncertainty:* This is induced by a lack of information, by some imprecision, ordering on the several worlds that satisfy the statement: *objective* (property of the information) and *subjective* (property of the observer).

Smets considers *imperfection* to be a general term, with uncertainty being a kind of imperfection. Uncertainty can be either *objective* (a property of the information, that is, sense II) or *subjective* (a property of the observer, that is, sense I). Smets's vision confirms what we have already mentioned, that "uncertainty induces uncertainty." The uncertainty (sense I) of an individual agent can also be a cause of imperfect information (sense II) circulating in a multiagent system. A way to clarify this would be to adopt Smets's vocabulary and talk about imperfect information instead of uncertainty as a physical feature. So only imperfect information (being of several kinds) induces uncertainty as a mental state.

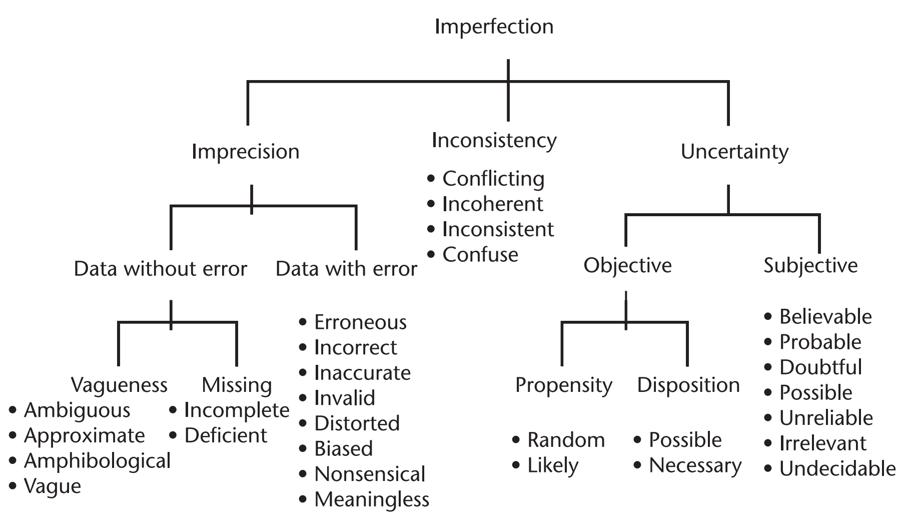


Figure 6.6 Adaptation of Smets's structured thesaurus on imperfection. (*After:* [16].)

6.4 Conclusion

We have decided to adopt Smithson's taxonomy as a basis for developing automated fusion systems that deal with uncertainty and knowledge. Knowledge involves belief, truth, and justification where these elements can be expressed as follows:

- The *truth* of a proposition can be asserted by asserting a propositional expression σ expressing the proposition.
- The stored *belief*, by individual x, of a proposition expressed by σ can be asserted by having Σ_x denote the set of stored beliefs of x and asserting $\sigma \in \Sigma_x$.
- The *justified* belief, by individual x of a proposition expressed by σ, can be asserted by having Σ_x denote the set of stored beliefs of x, having $\vdash$ denote the inference relation used by x and asserting $\Sigma_x \vdash \sigma$.

So,

$$x \text{ believes that } \sigma \text{ if } \Sigma_x \vdash \sigma$$

while

$$\sigma \text{ if } \sigma \text{ is true}$$

Smithson's taxonomy interprets ignorance as nonknowledge. This could rise to formal definitions as part of an ontology for high-level fusion.

References

[1] Barwise, J., and J. Perry, *Situations and Attitudes*, Cambridge, MA: MIT Press, a Bradford Book, 1983.

[2] Fodor, J. A., *Psychosemantics: The Problem of Meaning in the Philosophy of Mind*, Cambridge, MA: MIT Press, 1987.

[3] Block, N., *Advertisement for a Semantics of Psychology*, Studies in the Philosophy of Mind 10, Midwest Studies in Philosophy, Minneapolis, MN: University of Minnesota Press, 1986.

[4] Dennett, D. C., "True Believers," in D. C. Dennett, (ed.), *The Intentional Stance*, Cambridge, MA: MIT Press, 1987, pp. 13–35.

[5] Stich, S. P., *From Folk Psychology to Cognitive Science: The Case Against Belief*, Cambridge, MA: MIT Press, 1983.

[6] Brooks, R. A., "Intelligence without Representation," *Artificial Intelligence*, Vol. 47, 1991, pp. 139–160.

[7] Frege, G., "Sense and Reference," in I. M. Copi and J. A. Gould, (eds.), *Contemporary Philosophical Logic*, New York: St. Martin's Press, 1892, pp. 65–83.

[8] Lehrer, K., *Knowledge*, Oxford, U.K.: Clarendon Press, 1974.

[9] Chang, C. C., and H. J. Keisler, *Model Theory*, 2nd ed., Amsterdam: North-Holland Publishing Company, 1977.

[10] Lambert, D. A. "Grand Challenges of Information Fusion," *Proceedings of the ISIF Fusion 2003 Conference*, Cairns, Australia, July 7–10, 2003, pp. 213–220.

[11] Bronner, G., *L'incertitude*, Vol. 3187 of *Que sais-je?* Paris: Presses Universitaires de France, 1997.

[12] Smithson, M., *Ignorance and Uncertainty*, Berlin, Germany: Springer-Verlag, 1989.

[13] Krause, P., and D. Clark, *Representing Uncertain Knowledge: An Artificial Intelligence Approach*, Norwell, MA: Kluwer Academic Publishers, 1993.

[14] Bouchon-Meunier, B., and H. T. Nguyen, *Les incertitudes dans les systèmes intelligents*, Vol. 3110 of *Que sais-je?* Paris: Presses Universitaires de France, 1996.

[15] Klir, G. J., and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Upper Saddle River, NJ: Prentice Hall, 1995.

[16] Smets, P., "Imperfect Information: Imprecision—Uncertainty," in A. Motro and P. Smets, (eds.), *Uncertainty Management in Information Systems: From Needs to Solutions*, Norwell, MA: Kluwer Academic Publishers, 1997, pp. 225–254.

CHAPTER 7

Qualitative and Symbolic Approaches

Patrick Maupin, Anne-Laure Jousselme, and Éloi Bossé

7.1 Introduction

A formalization is necessary to be able to deal with knowledge or uncertainty: a formal framework in which knowledge, information, and uncertainty can be represented, combined, managed, reduced, increased, and updated. The objective is: (1) to build a model of a situation directly usable by the different theories of reasoning under uncertainty, and (2) to be able to deal with both knowledge and uncertainty. The potential theoretical frameworks available to model the situation-analysis process can be divided into two main categories: qualitative approaches (such as modal logic, nonmonotonic logic, truth-maintenance systems) and quantitative approaches (such as probability theory, evidence theory, fuzzy sets, random sets, possibility theory). Qualitative approaches seem better suited to reasoning on knowledge, while quantitative approaches are better candidates for uncertainty representation and management. Hence, a good solution for a global modelization of the situation could be a hybrid approach (quantitative logics, incidence calculus), mixing quantified evaluations of uncertainty and high reasoning capabilities.

In this chapter, some logical frameworks for uncertainty and knowledge processing are introduced. These frameworks, also called qualitative, logical, truth-functional, and intensional [1], are all extensions of classical logic.

7.2 Classical Logic: Propositional Logic

Since the beginnings of computer science, propositional logic together with first-order logic (also called predicate logic) has played an important role in the development of programming languages, software, and hardware, as well as architectures.

Propositional logic and first-order logic (together known under the designation classical logic) can be used when expert systems based on production rules cannot deal with the complexity of the problem to solve. Propositional logic and first-order logic require the formulation of a practical-problem in the form of a logical theorem. As far as reasoning (deduction) is concerned, the goal is to prove this theorem by the syntactic and the semantic tools available in these frameworks.

But reasoning is not necessarily the ultimate goal when dealing with these frameworks. For many tasks in high-level data fusion, aggregation of information is a very important task, and many tools are provided by classical logic. A language of propositions, together with a set of situations and a way of assigning a truth value to a proposition in a situation, is called a logic.

Definition (proposition): Several meanings have been given historically to the word "proposition." Before the modern area of formal logic (i.e., before the late nineteenth century), a proposition was a declarative sentence considered with its meaning.

In the modern area, a proposition expresses:

1. The meaning of a sentence;
2. The fully determinate circumstance or content capable of being asserted or expressed by a particular utterance of a sentence [2].

Propositional logic deals with propositional variables (or logic variables) that stand for arbitrary propositions. These variables stand for hypothetical propositions, and unless a particular proposition is substituted by a variable, the latter remain uninstantiated. The purpose of logic is to study the ways to achieve correct modes of inference. In this section, basic notions of entailment, inference, and grammatical concepts such as soundness, completeness, and decidability are also defined.

Definition (entailment): In a given logic, we say that a set of propositions Γ entails a proposition ϕ, written $\Gamma \vDash \phi$, if ϕ is true in every situation of Γ.

Definition (soundness): An inference relation $\vdash$ is called sound if, for any assertion set and any assertion set Γ, if $\Gamma \vdash \phi$ then $\Gamma \vDash \phi$.

Definition (completeness): An inference relation $\vdash$ is called complete if, for any assertion set and any assertion set Γ, if $\Gamma \vDash \phi$ then $\Gamma \vdash \phi$.

Definition (decidability): A set of formulas in a formal language is decidable if there is a decision procedure for membership in it. This decision procedure is called an algorithm, and this algorithm determines for any item whether it is a member of the set.

A set is semidecidable if there is an algorithm that confirms, when presented with a member of the set of formulas, that it is effectively a member of the set but that cannot give any answer when presented with a nonmember. Propositional logic is decidable since the truth-table method provides such a decision procedure, but first-order logic is only semidecidable.

Definition (derivation): If R is a set of inference rules, and Γ is a set of formulas, then a derivation of a formula ϕ from the premises Γ is a sequence of assertions $\psi_1, \ldots, \psi_n$, where ψ_n is the derived assertion ϕ, and for each ψ_i in the derivation, either ψ_i is a member of Γ, or there exists an inference rule in R whose conclusion is ψ_i, all of whose antecedents occur prior to ψ_i in the derivation.

Definition (generation by a set of rules): If there exists a derivation of ϕ from Γ under rule set R, then write $\Gamma \vdash_R \phi$. We call $\Gamma \vdash_R \phi$ the inference relation generated by the rule set R.

Definition (deduction/derivation) [3, 4]. Also designated as derivation. Typically, a deduction is a finite sequence of sentences of a logical system (see definition below) whose last sentence is a conclusion of the sequence in which the first sentence is an axiom and each subsequent sentence is either an axiom or follows from previous sentences through rules of transformation. For Sayward [4], "Deduction is a system-relative concept. It makes sense to say something is a deduction only relative to a particular system of axioms and rules of inference. The very same sequence of sentences might be a deduction relative to one such system but not relative to another." It should be noted that so-called systems of natural deduction are axiomless. It is said that proofs of theorems within a system are obtained more easily with natural deduction, whereas proofs of theorems about a system are obtained with more ease if this system has axioms.

Below, an axiomatization of propositional logic is presented briefly.

The set of primitive connectives is $\{\neg, \wedge\}$. The other operators are obtained by definition.

Definitions:

$$\phi \rightarrow \psi \triangleq \neg \phi \vee \psi$$

$$\phi \wedge \psi \triangleq \neg (\neg \phi \wedge \neg \psi)$$

$$\phi \leftrightarrow \psi \triangleq ((\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi))$$

Axioms:

R1: $(\phi \vee \psi) \rightarrow \phi$

R2: $\psi \rightarrow (\phi \vee \psi)$

R3: $(\phi \vee \psi) \rightarrow (\psi \vee \phi)$

R4: $(\psi \rightarrow \chi) \rightarrow ((\phi \vee \psi) \rightarrow (\phi \vee \chi))$

Rules:

Modus ponens (MP), and

Uniform substitution (US)

7.2.1 Dealing with Uncertainty

The task of preparing knowledge for reasoning necessitates a special treatment of exceptions. The problem is that in order to perform good reasoning, most of the time one cannot enumerate all the possible exceptions (because the list would be too long or simply because the agent in question is not aware of all these exceptions) or ignore the exceptions (because the system might fail at the task). The summarization of exceptions is, for Pearl [1], a compromise between enumerating and ignoring information and is therefore the key to reasoning under uncertainty.

As in propositional logic, it is possible to assign to each proposition a numerical measure of uncertainty. These measures are then combined, just as truth values are, using the same syntactic principles as propositional calculus. This is the rule-based approach adopted by first-generation expert systems.

Boolean algebra is of no use when it comes to studying how the exceptions to $\phi \rightarrow \psi$ interact with those of $\psi \rightarrow \chi$, and, furthermore, produce a new set of exceptions to $(\phi \wedge \psi) \rightarrow \chi$. Since these exceptions might be dependent in some unknown (or computationally intractable) way, as Pearl puts it, these exceptions are "robbing us of the modularity and monotonicity that make classical logic computationally attractive" [1].

The representation of knowledge in classical logic is not modular since any update of the knowledge base (i.e., adding an exception) requires a complete revision of the possible derivations (deductions). Some inconsistencies may arise, requiring the addition of new propositions to existing formulas in order to restore consistency.

Monotonicity is a very strong property of classical logic, but this property is after all denied very often by commonsense reasoning. For Horty [5], "Monotony states that if φ is a consequence of Γ then it is also a consequence of any set containing Γ (as a subset)."

In other words, if $\Gamma \vdash \varphi$, then $\Gamma \cup \{\phi\} \vdash \varphi$ obtains also. Horty continues:

> The import of monotony is that one cannot pre-empt conclusions by adding new premises. However, there are many inferences typical of everyday (as opposed to mathematical or formal) reasoning, that do not satisfy monotony. These are cases in which we reach our conclusions *defeasibly* (i.e., tentatively), reserving the right to retract them in the light of further information. Perhaps the clearest examples are derived from legal reasoning, in which defeasible assumptions abound. In the judicial system, the principle of *presumption of innocence* leads us to infer (defeasibly) from the fact that x is to stand trial, the conclusion that x is innocent; but clearly the conclusion can be retracted in the light of further information.

The impact of a new fact cannot be calculated "in stages" unless restrictive independence assumptions are made. Incrementality is also lost (i.e., we cannot account for the individual impact of items of evidence) unless restrictive independence assumptions are made again. For Pearl, "uncertainty forces us to compute the impact of the entire set of past observations to the entire set of sentences in one global step—this of course, is an impossible task" [1].

This difficulty in processing uncertainty with classical logic led to the development of numerous theoretical frameworks. Pearl [1] proposed a classification of methods processing uncertainty, opposing semantic and syntactic approaches (or equivalently, as Pearl puts it, intensional and extensional approaches). On the one hand, syntactic approaches, like production systems (also known as rule-based systems) and procedure-based systems, view uncertainty as a generalized truth-value. And just like in the purely syntactic processing of classical logic, these approaches "compute the uncertainty of any formula as a function of the uncertainties of its subformulas" [1].

The semantic approach models uncertainty using concepts such as possible worlds, situations, and state of affairs; in other words, this approach gives meaning to uncertainty characterization. For Pearl, "Extensional (syntactic) systems are computationally convenient but semantically sloppy, while intensional (semantic) systems are semantically clear but computationally clumsy" [1].

It seems that much of the difficulty encountered in the formalization of approaches dealing with computation and reasoning under uncertainty lie in the reconciliation of the syntactic and semantic aspects of the problem.

7.2.2 Calculus and Reasoning (Aggregation/Fusion)

Connectives, or logical operations, are used in proposition logic to link propositions together, to make new propositions appear. Combinations are obtained without any reference to the meaning of the propositions. A logical operation is a law of the composition of propositions themselves defined by a law of composition of their logical values.

Consider a connective acting on m propositions, where m is called the order of this connective. A function f is then a function acting on m variables, the latter being binary valued or Boolean. One has to determine $n = 2^m$ values of f. Since f can itself receive two values, the evaluation of the logical operation leads to $N = 2^{2^m}$ possible schemes.

7.2.3 Final Remarks

The following remarks are presented to summarize our comprehensive survey of proportional logic:

- Propositional logic is sound, complete, and decidable.
- Propositional logic, although not designed for uncertainty processing, can be extended in numerous ways, making it fit for applications requiring uncertain reasoning capabilities.
- Propositional logic is foundational to the domain of neural networks and thus connectionist artificial intelligence. Since high-level data fusion is concerned with parallel data processing and machine learning, it should be interesting to investigate the relationships between aggregation and machine-learning issues.
- It is possible to formalize propositional logic using a single connective. Is it possible to formalize high-level data-fusion aggregative functions using the same principle?
- The expressive power of propositional logic is not enough to be used as a foundation for arithmetic, and thus to high-level data fusion, requiring and extension making use of additional connective, the quantifiers. This extension will be presented in the next section on first-order logic.

7.3 Classical Logic: First-Order Logic

The pioneers of first-order logic are Boole, Frege, and Peirce, who formalized it not only for the study of deductive arguments but also as a tool for definition and conceptual analysis. First-order logic is an extension of propositional logic, which is considered a subset of first-order logic. Fist-order logic thus contains all axioms and theorems of propositional logic.

First-order logic, also called functional logic, the logic of predicates, or simply the logic of properties and relations, is a formal theory used to reason about the internal structure of a proposition.

1. *Objects* or *individual terms* are logical beings that can be linked in a proposition.
2. *Predicates* are logical beings that are used to link objects in a proposition. Predicates can act on a single object (unary) or on multiple objects (n-ary).
3. *General propositions* are an extension of the concept of proposition previously defined in propositional calculus, which can be described as a compound of one or many objects and a unary or n-ary predicate acting on it.

As will be seen, first-order logic can be used when objects of interest have attributes (or properties) and when it is of interest to represent and calculate relations between different objects. It is clear, thus, that first-order logic is fundamental to high-level data fusion. In fact, first-order logic is used as a foundation of most of mathematics.

Because of their expressive power, logical programming languages derived from first-order logic (such as LISP, which is, in fact, based on lambda-calculus, a higher-order logic, or PROLOG) are thus important tools in artificial intelligence, a domain highly related to high-level data fusion.

Properties are propositional functions acting on a single object, for example $f(x)$. This function could express any of the following: *x is dead, x is red, x runs,* and so forth. Relations are propositional functions acting on a multiple objects, for example $f(x, y, z)$. This function could express any of the following: *x, y, z is a set, x tells y to kill z, x is between y and z,* and so forth. There are two methods to construct an associated proposition using a functional proposition, namely, the *specialization of propositional functions* and the *quantification of propositional functions*.

Specialization consists of giving to the variables x, y, z related in $f(x, y, z)$ certain values x_i, y_j, z_k taken in the domain of definition of this last function. By this means, it is possible to write $f(x_i, y_j, z_k)$, meaning that x_i, y_j, z_k satisfy $f(x, y, z)$. It is common in first-order logic to call x_i, y_j, z_k free variables.

There are two distinct ways to quantify a propositional function, namely, the universal and existential quantifications.

1. Universal quantification
   All x, all y, and all z satisfy the relation $f(x, y, z)$, also noted

$$(\forall x)(\forall y)(\forall z) f(x, y, z) \tag{7.1}$$

2. Existential quantification
   There is at least a x, at least a y, and at least a z that satisfies the relation $f(x, y, z)$, also noted

$$(\exists x)(\exists y)(\exists z) f(x, y, z) \tag{7.2}$$

Since fist-order logic is only an extension of propositional logic, this fact is directly reflected in its axiomatic formalization.

The set of primitive connectives is $\{\neg, \wedge\}$. The other operators are obtained by definition.

Definitions:

$$\phi \rightarrow \psi \triangleq \neg \phi \vee \psi$$

$$\phi \wedge \psi \triangleq \neg (\neg \phi \wedge \neg \psi)$$

$$\phi \leftrightarrow \psi \triangleq ((\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi))$$

Axioms:

R1: $(\phi \vee \psi) \rightarrow \phi$

R2: $\psi \rightarrow (\phi \vee \psi)$

R3: $(\phi \vee \psi) \rightarrow (\psi \vee \phi)$

R4: $(\psi \rightarrow \chi) \rightarrow ((\phi \vee \psi) \rightarrow (\phi \vee \chi))$

Rules:

Modus ponens (MP)

Uniform substitution (US)

Universal elimination (UE)

Universal introduction (UI)

Since it is possible by *negation of universals and existential principles* to switch between the universal and existential quantifiers, it would be redundant to add rules for the other quantifier.

7.3.1 Dealing with Uncertainty

Historically, extensions to first-order logic for the processing of uncertain information are rare. Most of work has dealt with the extension of propositional logic. Léa Sombé [6] showed the limitations of this language to cope with simple problems of reasoning under uncertainty. For instance, following Léa Sombé, wanting to express a sentence like "Generally, students are young" in first-order logic would give the obvious "All students are young;" formally

$$(\forall x) Student(x) \rightarrow Young(x) \tag{7.3}$$

This formulation is sound if the only information available is "x is a student," but it is clearly insufficient to deal with exceptions. If one wants to deal with an exception, say, "Léa is not young," then the knowledge base must be rewritten as

$$(\forall x) Student(x) \wedge (x \neq Léa) \rightarrow Young(x) \tag{7.4}$$

and this process must be repeated for every other incoming exception. However, it is not realistic to express all possible exceptions. As will be shown in the section on nonmonotonic logics, one can used the concept of negation as failure to deal with such problems, but this is "stepping outside the classical logic framework."

Another drawback of first-order logic when processing uncertain information is that, as Léa Sombé puts it, "$\neq$ is not provable." The problem is that it is not possible to infer from "Paul is a student" that "Paul is a young." Indeed, "for every case which is not a priori exceptional, in order to deduce the desired conclusions, it is first necessary to prove that indeed it is not an exceptional case, and this is not necessarily feasible." Just like for propositional logic, the representation of information in first-order logic is not modular. It is not possible to express in Léa Sombé's simple example that "some young people are students," for in Léa Sombé's words the formula

$$(\exists x)\, Young(x) \rightarrow (\exists y)\, Young(y) \wedge Student(y) \tag{7.5}$$

is "a very poor formulation." The reason is the poor expressive power of the existential quantifier since it can hardly express anything else than "there exists at least one x. . . ." But for high-level data fusion not involving inference in the logical sense of the word, but rather involving basic arithmetic, geometrical transformations, aggregation operators, or relational algebra, first-order logic seems sufficient.

7.3.2 Calculus and Reasoning (Aggregation/Fusion)

In first-order logic, supplementary connectives are introduced in order to be able to act on the internal structure of propositions. High-level data fusion finds a lot of foundational concepts, such as membership relation, class of objects, and related ideas, found in the classical-sets-theory framework. Inference in first-order logic makes use of the rules defined for propositional logic but also makes use of special rules for the manipulation of quantified propositions.

7.3.3 Final Remarks

The following remarks are presented to summarize our comprehensive survey of first-order logic:

- Unlike propositional logic, first-order logic is only semidecidable. That is, there is no decision procedure for determining if an arbitrary formula is a theorem. By Alonzo Church's theorem, it is possible to show that any first-order language using a least one binary predicate symbol is doomed to see its validity rendered undecidable.
- First-order logic is not capable of expressing the notion of transitive closure.
- It is possible to extend first-order logic since this language is only about collections of objects or, equivalently, about the universe of discourse. A second-order logic's domain ranges over properties, sets, the relations of items in the domain of discourse, or functions from the domain to itself. A third-order logic would range over properties of properties, and so forth. Such logics are called higher-order logics.

7.4 Modal Logics and Knowledge Logics

Modal logic as been described as the logic of necessity and possibility, of *must be* and *may be*. Modal logic is used to model propositional attitudes and reasoning under uncertainty. This section on systems of modal logic will first give a definition of the basic system of modal logic, the system K, after which are built a wide variety of modal systems. System K itself contains propositional logic as a subset. Thus, every theorem and axiom of propositional logic is also valid in system K and in all systems build upon it.

The alphabet of K includes "$\neg$," "$\rightarrow$," and "$\Box$" for the modal operator "it is necessary that." Here the meaning "it is known that . . ." will be used instead of "it is necessary that" and the symbol "$\Box$" switched to K. When appropriate, the symbol B will also be introduced to formalize systems where statement like "it is believed that" can occur.

An important feature of modal logic, and this may explain its popularity in the communities of computer science, artificial intelligence and distributed computing, is the possibility of modeling situations and agent behavior just by selecting the appropriate set of axioms.

7.4.1 Dealing with Uncertainty

From a practical point of view, but also seen from the angle of the designer of humanlike machines, artificial agents based on traditional logical concepts are quite disappointing. These agents are disappointing because they "lack" the usual natural weaknesses of real agents. It is also perhaps because they lack these weaknesses that, for certain tasks, these artificial agents are not able to achieve human performances.

Limitations in agents arise for numerous reasons [7]:

1. *Agents lack awareness:* As far as knowledge is concerned, an agent cannot always give a value to a proposition, for example, if it is not even aware of the existence of the concept denoted by the proposition.
2. *Agents are resource bounded:* Agents have only limited memorization capabilities; in some cases, they have power-supply limitations (food, water, electricity), and so forth, or they have only limited cognitive and computational capabilities. Agents may have limited visual or auditory acuity. Sometimes, these limitations come from the outside and are situation driven: only a limited amount of time or money is available to do the job, and so forth.
3. *Agents don't always know the relevant rules:* In mathematics or logics, this is very frequent (as well as when it comes to cooking)!
4. *People don't focus on all issues simultaneously:* As Fagin and Halpern write [7]: "Even if a does perfect reasoning with respect to the limited number of issues on which he is focusing in any given frame of mind, he may not put his conclusions together. Indeed, although in each frame of mind agent a may be consistent, the conclusions a draws in different frames of mind may be inconsistent."

From a logical point of view, the set of beliefs of a real agent cannot be closed under logical consequence [8] "since it would mean that the agent has a decision procedure for first-order predicate logic.'' And there is no such decision procedure for first-order logic, according to Church's theorem.

The consequence of these limitations is that artificial agents, just like their human counterparts, may only have an incomplete, imprecise, perhaps inconsistent, set of beliefs about reality. They should thus be, like humans, in a ''mental'' state of uncertainty. Unfortunately, the conjoint use of possible-worlds semantics and Kripke models leads to perfect agents that are said to be logically omniscient.

7.4.2 Calculus and Reasoning (Aggregation/Fusion)

In propositional modal logic, two unary connectives (or operators) ($\Box$ and $\Diamond$) are added to propositional logic. The possibility modal connective ''$\Diamond$'' can be defined in terms of the necessity modal connective ''$\Box$:''

$$(\Diamond\phi) \leftrightarrow (\neg \Box \neg \phi) \tag{7.6}$$

It is interesting to note that the operators $\Box$ and $\Diamond$ behave similarly to the first-order logic quantifiers $\forall$ and $\exists$, as the definition of $\Diamond$ from $\Box$ mirrors the equivalence of $\forall(x)\phi$ with

$$\neg \exists(x) \neg \phi \tag{7.7}$$

The following relations also hold:

$$\Box(\phi \wedge \psi) \models (\Box\phi \wedge \Box\psi) \tag{7.8}$$

and

$$(\Box\phi \wedge \Box\psi) \models \Box(\phi \wedge \psi) \tag{7.9}$$

but while $(\Box\phi \vee \Box\psi) \models \Box(\phi \vee \psi)$,the converse is not true.

Similarly, in first-order logic, one obtains

$$\forall(x)(\phi \wedge \psi) \models \forall(x)\phi \wedge \forall(x)\psi \tag{7.10}$$

and

$$\forall(x)\phi \wedge \forall(x)\psi \models \forall(x)(\phi \wedge \psi) \tag{7.11}$$

while $\forall(x)\phi \vee \forall(x)\psi \models \forall(x)(\phi \vee \psi)$but the converse is not true. The same relations can also be exposed for the existential quantifier and the necessity operator.

In the different systems, it is also possible to study the implications occurring between the different operators.

7.4.3 Final Remarks

In a recent report, McCarthy [9], known for the introduction of circumscription in the nonmonotonic framework, questioned the ability of modal logics to model human commonsense reasoning and correctly formalize the use of modalities such as know, believe, want, and intend, or any combination of them. As the title of his reports suggests, McCarthy [9] is skeptical about modal logics but acknowledges the importance of modalities in formal systems.

Here are the principal questions asked by McCarthy as a challenge to proponents of modal logic:

1. *Many modalities:* What about the case of many modalities in the same sentence? Formalisms have been proposed, but what are their limitations? An example of such a complex sentence could be, "Agent *S* knows that agent *R* believes that it is possible that agent *T* intends to fight back."
2. *New modalities:* What about the ability of modal logic to incorporate modalities on an ad hoc basis just as its possible to do in predicate logic? New modalities may arise in the middle of a decision-support task, like new political constraints on the unfolding of a given mission. For McCarthy, "human-level AI requires that programs be able to introduce modalities when this is appropriate, e.g. have function taking modalities as values."
3. *Knowing what:* What about the fact that a theory of knowledge must be able to treat knowing *what* as well as knowing *that*? An example of knowing *what* could be "Agent *S* knows the range of the weapons on this frigate." For the time being, modal logic of knowledge is concerned with inferences based on knowing *that*. In situation analysis, it is, however, of primary importance to model and reason about one's own knowledge about the situation as well as what the other agents (friend or foe) know, or might know, about this same situation.
4. *Proving nonknowledge:* What about the possibility of using a variant of a Kripke-style accessibility relation in first-order logic, rather than as a mean to give modal logic a semantic interpreting the absence of knowledge? Autoepistemic logic, a nonmonotonic modal logic, can be use to model "all I know is . . ." which is not enough for McCarthy, since in many cases one would rather like to express the fact the "all I know about the value of *x* is. . . ." Instead, autocircumscription, which is a variant of circumscription, can be used to reason about agent *S*, ignoring the actual state of agent *R*, as well as about agent *S* inferring whether *R* ignores or not the actual state of *S*.
5. *Joint knowledge and learning:* What about the case where several agents' knowing something jointly implies not only that each of them knows but that they know it jointly? McCarthy proposed a way to formalize this situation by introducing pseudoagents into each subset of real agents, with these pseudoagents knowing what the subset knows.

7.5 Nonmonotonic Logics

The first-order logic framework lacks expressive power. Extensions of first-order logic were proposed, mainly by AI scientists, and one of these extensions promulgated the violation of a fundamental characteristic of classical logic reasoning, the monotonicity property. A logic is said to be monotonic if its consequence relation $\models$ has the following property:

$$\text{If } \Sigma \models \phi \text{ and } \Sigma' \models \phi \text{ then} \tag{7.12}$$

This property says that whenever a conclusion ϕ is derivable from a body of knowledge Σ, being a subset of a larger body of knowledge Σ', then ϕ will also be derivable from this larger body of knowledge. From the practical point of view, monotonicity implies that even if a body of knowledge grows, previously derivable conclusions will still be valid, which is not very often in agreement with everyday experience. As Niemela [10] puts it:

> Human commonsense reasoning and reasoning in artificial intelligence programs typically violates the monotonicity principle. People "jump" to conclusions which they may later retract when given more information. The conclusion that the local cafeteria is open at 2 p.m. on Friday is a typical example of conclusion that would be retracted when given more information, e.g. that the cafeteria is under repair.

A typical problem found in database theory involves nonmonotonic reasoning. Suppose a database contains information about available flights between two airports. Imagine now a query asking whether a flight connects these two cities. If the connection is found in the database, the reservation system will answer yes; otherwise, the answer will be no. This no answer is in fact obtained using the so-called closed world assumption [11], a reasoning principle stating informally that all positive facts are given in the database under consideration. The consequence is that all positive facts not in the database are assumed not to hold.

Similar patterns of reasoning found in the literature are the *negation as failure rule* [12], which is a common negation principle used in logic programming, and *inheritance by default* [13], which allows the inheritance of properties in hierarchies. When reasoning about action is involved, the *frame* [14] and *qualification problem* have to be solved. *Diagnosis*, as Reiter formalized it [15], is also a topic involving patterns of nonmonotonic reasoning. For Morgenstern nonmonotonic logics are the principal means to formalize plausible reasoning and "allow more general reasoning than standard logics, which deal with universal statements" [16].

What makes nonmonotonic logics extremely interesting for situation analysis is that the different formalisms proposed can implement reasoning under uncertainty and knowledge. For example, reasoning under uncertainty can be dealt with by default logic, a formalism allowing reasoning with incomplete information. On the other hand, as the name suggests, autoepistemic logic allows a form of nonmonotonic reasoning about belief.

Extending a discussion by Moses and Shoham [17] on defeasible forms of knowledge, to high-level data fusion in general and to situation analysis in particular, one obtains the relationships between nonmonotonic reasoning, knowledge and belief logics, and uncertainty formalisms. We believe that a formal triad including nonmonotonic reasoning should be taken as a basis for the formalization of situation analysis. Moses and Shoham [17] proposed a formal connection between knowledge and belief logics on one hand and nonmonotonic logics on the other.

In the same paper, Moses and Shoham recall that Halpern [18] proposed a logical setting allowing the connection between knowledge (and belief) logics and probability theory, and that Geffner [19] and Pearl [1] made the formal connection between probability theory and nonmonotonic reasoning.

A system of logic is called nonmonotonic when this system fails to meet the condition that for all statements $\Gamma = \{v_1, \ldots, v_n\}$, ϕ, ψ, if $\Gamma \vDash \phi$, then, for any ψ,

$$\Gamma, \psi \vDash \phi \tag{7.13}$$

A weak nonmonotonic logic is any logic with the following property: For some Γ, ϕ, and ψ,

$$\Gamma \vDash_{nm} \phi \tag{7.14}$$

where $\vDash_{nm}$ stands for the nonmonotonic consequence relation, but

$$\Gamma, \psi \nvDash_{nm} \phi \tag{7.15}$$

In a strong nonmonotonic logic, for some Γ, ϕ, ψ, where Γ and $\Gamma \wedge \phi$ are consistent, the following consequence relation:

$$\Gamma, \psi \vDash_{nm} \neg \phi \tag{7.16}$$

stating that the negation of ϕ follows from the database Γ (or set of theorems or axioms) updated with the piece of information ϕ.

Another way, perhaps more practical, of understanding the difference between a monotonic logic, like first-order logic, and a nonmonotonic logic, for example, default logic, is by looking at the way theorems are proved. In first-order logic, a derived theorem cannot contradict the axioms or the theorems used in the derivation process. A theorem $\vdash \phi_1$ derived from a set of axioms Γ (i.e., $\Gamma \vDash \phi_1$) will not contradict a theorem $\vdash \phi_2$ derived from $\Gamma \cup \vdash \phi_1$, thus preserving the consistency of the deductive system of first-order logic. Adding new theorems to a set of theorems thus preserves the integrity of the original theory.

In everyday life, however, in practical problem-solving situations, matters are more complex, and one has to deal with inconsistencies, personal or occurring in the outside world. G. A. Antonelli recalls that "A primary motivation (among AI researchers) for nonmonotonic logic or defeasible reasoning, which is so evident in commonsense reasoning, is to produce a machine representation for *default* reasoning or *defeasible* reasoning" [20].

7.5.1 Dealing with Uncertainty

When reasoning under uncertainty, for example, using the probability theory framework, the degree to which premises support a conclusion is inversely correlated to the length of the proof. Hence, probabilistic reasoning formalizations often fail to satisfy the cut property of first-order logic expressed previously in *consequence relations* [see (7.15) and (7.16)]. Here is an example from Antonelli [20]:

> Let Ax abbreviate "x is a Pennsylvania Dutch," Bx abbreviate "x is a native speaker of German," and Cx abbreviate "x was born in Germany." Further, let Γ comprise the statements "Most As are Bs," "Most Bs are Cs," and Ax. Then Γ supports Bx, and Γ together with Bx supports Cx, but Γ by itself does not support Cx. (Here statements of the form "Most As are Bs" are interpreted probabilistically, as saying that the conditional probability of B given A is, say, greater that 50%.)

If one considers that the cut property is a necessary feature of a well-behaved consequence relation, Antonelli suggests after examining his example of inductive reasoning, that it might not be possible to expect well-behaved relations of probabilistic consequence. Many formalisms of nonmonotonic logic have been proposed in order to deal with uncertainty. In the following section. the most common are briefly discussed.

Default logic, one of the leading and most flexible formalizations of nonmonotonic logics is a formal system for reasoning with defaults, developed by Reiter [21]. For Horty, "one can think of default reasoning, very roughly, as reasoning that relies on the absence of information as well as its presence, often mediated by rules of the general form: given P, conclude Q unless there is information to the contrary" [5].

In classical logic, a typical and simple rule of inference can be expressed by the use of a premise and conclusion pair of the form

$$\frac{\phi}{\psi} \tag{7.17}$$

expressing the idea that whenever ϕ occurs or is established, ψ follows. Default logic rather extends first-order logic using special rules having the form

$$\frac{\phi\colon \psi_1, \ldots, \psi_n}{\chi} \tag{7.18}$$

which reads "If ϕ (the prerequisite) is believed and it is consistent to assume $\psi_1, \ldots, \psi_n$ (the justifications), then χ (the conclusion) may be inferred." The usual interpretation given to consistency is related to the notion of the extendibility of first-order logic by such default rules. It should be noted that all ϕ, $\psi_1, \ldots, \psi_n$, and χ are closed formulas of first-order logic; that is, these formulas should not contain any free variables. Free variables are variables that are not bound to any quantifier in a given expression, as in

$$\forall x \phi(x, y) \rightarrow \psi(y) \tag{7.19}$$

where y is a free variable.

Formally, a default theory is a pair $\Delta = \langle W, D\rangle$, where W stands for a set of formulas, and D represents a set of default rules.

Niemela [10] proposed autoepistemic logic as a unified framework for nonmonotonic logics. For Niemela, autoepistemic logic is a modal logic with an operator L interpreted as "is believed." Autoepistemic logic was proposed by Moore [22, 23] in order to avoid defects encountered when using McDermott and Doyle's nonmonotonic logic [24, 25]. Autoepistemic logic models the introspection capability of a rational agent reasoning on its beliefs. Central to this process is, for Niemela, determining "the set of beliefs of the agent given a set of sentences as the initial assumptions or premises of the agent" [10].

An example of autoepistemic reasoning is $\neg L\phi \rightarrow \neg \phi$, an inference transmitting (similarly to the closed world assumption) the intuitive idea that if the agent does not believe ϕ, then $\neg \phi$ holds.

7.5.2 Calculus and Reasoning (Aggregation/Fusion)

This part is largely inspired by Antonelli [20]. Let $\vDash_{nm}$ represent any relation between sets of premises and single sentences. The following four properties are satisfied by the consequence relation $\vDash$ of first-order logic:

1. *Supraclassicality:* If $\Gamma \vDash \phi$, then $\Gamma \vDash_{nm} \phi$

   Supraclassicality requires that if φ follows from Γ in first-order logic, then it must also follow according to $\vDash_{nm}$. In other words, supraclassicality means that $\vDash_{nm}$ is an extension of the classical $\vDash$.
2. *Reflexivity:* If $\phi \in \Gamma$, then $\Gamma \vDash_{nm} \phi$.

   Reflexivity means that all sentences contained in Γ are inferable from Γ (i.e., if ϕ belongs to the set Γ, then ϕ is a consequence of Γ). Reflexivity is usually accepted as a rather minimal requirement for a relation of logical consequence.
3. *Cut:* If $\Gamma \vDash_{nm} \phi$ and $\Gamma, \phi \vDash_{nm} \psi$, then $\Gamma \vDash_{nm} \psi$.

   Cut is a conservation principle that states that if ϕ is a consequence of Γ, then ψ is a consequence of Γ, together with ψ only if it as been previously defined as a consequence of Γ alone. Cut can be seen as a condition of the length of a proof and states that the length of the proof does not affect the degree of the support of assumptions on the conclusion. If ϕ is already a consequence of Γ, and if ψ can be inferred from Γ in conjunction with ϕ, it follows then that ψ can also be obtained via a longer proof that proceeds indirectly by first inferring ϕ.

In nonmonotonic logics, the monotonicity property of the classical consequence relation is modified and replaced by either of the following rules:

- *Cautious monotony:* If $\Gamma \vDash_{nm} \phi$ and $\Gamma \vDash_{nm} \psi$, then $\Gamma, \phi \vDash_{nm} \psi$.
- *Rational monotony*: If it's not the case that $\Gamma \vDash_{nm} \neg \phi$, and moreover $\Gamma \vDash_{nm} \psi$, then $\Gamma, \phi \vDash_{nm} \psi$.

Antonelli [20] explains and comments that:

> Cautious Monotony is the converse of Cut: it states that adding a consequence ϕ back into the premise-set Γ does not lead to any decrease in inferential power. Cautious Monotony tells us that inference is a cumulative enterprise: we can keep drawing consequences that can in turn be used as additional premises, without

affecting the set of conclusion. Together with Cut, Cautious Monotony says that if ϕ is a consequence of Γ then for any proposition ψ, ψ is a consequence of Γ if and only if it is a consequence of Γ together with ϕ. . . . Rational Monotony can be regarded as a strengthening of Cautious Monotony, and like the latter, it is a special case of Monotony. However, there are reasons to think that Rational Monotony might not be a correct feature of a non-monotonic consequence relation.

7.5.3 Final Remarks

Antonelli identifies three major issues connected with the development of logical frameworks that can adequately represent defeasible reasoning [20]:

i. Material adequacy
Material adequacy concerns the question of how broad a range of examples is captured by the framework, and the extent to which the framework can do justice to our intuitions on the subject (at least the most entrenched ones).

ii. Formal properties
The question of formal properties has to do with the degree to which the framework allows for a relation of logical consequence that satisfies the above mentioned conditions of Supraclassicality, Reflexivity, Cut, and Cautious Monotony.

iii. Complexity
The third set of issues has to do with computational complexity of the most basic questions concerning the framework.

One should note that there are still few applications that can help to form a quick opinion on the usefulness of the framework for specific problem-solving situations. Nonmonotonic logic is a very important framework and an active field of research in AI. The growing literature on the subject is an indication that high-level data-fusion practitioners should study nonmonotonic logic paradigms and try to make connections with reasoning and belief or knowledge updating.

References

[1] Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, San Mateo, CA: Morgan Kaufmann Publishers, 1988.

[2] Bringsjord, S., and D. A. Ferrucci, "Logic and Artificial Intelligence: Divorced, Still Married, Separated . . . ?" *Minds and Machines*, Vol. 8, 1998, pp. 273–308.

[3] Detlefsen, M., D. C. McCarthy, and J. B. Bacon, *Logic from A to Z*, London, U.K.: Routledge, 1999.

[4] Sayward, C., "Deduction," in R. Audi, (ed.), *The Cambridge Dictionary of Philosophy*, Cambridge, U.K.: Cambridge University Press, 1999.

[5] Horty, J. F., "Nonmonotonic Logic," in L. Gobble, (ed.), *The Blackwell Guide to Philosophical Logic*, Blackwell Philosophy Guides, Oxford, U.K.: Blackwell Publishers, 2001, pp. 336–361.

[6] Sombé, L., "Reasoning under Incomplete Information in Artificial Intelligence: A Comparison of Formalisms Using a Single Example," *International Journal of Intelligent Systems*, Vol. 5, 1990, pp. 323–472.

[7] Fagin, R., and J. Y. Halpern, "Belief, Awareness and Limited Reasoning," *Artificial Intelligence*, Vol. 34, 1988, pp. 39–76.

[8] Moreno, A., *How to Avoid Knowing It All*, Computer Science Department, University Rovira i Virgili, Tarrogona, Spain, Research Report DEI-RR-97-013, 1997.

[9] McCarthy, J., *Modality, Si! Modal Logic, No!* Computer Science Department, Stanford University, Palo Alto, CA, 1997.

[10] Niemela, I., *Autoepistemic Logic as a Unified Basis for Nonmonotonic Reasoning*, Helsinki University of Technology, Department of Computer Science and Engineering, Digital Systems Laboratory, Espoo, Finland, Research Report A24, 1993.

[11] Reiter, R., "On Closed World Databases," in H. Gallaire and J. Minker, (eds.)," *Logic and Data Bases*, New York: Plenum Press, 1978, pp. 55–76.

[12] Clark, K. L., "Negation as Failure," in H. Gallaire and J. Minker, (eds.), *Logic and Data Bases*, New York: Plenum Press, 1978, pp. 293–322.

[13] Minsky, M., *A Framework for Representing Knowledge*, Cambridge, MA: MIT Press, 1974.

[14] McCarthy, J., and P. J. Hayes, "Some Philosophical Problems from the Standpoint of Artificial Intelligence," in B. Meltzer and D. Michie, (eds.), *Machine Intelligence*, Vol. 4, Edinburgh: Edinburgh University Press, 1969, pp. 463–502.

[15] Reiter, R., "A Theory of Diagnosis from First Principles," *Artificial Intelligence*, Vol. 32, 1987, pp. 57–95.

[16] Morgenstern, L., "Nonmonotonic Logics," in R. A. Wilson and F. Keil, (eds.)," *The MIT Encyclopedia of the Cognitive Sciences* (MITECS), Cambridge, MA: MIT Press, a Bradford Book, 1999.

[17] Moses, Y., and Y. Shoram, "Belief as Defeasible Knowledge," *Artificial Intelligence*, Vol. 64, 1993, pp. 299–322.

[18] Halpern, J. Y., "The Relationship between Knowledge, Belief, and Certainty," *Annals of Mathematics and Artificial Intelligence*, 1991, pp. 301–322.

[19] Geffner, H., *Reasoning with Defaults: Causal and Conditional Theories*, Cambridge, MA: MIT Press, 1991.

[20] Antonelli, G. A., "Non-Monotonic Logic," in E. N. Zalta, (ed.), *Stanford Encyclopedia of Philosophy*, Stanford, CA: The Metaphysics Research Lab, Center for the Study of Language and Information, Stanford University, 2001.

[21] Reiter, R., "A Logic for Default Reasoning," *Artificial Intelligence*, Vol. 13, 1980, pp. 81–132.

[22] Moore, R. C., "Autoepistemic Logic," in P. Smets, E. H. Mamdani, and D. Dubois, (eds.), *Non-Standard Logics for Automated Reasoning*, London, U.K.: Academic Press, 1989, pp. 105–136.

[23] Moore, R. C., "Semantical Considerations on Nonmonotonic Logic," *Artificial Intelligence*, Vol. 25, 1985, pp. 75–94.

[24] McDermott, D., and J. Doyle, "Non-Monotonic Logic I," *Artificial Intelligence*, Vol. 25, 1980, pp. 41–72.

[25] McDermott, D., "Non-Monotonic Logic II," *Journal of the ACM*, Vol. 29, 1982, pp. 33–57.

CHAPTER 8
Quantitative Approaches

Anne-Laure Jousselme, Patrick Maupin, and Éloi Bossé

8.1 Introduction

As mentioned in the previous chapter, the potential theoretical frameworks available to model the situation-analysis process can be divided into two main categories: qualitative approaches and quantitative approaches (e.g., probability theory, evidence theory, fuzzy sets, random sets, possibility theory). Quantitative approaches are better candidates for uncertainty representation and management. In this chapter, some numerical frameworks for uncertainty and knowledge processing are described in view of their use in the domain of information fusion for situation analysis.

8.2 Probability Theory

Probability theory is the oldest method for quantifying uncertainty. It is the branch of mathematics that develops models for "chance variations" or "random phenomena." The methods of probability assist us in understanding randomness and therefore provide us with tools for defining the measures of unpredictability or uncertainty. Different views of the theory of probabilities emerged over the twentieth century. The *objective* (or empirical) view opposes the *epistemological* view. On one hand, *classical, relative frequentist*, and *propensity* are the three major empirical interpretations for probabilities. On the other hand, *logical relationist* (John Maynard Keynes, Rudolf Carnap) and *intuitionist* (Frank Ramsey's degrees of belief) are both epistemologic interpretations. The axiomatic approach to probability was formulated by Kolmogorov in 1933.

Let Θ be the *sample space* of a random experiment. Θ is then the set of all outcomes for a given experiment (i.e., a collection of elements $\theta_1, \theta_2, \ldots$ called the *elementary events*). To each subset A of Θ is assigned a non-negative real number $P(A)$. This number $P(A)$ is called the *probability of the event A* and must satisfy the three following axioms:

Axiom 1: $$P(A) \geq 0 \tag{8.1}$$

Axiom 2: $$P(\Theta) = 1 \tag{8.2}$$

$$\text{Axiom 3:} \quad P\left(\bigcup_{n=1}^{\infty} A_n\right) = \sum_{n=1}^{\infty} P(A_n) \text{ if } A_n \cap A_m = \varnothing \text{ for } A_n \neq A_m \tag{8.3}$$

Axiom 3 is known as the condition of σ-additivity, or simply the axiom of additivity, and plays a crucial role in the theory of probability. For two events, it reduces to $P(A \cup B) = P(A) + P(B)$ if A and B have no elements in common. These axioms have been proposed by Kolmogorov. In its original contribution, five axioms were present, with the first and second axioms stating that subset A belongs to a set of subsets of Θ, σ_Θ being a field of sets containing Θ.

A *probability space* is a 3-tuple $(\Theta, \sigma_\Theta, P)$ where

- Θ is the sample space, the set of the elementary events.
- σ_Θ is a σ-algebra of Θ.
- P is a probability measure.

Let P be a probability measure (function) on Θ. The *probability distribution* is a mapping from Θ to [0, 1] such that

$$p(\theta) = P(\{\theta\}), \forall \theta \in \Theta \tag{8.4}$$

Following the additivity axiom (8.3), if Θ is a discrete, finite, nonempty set, the probability of A can be computed by the individual probabilities of each of its elementary events:

$$P(A) = \sum_{\theta \in A} p(\theta) \tag{8.5}$$

An *event* A is a subset of the sample space; in other words, an event is any collection of outcomes:

$$A = \{\theta \in \Theta \mid \theta \in A\} \tag{8.6}$$

The event that A does not occur is the *complement* of A, noted A^{c} or $\overline{A}$ or A':

$$\overline{A} = \Theta - A = \{\theta \in \Theta \mid \theta \notin A\} \tag{8.7}$$

The symbol – here denotes the substraction sets operator (the symbol "\" can also be used).

If A and B are two arbitrary events, then

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) \tag{8.8}$$

From axioms (8.2) and (8.3), it follows

$$P(A^{\mathrm{c}}) = 1 - P(A) \tag{8.9}$$

This is a very important property of probability theory. It means that once the event A occurs at a given time, its complement $\overline{A}$ can no longer occur. In other words, the proposition [θ is A] can be either true or false (probability theory is based on a Boolean logic).

$P(A \cap B)$ is called *joint probability* and can also be denoted by $P(A, B)$. In general, if we consider two distinct universes Θ_1 and Θ_2, with their associated probability measures $P_{\Theta 1}$ and $P_{\Theta 2}$, then the *joint probability measure* is defined by $\Theta = \Theta_1 \times \Theta_2$, the Cartesian product of universes Θ_1 and Θ_2, noted $P_{\Theta_1 \times \Theta_2}(\theta_1, \theta_2)$, where (θ_1, θ_2) is an element of the joint space Θ.

The probabilistic framework enables us to introduce the key notion of *conditional probability*. The conditional probability $P(A \mid B)$, is the probability of A being true given that B is. It is usually defined as

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} \tag{8.10}$$

on the condition that $P(B) > 0$. Because (8.10) is undefined when $P(B) = 0$, some prefer to take this definition as a primitive, basic concept. The quantity $P(A \mid B)$ is known as the *posterior probability* of A, that is, the probability of A evaluated not only in the light of background data but also given the further assumption that evidence or empirical prediction B obtains. $P(A \mid B)$ is often denoted by $P_B(A)$ to express the probability of A when the universe Θ is reduced to B. Hence, when B is fixed, P_B is an unconditional probability measure. In probability theory, no particular significance is given to the object $(A \mid B)$; this lack gave birth to the conditional event (CE) theory, described later in this book. Note that, given (8.10), we can compute the joint probability of two events A and B:

$$P(A, B) = P(A \mid B)P(B) \tag{8.11}$$

and for three events, A, B, and C, we have $P(A, B, C) = P(A \mid B, C)P(B \mid C)P(C)$.

Let $\wp_\Theta$ be a partition of Θ. Thus, the total probability for a given event A of Θ is

$$P(A) = \sum_{B \in \wp_\Theta} P(A \mid B)P(B) \tag{8.12}$$

In particular, $P(A) = P(A \mid B)P(B) + P(A \mid \overline{B})P(\overline{B})$.

Let Θ be a sample space. A *random variable* (r.v.) X is a function from Θ to some domain $\mathcal{D}$ (being possibly $\mathbb{R}$ or any subset of it), assigning a number $X(\theta)$ to every outcome θ of Θ. Whenever P is a probability measure, the function X must however satisfy the two following conditions:

1. The set $\{X \leq x\}$ is an event for every x.
2. $\{X = -\infty\}$ and $\{X = +\infty\}$ are two impossible events:

$$P(\{X = -\infty\}) = P(\{X = +\infty\}) = 0 \tag{8.13}$$

For a random variable, $P(X = x)$ expresses the probability of the event $\{X = x\}$ (i.e., the random variable $X(\theta)$ takes the particular value x of $\mathcal{D}$).

An event A of Θ can then be generated by a random variable X:

$$A = \{X \le x\} = \{\theta \in \Theta \mid X(\theta) \le x,\ x \in D\} \tag{8.14}$$

$\{X \le x\}$ is not a set of numbers but remains a set of experimental outcomes (a subset of Θ). In order to keep generality, an event will be denoted by A (some subset of the universe), defined or not defined from a random variable.

The *distribution function* of the random variable X is defined by

$$F_X(x) = P(\{X \le x\}) = P(X \le x) \tag{8.15}$$

and whenever no ambiguity exists, we write $F_X(x) = F(x)$. It follows that

$$P(x_1 \le X \le x_2) = F(x_2) - F(x_1) \tag{8.16}$$

The *density function* of the random variable X is then defined by

$$f(x) = \frac{dF(x)}{dx} \tag{8.17}$$

where F is the distribution function defined by (8.15). Consequently,

$$F(x) = \int_{x_x}^{x_2} f(x)\, dx \tag{8.18}$$

A random variable is said to be *discrete* (respectively *continuous*) if its distribution function is *discrete* (respectively *continuous*). If X is continuous, $P(X = x) = 0$, $\forall x \in \mathcal{D}$. If X is discrete, $P(X = x_i) = p_i = P(\theta_i)$. Then, (8.17) becomes $f(x) = \Sigma_i p_i \delta(x - x_i)$, where $\delta(.)$ is the impulse function.

Let X and Y be two random variables. Thus, the *marginal distributions* are

$$F_X(x) = F(x, +\infty) \text{ and } F_Y(y) = F(+\infty, y) \tag{8.19}$$

and the *marginal densities* are

$$f_X(x) = \int_{-\infty}^{+\infty} f(x, y)\, dy \text{ and } f_Y(y) = \int_{-\infty}^{+\infty} f(x, y)\, dx \tag{8.20}$$

For the discrete case, the *marginal probabilities* are

$$p_i = P(X = x_i) = \sum_j P(X = x_i,\ Y = y_j) \text{ and} \tag{8.21}$$
$$q_j = P(Y = y_j) = \sum_i P(X = x_i,\ Y = y_j)$$

8.2.1 Dealing with Uncertainty

8.2.1.1 Representation of Uncertainty

In the Bayesian theory of probability, knowledge, belief, and uncertainty are represented by probability measures in the form of $P(A)$. The probability of an event is a measure of the *likelihood* of its occurrence. $P(A)$ is often referred to as the *prior probability* of A, that is, the probability, evaluated only against the background data taken for granted, that A is true. $P(A) = P(\theta \in A)$ is the probability that θ belongs to A (i.e., that the event A occurs, in other words, that the proposition [θ is A] is true). In case of an empirical interpretation of probabilities, $P(A)$ represents the chances that the event A has to occur; in case of a subjective interpretation of probabilities, $P(A)$ represents our degree of *belief* (or degree of *certainty*) in the occurrence of the event A. Whatever the interpretation we ought to give to $P(A)$, the theory stays unchanged, and the axioms and inference rules stay valid. However, the results may be difficult to interpret and contrary to our intuition. Knowledge (or certainty) about the occurrence of an event is either modeled by $P(A) = 0$ (the impossible event) or $P(A) = 1$ (the certain event). An *evidence* is a certain event because it has been observed. However, the only way to model *total uncertainty* (or total ignorance) about the occurrence of a particular event is to uniformly distribute the amount of available probability, 1, among all the possible outcomes of the experiment, so $P(\theta_i) = 1/N$ if $N = |\Theta|$ is the number of possible outcomes. This is a weakness of the theory of probability.

8.2.1.2 Measures of Uncertainty

The only measure of probabilistic uncertainty is Shannon's entropy, defined by

$$H(p) = -\sum_{\theta \in \Theta} p(\theta) \log_2(p(\theta)) \tag{8.22}$$

where p is a discrete probability distribution (an equivalent formula exists for continuous measures). Klir and Folger [1] prove that (8.22) is also a measure of conflict.

If p_1 and p_2 are two probability densities, the Kullback-Leibler distance is defined by

Conflict (binary):

$$d_{KL}(p_1, p_2) = \int p_2(x) \log_2\left(\frac{p_2(x)}{p_1(x)}\right) dx \tag{8.23}$$

It quantifies a distance between two probability distributions and can be used as a measure of performance.

8.2.2 Calculus and Reasoning (Aggregation/Fusion)

Two any events A and B are *disjoint* (or *mutually exclusive*) if

Disjoint events: $$A \cap B = \varnothing \tag{8.24}$$

Note that disjointness is quite different from *independence*.

Physic independence: Two events A and B are independent if the occurrence of A cannot affect the occurrence of B.

The necessary and sufficient condition that A and B be stochastically independent events is

Stochastic independence (noninteractivity):

$$P(A \cap B) = P(A)P(B) \tag{8.25}$$

This is not always equivalent to physical independence.

A and B are probabilistically independent if, and only if,

Probabilistic independence:

$$P(A \mid B) = P(A) \text{ and } P(B \mid A) = P(B) \tag{8.26}$$

Stochastic independence and probabilistic independence are equivalent.

A and B are conditionally independent with respect to C if, and only if,

Conditional independence:

$$P(A, B \mid C) = P(A \mid C)P(B \mid C) \tag{8.27}$$

This means that, given the knowledge of C, A and B are independent. But the stochastic independence between A and B may not hold.

Two random variables X and Y defined on two distinct universes Θ_1 and Θ_2 with their probability measures P_{Θ_1} and P_{Θ_2} are *marginally independent* if

Marginal independence:

$$P_{\Theta_1 \times \Theta_2}(\theta_1, \theta_2) = P_{\Theta_1}(\theta_1)P_{\Theta_2}(\theta_2) \tag{8.28}$$

Classical inference: Suppose two sources of information provide two distinct probability measures for two distinct events defined on the same universe Θ. Then, classical inference in probability theory computes the joint probability $P(A, B)$ from $P(A)$ and $P(B)$. If A and B are independent, then $P(A, B) = P(A)P(B)$. Classical inferential models do not permit the introduction of prior knowledge into the calculations.

Bayes's theorem or Bayes's rule is the basic starting point for inference problems using probability theory. Given the definition of conditional probability (8.10) and noticing that $A \cap B$ is equal to $B \cap A$, it is easy to get

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{8.29}$$

where $P(B) = \Sigma_{C \in \wp_\Theta} P(B \mid C)P(C)$, $\wp_\Theta$ being a partition of Θ. Bayes's theorem is an "adjustment of subjective confidence" and "conditional probabilities" [2]. The rule of conditionalization says that when prediction B is verified, you should give a new prior probability to $P(A)$ (i.e., the unconditional probability) equal to its old posterior probability relative to B. Bayes's rule specifies how to do evidential reasoning (i.e., how to infer the cause B from the effect A) [2]. Bayesian inference can be implemented as a network, leading to Bayesian networks, a concept detailed in the next chapter, and can also be used to compute $P(A, B)$ from $P(A \mid B)$ and $P(B)$.

Consensus method: Suppose that each of n available sources of information provides a probability measure P_i, $i = 1, \ldots, n$. We affect each measure with a weight w_i representing the reliability of the source. Thus, the resulting probability for an event A is

$$P(A) = \sum_{i=1}^{n} w_i P_i(A) \tag{8.30}$$

where

$$\sum_{i=1}^{n} w_i = 1$$

This method is known as the *consensus method* [3] and belongs to the two main methods to aggregate experts opinions (the other being the Bayesian approach).

8.2.3 Final Remarks

To summarize our survey on probability theory, here are some remarks:

1. A probability measure clearly concerns probabilistic uncertainty, also called randomness.
2. Propositions are represented by events (subsets of a universe), which have only two possible truth-values: true or false. This limitation of Boolean logic led to the fuzzy-set theory and fuzzy logic.
3. The chance that this event has to be true (respectively false) is quantified by its probability (respectively the probability of its complement).
4. The axiom of additivity plays a crucial role since refuting the truth of an event imposes the acceptance of its complement. This limitation gave birth to the theory of evidence.
5. Independence between random variables can significantly reduce the complexity of the algorithms. This property is used to build Bayesian networks.
6. Prior probabilities are required, and their interpretation may be crucial. Moreover, a high number of these probabilities may be difficult to obtain.
7. Probability theory seems very well adapted to problems involving independent random variables, but such problems essentially arise at low levels of the fusion process.

8.3 Dempster-Shafer Theory

The theory of evidence, or the Dempster-Shafer theory, was originally developed by Dempster [4] in his work on upper and lower probabilities, then later written about by Shafer [5] in the famous book *A Mathematical Theory of Evidence*. Often interpreted as an extension of the Bayesian theory of probabilities, the theory of evidence offers two main advantages: (1) Uncertainty can be better represented in its framework because the measures are defined on the power set of the universe of discourse, instead of on the universe itself as in probability theory. This particularity leads to the relaxation of the additivity axiom of the probability theory and its replacement with a less restrictive one, a superadditivity axiom. (2) The combination of information is well defined through Dempster's rule of combination. This rule, when conditions of application are respected (independence of sources), leads to intuitive results and is a common way to fuse information coming from different sources. What makes the Dempster-Shafer theory widely used in the artificial intelligence area is probably its generalized aspect, many links having been established to it.

Let Θ be the *frame of discernment*, the set of all outcomes of an experiment, of all hypotheses. The *power set* of Θ, noted in general by $\mathbf{P}(\Theta)$ or 2^{Θ} in the discrete case, is the set of all the subsets of Θ. If $\Theta = \{\theta_1, \theta_2, \ldots, \theta_N\}$ then

$$2^{\Theta} = \{\varnothing, \theta_1, \ldots, \theta_N, (\theta_1, \theta_2), \ldots, \Theta\} \tag{8.31}$$

and thus contains 2^N elements.

A *belief function* is defined from 2^{Θ} to [0, 1], satisfying the following axioms:

Axiom 1: $$Bel(\varnothing) = 0 \tag{8.32}$$

Axiom 2: $$Bel(\Theta) = 1 \tag{8.33}$$

Axiom 3: For every positive integer n, and for every collection $A_1, \ldots, A_n$ of subsets of Θ,

$$\begin{aligned} Bel(A_1 \cup \ldots \cup A_n) \geq \sum_i Bel(A_i) - \sum_{i<j} Bel(A_i \cap A_j) + \ldots \\ + (-1)^{n+1} Bel(A_1 \cap \ldots \cap A_n) \end{aligned} \tag{8.34}$$

Contrary to the probability measure, the belief measure is nonadditive, and the axiom 3 [(8.3)] for probability theory is replaced by (8.34), the superadditivity axiom. For two subsets of Θ, we simply have

$$Bel(A \cup B) \geq Bel(A) + Bel(B) \ if \ A \cap B = \varnothing \tag{8.35}$$

A basic probability assignment m is a mapping defined from 2^{Θ} to [0, 1], which must satisfy the two following conditions:

$$m(\varnothing) = 0 \tag{8.36}$$

$$\sum_{A \subseteq \Theta} m(A) = 1 \tag{8.37}$$

$m(A)$ is the belief that a particular element θ of Θ belongs exactly to A (the exact belief committed to A). The belief function Bel can be deduced from m:

$$Bel(A) = \sum_{B \subseteq A} m(B) \tag{8.38}$$

A *focal element* is a subset A with a non-null mass, and the union of all focal elements is the *core* of the belief function. In practice, a small number of focal elements is necessary to describe a belief function.

A *plausibility function Pl* is defined from 2^{Θ} to [0, 1] as the dual function of the belief function:

$$Pl(A) = 1 - Bel(A^{c}) \tag{8.39}$$

However, considering (8.38), (8.39) can be rewritten, and *Pl* can be directly defined from m:

$$Pl(A) = \sum_{A \cap B \neq \emptyset} m(B) \tag{8.40}$$

Another function, called the *commonality function* noted by Q, is also sometimes used. Q is defined from 2^{Θ} to [0, 1], and the commonality numbers can be expressed through m:

$$Q(A) = \sum_{A \subseteq B} m(B) \tag{8.41}$$

The four measures m, Bel, Pl, and Q are one-to-one corresponding, each of them being possibly recovered from one of the others; they thus contain exactly the same information. (A recapitulation of these transformations can be found in Klir and Folger [1].)

A belief function Bel can be also represented by its corresponding *body of evidence*, noted by $(\mathcal{B}, m)$, where $\mathcal{B}$ is an element of 2^{Θ} containing all the focal elements of Bel, and m is the corresponding basic probability assignment. A body of evidence is thus a series of couples $(A, m(A))$:

$$(\mathcal{B}, m) = \{(A, m(A)) \mid m(A) > 0 \text{ and } A \subseteq \Theta\} \tag{8.42}$$

As Shafer stated, "belief functions readily lend themselves to the representation of ignorance." Indeed, complete ignorance is represented by the *vacuous belief function* defined by

$$m(A) = \begin{cases} 0 & \forall A \subset \Theta,\ A \neq \Theta \\ 1 & \mathit{if}\ A = \Theta \end{cases} \tag{8.43}$$

or simply $m(\Theta) = 1$.

Bayesian belief functions (or probability functions) are a subclass of belief functions.

A *simple support belief function* possesses only two focal elements, one being Θ:

$$m(A) = \begin{cases} s & \text{for } A \subseteq \Theta \\ 1 - s & \text{for } A = \Theta \end{cases} \tag{8.44}$$

where $0 < s < 1$.

A *dichotomous belief function* is a belief function with only three focal elements. being A, A^c and Θ. Many computation simplifications are based on the assumptions of simple support or dichotomous belief functions (see [6–8]).

8.3.1 Other Interpretations of the Dempster-Shafer Theory

Here are some important definitions.

Upper and lower probabilities: The first interpretation of belief functions is that introduced by Dempster [4]. The plausibility *Pl* and belief *Bel* measures are interpreted as two bounds (upper and lower, respectively) of the probability measure, defining thus a probability interval. This multivalued mapping interpretation is also shared by Shafer [5].

The transferable belief model (TBM): Smets and Kennes [9] introduced the TBM as a model for quantifying belief using belief functions. Its interpretation differs from Shafer's according to the following points [10]:

1. The TBM consists of a two-level model: a *credal level* where beliefs are entrained and a *pignistic level* where beliefs are used to make decisions. Bayesian theory considers that beliefs and decisions coexist; thus, they do not consider the credal level.
2. The TBM lies on two possible hypotheses: the *open-world* assumption or the *closed-world* assumption. In the first case, the empty set is allowed to have a non-null mass, whereas in the second case, its mass is restricted to zero. Under the open-world assumption, the normalization in Dempster's rule of combination can thus be avoided.
3. $m(A)$ is called the *basic belief mass* of A (and m, the basic belief assignment) and is interpreted as "a part of our belief that supports A and that, due to a lack of information, does not support any subproposition of A." *Bel* and *Pl* are then defined from m so that the mapping *Bel*: $2^\Theta \rightarrow [0, 1]$ is an (unnormalized) belief function if, and only if, there exists an m such that

$$\sum_{A \subseteq \Theta} m(A) = 1 \tag{8.45}$$

$$Bel(A) = \sum_{B \subseteq \Theta, B \neq \varnothing} m(A) \tag{8.46}$$

$$Bel(\varnothing) = 0 \tag{8.47}$$

The TBM is thus a model for representing beliefs not related to any probabilistic model; it aims at quantifying personal beliefs as *point values*, instead of *interval values*. Moreover, any connection with randomization has been eliminated. This model owes its name to the following: Suppose that at the credal level, our belief is represented by a mass function m. Suppose new evidence arises (such as "truth" is in B); then, the mass previously assigned to A is *transferred* to $A \cap B$ by Dempster's rule of conditioning.

Random sets: Another interpretation of belief functions has been proposed by Nguyen [11] as an equivalent model of random sets. This probabilistic interpretation has been fully developed by Goodman, Mahler, and Nguyen [12] and by Goodman and Kramer [13]. Because random sets appears to be good candidates for a unification framework, able to represent most of the quantitative approaches to uncertainty, this theory (and of course its close links to Dempster-Shafer theory) will be discussed later in this chapter.

8.3.2 Calculus and Reasoning (Aggregation/Fusion)

Even if independence is a crucial notion, it has not widely been studied in the theory of evidence. In his original work, Shafer [5] mentions some kinds of independence concept, such as cognitive and evidential independence. From our knowledge, the most relevant discussions and studies in this area are the works of Yaghlane, Smets, and Mellouli [14]. They investigate different ways to define independence relationships between variables in the framework of belief functions [14, 15]. Because clear examples are missing due to the quite complicated domain, we essentially give here some intuitive definitions of the concepts, referring to Yaghlane's, Smets's, and Mellouli's papers and Shafer's book for formal definitions. On the other hand, Bauer [16] and Voorbraak [17] have announced the concept of Dempster-Schafer (DS) independence, the required condition that sources to be combined through Dempster's rule of combination. This concept is needed in practice to guarantee that the results given by this rule are not counterintuitive.

Here are some important definitions.

Independent frames: Two compatible frames of discernment Θ_1 and Θ_2 are *independent* if no proposition discerned by one of them nontrivially implies a proposition discerned by the other [5].

Cognitive independence: Two variables are cognitively independent with respect to a belief function if new evidence that bears on only one of them does not change the degree of belief for propositions discerned by the other [5].

Evidential independence: Two variables are evidentially independent if their joint belief function is represented by the combination of their marginals using Dempster's rule of combination [5].

Noninteractivity: Two variables X and Y defined on Θ_1 and Θ_2, respectively, are noninteractive with respect to m defined on $\Theta_1 \times \Theta_2$ if the joint mass can be reconstructed from its marginals [14].

Irrelevance: In probability theory, the notion of irrelevance is equivalent to conditional independence—knowing Y has the value y does not affect belief in X [14].

Doxastic independence: This kind of independence has been introduced by Yaghlane Smets, and Mellouli [14] to specify the distinction between irrelevance and independence in the belief function framework (in Greek, *doxein* means "to believe"). The authors give the following interpretation: "Two variables are considered *doxastically independent* only when they are irrelevant and this irrelevance is preserved under Dempster's combination rule" [14].

DS independence: Voorbraak [17] proposes a requirement for belief functions to be combined with Dempster's rule—these functions must be DS independent. Condition to apply Dempster's rule of combination.

Suppose that two sources provide two pieces of information represented by two belief functions Bel_1 and Bel_2 or, equivalently, by their associated BPAs m_1 and m_2. Here are different ways to combine these two pieces of information.

Conjunctive rule of combination: Given two BPAs m_1 and m_2, the new BPA m resulting from the conjunctive rule of combination is defined by

$$(m_1 \wedge m_2)(A) = \sum_{B \cap C = A} m_1(B)m_2(C) \tag{8.48}$$

This kind of rule is usually used for two reliable sources: Source 1 *and* source 2 are both right. Its main inconvenience is the exponential increase in the number of focal elements in successive combinations. Indeed, if N_1 and N_2 are the numbers of focal elements of m_1 and m_2, respectively, then m will have a maximum of $N_1 N_2$ focal elements. Some work has been done to reduce or control the number of focal elements, using approximations of belief functions [16, 18–20] or using special belief functions (simple support or dichotomous belief functions), combined with special data structures [6–8].

Disjunctive rule of combination: Given two BPAs m_1 and m_2, the new BPA m resulting from the disjunctive rule of combination is defined by

$$(m_1 \vee m_2)(A) = \sum_{B \cup C = A} m_1(B)m_2(C) \tag{8.49}$$

A disjunctive rule is preferred whenever one source may be unreliable. Indeed, computing the union of sets (instead of the intersection) means that source 1 *or* source 2 may be right (we don't know which one). Of course, such a rule cannot be used alone in an algorithm since it will never converge to any singleton, putting more and more weight on larger and larger sets. The meanings of the conjunctive and disjunctive rules can be found in Smets [21].

Dempster's rule of combination: The rule proposed by Dempster for combining two belief functions is a normalized conjunctive rule, the normalization factor representing the conflict between the two sources. Let m_1 and m_2 be two basic probability assignments. The *conflict factor* $K(m_1, m_2)$ between m_1 and m_2 is defined by

$$K(m_1, m_2) = \sum_{B \cap C = \phi} m_1(B)m_2(C) \tag{8.50}$$

$K(m_1, m_2) = 0$ corresponds to the absence of conflict between m_1 and m_2, whereas $K(m_1, m_2) = 1$ implies a complete contradiction between m_1 and m_2.

Indeed, $K(m_1, m_2) = 0$ if, and only if, no empty set is created when m_1 and m_2 are combined. On the other hand, $K(m_1, m_2) = 1$ if, and only if, all the sets resulting from this combination are empty. To force the empty set to have a null mass, a normalization factor must be included in the conjunctive rule of combination. The normalized conjunctive rule can be written in this case by

$$(m_1 \oplus m_2)(A) = \frac{(m_1 \wedge m_2)(A)}{1 - K(m_1, m_2)} \tag{8.51}$$

where $K(m_1, m_2)$ is the conflict factor. This rule is then called *orthogonal sum* or *Dempster's rule of combination*. Although Dempster's rule of combination is the most classical (and probably the most used) rule for aggregation of belief functions in the evidential theory framework, some precautions must be taken. In particular, the belief functions to be combined must be DS independent; otherwise, the result could disagree with intuition (because a piece of evidence may be taken into account twice). Most of the criticism of Dempster's rule of combination is based on a misapplication of the rule (i.e., not respecting the DS independence of the sources). Moreover, being a conjunctive rule, it is subject to exponential increase in the number of focal elements, a problem mentioned above.

Conditioning: Suppose that we receive an evidence concerning the event B. The knowledge that the event B occurred can then be represented by $m(B) = 1$. From (8.51), and noticing that $(B \cap C = A) \equiv (B \cap C \cap A \neq \phi)$, we obtain *Dempster's rule of conditioning* (expressed with Pl):

$$Pl(A \mid B) = \frac{Pl(A \cap B)}{Pl(B)} \tag{8.52}$$

The same rule can also be expressed less easily with Bel:

$$Bel(A \mid B) = \frac{Bel(A \cup \overline{B}) - Bel(\overline{B})}{1 - Bel(\overline{B})} \tag{8.53}$$

This rule produces inferences that seem to be seriously wrong [22]. Smets criticizes the fact that (8.52) is often seen as a special case of (8.51), which leads to a "surrealistic" definition of conditional probability [11]. In the transferable belief model, Dempster's rule of conditioning is explained by the fact that when a new evidence implying that the truth is in B (subset of A) becomes available, the mass initially allocated to A is transferred to B. Another rule of conditioning known as the *geometric rule of conditioning* has been proposed by Suppes and Zanotti [23]:

$$Bel(A \mid B) = \frac{Bel(A \cap B)}{Bel(B)} \tag{8.54}$$

8.3.3 Dealing with Uncertainty

8.3.3.1 Representation of Uncertainty

Shafer interprets $Bel(A)$ as "one's degree of belief that the truth lies in A." So, $Bel(A)$ is the total belief committed to A, $Pl(A)$ is the total probability mass that can move into A, and $m(A)$ is the belief committed exactly to A (and to no other subset of A). The Dempster-Shafer theory is particularly suitable for representing ignorance (and uncertainty) as it has been built for this. Total uncertainty is thus represented by the vacuous belief function [i.e., $m(\Theta) = 1$]. This differs significantly from the representation capacity of probability theory constrained to the uniform distribution among the elements of Θ. This theory is adapted for representing and dealing with: (1) randomness (as it can be interpreted as a generalized Bayesian theory), (2) nonspecificity (because the measures are defined on subsets of Θ instead of singletons), and (3) conflict (for previous two reasons combined).

8.3.3.2 Measures of Uncertainty

In the Dempster-Shafer framework, we can define two kinds of probabilistic uncertainty measures:

Dissonance

$$E(m) = -\sum_{A \subseteq \Theta} m(A) \log_2(Pl(A)) \tag{8.55}$$

Confusion

$$C(m) = -\sum_{A \subseteq \Theta} m(A) \log_2(Bel(A)) \tag{8.56}$$

These both functions reduce to Shannon's entropy when *Bel* is a Bayesian belief function.

The first measure of nonspecificity is the Hartley measure defined for classical set theory by

$$U(A) = \log_2(|A|) \tag{8.57}$$

An extension of this measure is then

Nonspecifity

$$N(m) = \sum_{A \subseteq \Theta} m(A) \log_2(|A|) \tag{8.58}$$

Discord

$$D(m) = -\sum_{A \subseteq \Theta} m(A) \log_2\left(\sum_{B \subseteq \Theta} m(B) \frac{|A \cap B|}{|B|} \right) \tag{8.59}$$

Strife

$$S(m) = -\sum_{A \subseteq \Theta} m(A) \log_2\left(\sum_{B \subseteq \Theta} m(B) \frac{|A \cap B|}{|A|} \right) \tag{8.60}$$

A total uncertainty measure that takes into account both nonspecificity and strife can be defined as $NS(m) = N(m) + S(m)$, which means

$$NS(m) = -\sum_{A \subseteq \Theta} m(A) \log_2 \left(\frac{|A|^2}{\sum_{B \subseteq \Theta} m(B)\,|A \cap B|} \right) \tag{8.61}$$

Other measures of uncertainty, as well as a more profound discussion of general measures of uncertainty, can be found in [24, 25].

8.3.4 Final Remarks

Here are the final remarks for this section:

1. The Dempster-Shafer theory can be seen as an extension of the Bayesian theory of probability in the sense that it is built on the power set of the universe instead of being built on the universe itself.
2. The main implication is that the additivity axiom of the probabilities is eliminated and replaced by a superadditive one on belief functions. In particular, this allows the truth to be both in A and A^c (which is impossible in probability theory).
3. This theory deals with most of kinds of uncertainty (probability, nonspecificity, conflict), and ignorance can be well expressed by $m(\Theta) = 1$.
4. The fusion of two pieces of information is classically done by using Dempster's rule of combination. This rule must be manipulated carefully since (1) it produces an exponential increase of the number of focal elements (a problem for real-time applications) and (2) can lead to nonintuitive results whenever the belief functions to be combined are not independent. When two pieces of information present a high degree of conflict, other approaches [26] can be considered.
5. Probability judgments in natural languages cannot be modeled, in general, by belief functions. Rules like "if θ_1 is A, then θ_2 is B with a degree of belief 0.4" cannot be directly represented in the Dempster-Shafer theory.

8.4 Fuzzy-Set Theory

The concept of fuzzy sets was first introduced by Zadeh [27]. However, Black [28], in its paper on vagueness, was the first to develop the concept of fuzzy membership. A fuzzy set is a more general concept of the classical crisp set, where the membership of an element in this set is not described by a Boolean function but by a function defined on any other set. A fuzzy set is thus a set whose boundaries are not precise, not well defined (therefore, fuzzy). The theory of fuzzy sets is thus a generalization of the classical theory of sets. This theory has often been compared to probability theory [29, 30], but it appears that both theories are complementary, rather than in competition, since they address different types of uncertainty: vagueness (or fuzziness) for fuzzy sets and randomness for probability. Both theories differ essentially in the fact that one is based on a Boolean logic (probability) and the other is based on a multivalued logic (fuzzy sets). As fuzzy-set theory is a generalization of the classical sets concept, which is the basis of many theories, fuzzy-set theory is currently being extended to all areas with high formal content.

Let Θ be the universe of discourse. Fuzzy set A of Θ is defined by a membership function (or characteristic function):

$$\mu_A: \Theta \to L \tag{8.62}$$

$$\theta \propto \mu_A(\theta) \tag{8.63}$$

where L is an ordered set of membership values; generally, L is [0, 1]. $\mu_A(\theta)$ is the grade of membership of θ in A, the degree of compatibility of $x = \theta$ with $x \in A$, the degree of truth of the proposition [θ is A].

The characteristic function of a classical (crisp) set assigns only two values (0 or 1) to each element of Θ. Therefore, an element either belongs to the set or does not. Probability theory has to do with crisp sets, leading thus to a Boolean logic, whereas fuzzy-set theory leads to a multivalued logic. Whereas probability theory concerns the belonging of a random element θ of Θ to a fixed set A, in fuzzy-set theory, θ is a fixed element, and A is a not well-defined subset. In fuzzy-set theory, the question is not to determine the "most probable" fixed set A to which a random element θ can belong (as in probability theory); rather it is to determine the "most true" fuzzy set A to which the fixed element θ belongs.

A fuzzy set A is empty if, and only if,

Emptiness: $$\mu_A(\theta) = 0, \forall \theta \in \Theta \quad \mu_A(\theta)\,\mu_B(\theta) \equiv 0 \tag{8.64}$$

Two fuzzy sets A and B are equal iff

Equality: $$\mu_A(\theta) = \mu_B(\theta), \forall \theta \in \Theta \tag{8.65}$$

A fuzzy set A is normal if

Normality: $$\sup_{\theta \in \Theta} [\mu_A(\theta)] = 1 \tag{8.66}$$

Otherwise, it is called subnormal.

A fuzzy set A is contained in another fuzzy set B,

Inclusion: $$A \subseteq B \Leftrightarrow \mu_A \leq \mu_B \tag{8.67}$$

Let A be a fuzzy set defined on Θ, and let α be a real number of [0,1]. The α-cuts ${}^{\alpha}A$ of A are the crisp sets such that

α-cuts: $${}^{\alpha}A = \{\theta \mid \mu_A(\theta) \geq \alpha\} \tag{8.68}$$

The *level set* is the set of all levels for a fuzzy set A and is denoted by

$$\Lambda(A) = \{\alpha \mid \mu_A(\theta) = \alpha \text{ for some } \theta \in \Theta\} \tag{8.69}$$

We also define strong α-cuts ($^{+\alpha}A$) by replacing the symbol "≥" with ">" in (8.68). A very interesting property is that each fuzzy set can fully and uniquely be represented by its α-cuts [29].

The *scalar cardinality* for a fuzzy set A defined on a finite set Θ is

$$|A| = \sum_{\theta \in \Theta} \mu_A(\theta) \tag{8.70}$$

The *fuzzy cardinality* is defined by

Cardinality:

$$|\tilde{A}| = \sum_{\alpha \in \Lambda(A)} \frac{\alpha}{|{}^{\alpha}A|} \tag{8.71}$$

Let Θ and Ω be two universes. A (crisp) binary relation among Θ and Ω is a subset of $\Theta \times \Omega$:

Fuzzy binary relations:

$$R(\Theta, \Omega) = \{(\theta, \omega),\ \theta \in \Theta \text{ and } \omega \in \Omega\} \tag{8.72}$$

meaning that θ is associated (linked, related, connected) to ω in some manner. A *fuzzy binary relation* is a binary relation with degrees of strength associated to the relation between two elements θ and ω. Hence, for a (crisp) binary relation, these degrees of strength of the relation are either 0 or 1. A fuzzy binary relation is thus a fuzzy set defined on the Cartesian product of $\Theta \times \Omega$, where an element (θ, ω) may have varying degrees of membership within the relation [1]:

$$R(\Theta, \Omega) = \mu_{\Theta\Omega}(\theta, \omega) = \{[(\theta, \omega), x],\ \theta \in \Theta,\ \omega \in \Omega,\ x \in [0, 1]\} \tag{8.73}$$

The membership function $\mu_{\Theta\Omega}$ then defines the degrees of strength of the relation between two elements $\theta \in \Theta$ and $\omega \in \Omega$. As with every relation, a fuzzy relation can be symmetric, reflexive or transitive. If these three conditions are respected, then it is a fuzzy equivalence relation.

A *fuzzy number* is a fuzzy set defined on the set of numbers (for example $\Theta = \mathbb{R}$) and possessing at least the three following properties:

1. A is a normal fuzzy set.
2. $^{\alpha}A$ must be a closed interval for every $\alpha \in [0, 1]$.
3. The support of A must be bounded.

Fuzzy numbers are thus "numbers close to a given real number." A discussion of the application of fuzzy numbers is presented in Ma, Kandel, and Friedman [31].

A *linguistic variable* is a variable whose values are linguistic terms, being fully characterized by a 5-tuple (v, T, Θ, g, m):

- v is the name of the variable.

- T is the set of linguistic terms of v.
- Θ is the universal set.
- g is a syntactic rule for generating linguistic terms.
- m is a semantic rule assigning to each linguistic term in meaning $m(E)$ that is a fuzzy set on Θ.

8.4.1 Calculus and Reasoning (Aggregation/Fusion)

Here are some important definitions.

Disjointness: Two fuzzy sets A and B are *disjoint* if

$$\mu_A(\theta)\mu_B(\theta) \equiv 0 \tag{8.74}$$

Because fuzzy sets are before all sets, the notion of independence is not defined in this theory. The only equivalent concept thus concerns disjointness.

The aggregation of the information concerning a single object and coming from different sources (representing different fuzzy sets on the same support) into a single fuzzy set is performed by the aggregation operations on fuzzy sets. These operations are the generalization of the corresponding operations in classical set theory. Although fuzzy unions and intersections are not the only aggregation operations on fuzzy sets, they capture all the associative aggregation operations on fuzzy sets. For other (*idempotent*) aggregation operations, see [1]. There thus exist various fuzzy-set theories that differ in the operations they use. The *standard fuzzy-set theory* was developed by Zadeh [27], who defined the following *standard operators*.

Complement: The standard complement of the fuzzy set A ($\overline{A}$) is defined by

$$\mu_{\overline{A}}(\theta) = 1 - \mu_A(\theta) \tag{8.75}$$

$\mu_{\overline{A}}(\theta)$ is the degree to which θ *does not* belong to A.

Intersection: The standard intersection of the fuzzy sets A and B is defined by

$$(A \cap B) = \max[\mu_A(\theta), \mu_B(\theta)] \tag{8.76}$$

The max function can be replaced by any function T being a t-norm and define also an intersection between fuzzy sets. Some other frequently used fuzzy intersections are

Alegraic product: $$T[\mu_A(\theta), \mu_B(\theta)] = \mu_A(\theta), \mu_B(\theta) \tag{8.77}$$

Bounded difference:

$$T[\mu_A(\theta), \mu_B(\theta)] = \max[0, \mu_A(\theta) + \mu_B(\theta) - 1] \tag{8.78}$$

Drastic intersection:

$$T[\mu_A(\theta), \mu_B(\theta)] = \begin{cases} \mu_A(\theta) & \text{if } \mu_B(\theta) = 1 \\ \mu_B(\theta) & \text{if } \mu_A(\theta) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{8.79}$$

The standard union of the fuzzy sets A and B is defined by

$$(A \cup B) = \min[\mu_A(\theta), \mu_B(\theta)] \tag{8.80}$$

However, the min function can be replaced by any function S being a t-conorm and define also a union between fuzzy sets. Some other frequently used fuzzy unions are

Algebraic sum:

$$S[\mu_A(\theta), \mu_B(\theta)] = \mu_A(\theta) + \mu_B(\theta) - \mu_A(\theta)\mu_B(\theta) \tag{8.81}$$

Bounded difference:

$$S[\mu_A(\theta), \mu_B(\theta)] = \max[0, \mu_A(\theta) + \mu_B(\theta) - 1] \tag{8.82}$$

Drastic intersection:

$$S[\mu_A(\theta), \mu_B(\theta)] = \begin{cases} \mu_A(\theta) & \text{if } \mu_B(\theta) = 1 \\ \mu_B(\theta) & \text{if } \mu_A(\theta) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{8.83}$$

The duality of union and intersection (with respect to the complement) in the classical set theory is represented by the De Morgan Laws:

$$\overline{A \cap B} = \overline{A} \cup \overline{B} \tag{8.84}$$

$$\overline{A \cup B} = \overline{A} \cap \overline{B} \tag{8.85}$$

Even if all t-conorms and t-norms do not satisfy these laws, it is easy to show that the standard union and intersection, the algebraic sum and product, the bounded sum and difference, and the drastic union and intersection are dual t-conorms and t-norms, respectively, with respect to the standard complement defined in (8.75).

An *aggregation operation* on fuzzy sets performs the combination of several fuzzy sets to produce a single one:

$$h\colon [0, 1]^{\mathrm{n}} \to [0, 1] \tag{8.86}$$

$$(\mu_{A1}(\theta), \ldots, \mu_{AN}(\theta)) \to \mu_A(\theta) \tag{8.87}$$

Hence,

$$\mu_A(\theta) = h(\mu_{A1}(\theta), \ldots, \mu_{AN}(\theta)) \tag{8.88}$$

is the resulting aggregated fuzzy set of the aggregation operation of n fuzzy sets. h is a bounded, monotonic, increasing, continuous, symmetric, and idempotent function. For simplicity, let $\mu_{Ai}(\theta)$ by x_i denote the degrees of membership.

1. *Generalized means* is a class of averaging operations covering the interval of operations ranging from min to max operations:

$$h^{\beta}(x_1, \ldots, x_n) = \left(\frac{\sum_{i=1}^{n} x_i^{\beta}}{n} \right)^{1/\beta} \tag{8.89}$$

   where $\beta = \mathbb{R}^*$, and when $\beta < 0$, then $x_i \neq 0$. Special cases are the harmonic mean ($\beta = -1$) and the arithmetic mean ($\beta = 1$).
2. *Ordered weighted averaging (OWA) operations* are defined by first ordering the $x_i = \mu_{Ai}(\theta)$ in decreasing order for a fixed value θ. Then,

$$h_w(x_1, \ldots, x_n)(\theta) = \sum_{i=1}^{n} w_i y_i \tag{8.90}$$

   where

$$\sum_{i=1}^{n} w_i = 1$$

   and y_i is thus the ith largest $\mu_{Ai}(\theta)$.

A full list of aggregation operations can be found in [1, 32].

8.4.2 Dealing with Uncertainty

8.4.2.1 Representation of Uncertainty

In the fuzzy-set framework, knowledge is represented by membership functions. $\mu_A(\theta)$ represents the degree of truth of the proposition θ is A or, equivalently, the degree to which element θ belongs to the subset A. $\mu_A(\theta)$ is also the degree to which the constraint A is satisfied when θ is assigned to A [32]. $\mu_A(\theta)$ can also express our belief in this proposition. This clearly corresponds to vague concepts (i.e., concepts with ill-defined boundaries): θ is allowed to belong to more than one set with different degrees of membership. If $\mu_A(\theta) = 1$, then θ *certainly* belongs to A; if $\mu_A(\theta) = 0$, θ *certainly does not* belong to A. As we saw, fuzzy-set theory deals with vagueness (through degrees of membership) and with nonspecificity as it is an extension of the classical theory of sets. Total ignorance about the membership of θ is expressed by an equivalent distribution of the membership degrees among all possible sets to which θ can belong.

8.4.2.2 Measures of Uncertainty

The measure of nonspecificity for fuzzy sets is a generalization of the Hartley measure previously defined by (8.57):

Nonspecificity:
$$U(A) = \frac{1}{\mu_A^+} \int_0^{\mu_A^+} \log_2(|{}^{\alpha}A|)\, d\alpha \tag{8.91}$$

where A is a fuzzy set, μ_A^+ is the maximum value of its membership function, and $|{}^{\alpha}A|$ is the cardinality of the α-cut corresponding to α, $\alpha \in \{0, \mu_A^+\}$. This function is often called *U-uncertainty*.

A measure of fuzziness is defined for finite sets:

Fuzziness:
$$f(A) = \sum_{\theta \in \Theta} (1 - |2\mu_A(\theta) - 1|) \tag{8.92}$$

However, a generalization to infinite bounded sets is straightforward, replacing the sum with an integral over the considered infinite bounded set. $f(A)$ in (8.92) is thus the sum of all local distinctions of A and its complement, measured by the Hamming distance.

The conflict between two fuzzy sets A and B can be quantified by the degree of subsethood of A in B:

Conflict:
$$S(A, B) = \frac{|A \cap B|}{|B|} \tag{8.93}$$

where $\cap$ is the standard fuzzy intersection (8.76), and $|.|$ is the scalar cardinality (8.70).

Another possibility is to use the Hamming distance between A and B:

$$d(A, B) = \sum_{\theta \in \Theta} |\mu_A(\theta) - \mu_B(\theta)| \tag{8.94}$$

8.4.3 Final Remarks

Here are the final remarks for this section:

1. Fuzzy-set theory deals with vague (or fuzzy) information, especially that issued from human language.
2. Knowledge is represented by fuzzy propositions (fuzzy sets) (i.e., fuzzy memberships).
3. Contrary to events in probability theory, fuzzy propositions have more than two truth values, even an infinity: the main concern of fuzzy-set theory is thus the degree of truth of propositions such as [θ is A].
4. However, fuzzy membership functions are difficult to establish. One method is to equal them with probability densities but without forgetting their significance.

8.5 Possibility Theory

Zadeh [33] introduced the theory of possibility based on the theory of fuzzy sets that he presented in [27]. The main reason he advanced for this new theory is that imprecision is possibilistic rather than probabilistic in nature. Thus, when the meaning of information is the main purpose, analysis must be done within a possibilistic framework rather than a probabilistic one. The theory of possibility and its logical counterpart (possibilistic logic) have been fully studied by Dubois and Prade [28, 34–36]. Another possible formulation of possibility theory is that of a branch of the theory of evidence restricting focal elements to be nested. Possibility theory is not incompatible with probability theory but is rather a prolongation of the latter, in the case where the unique probability distribution hypothesis is not valid [35]. It is also more expressive than error intervals and less complex than a family of probability distributions since the information can be represented by a single possibility distribution. Possibility theory is thus a unifying framework for incomplete data and is a tool to aggregate information coming from multiple sources, such as expert opinions and measures from sensors or databases.

Possibility measure: Let Θ be a finite universe. A *possibility measure* Π is a mapping from the 2^{Θ} to [0, 1] such that

$$\Pi(\emptyset) = 0 \tag{8.95}$$

$$\Pi(\Theta) = 1 \tag{8.96}$$

$$\Pi(A \cup B) = \max[\Pi(A), \Pi(B)] \tag{8.97}$$

$\Pi(A)$ is the possibility that an element θ of Θ belongs to A.

Necessity measure: A necessity measure is the dual of Π:

$$N(A) = 1 - \Pi(\overline{A}) \tag{8.98}$$

Possibility and necessity measures also satisfy the following implications:

$$N(A) > 0 \Rightarrow \Pi(A) = 1 \tag{8.99}$$

$$\Pi(A) < 1 \Rightarrow N(A) = 0 \tag{8.100}$$

Possibility distribution: Possibility distributions generalize the membership functions of fuzzy sets. Let Θ be a finite universe of discourse. A possibility distribution π is a mapping from Θ to [0, 1] such that $\pi(\theta) = 1$ for some $\theta \in \Theta$. Every possibility measure is uniquely represented by the associated possibility distribution:

$$\Pi(A) = \max[\pi_x(\theta), \theta \in A] \tag{8.101}$$

We also have for the necessity measure

$$N(A) = \min[1 - \pi(\theta), \theta \notin A] \tag{8.102}$$

Note that whenever Θ is infinite, (8.101) and (8.102) must be replaced by

$$\Pi(A) = \sup_{\theta \in A}[\pi(\theta)] \tag{8.103}$$

and

$$N(A) = \inf_{\theta \in A}[1 - \pi(\theta)] \tag{8.104}$$

respectively. π is often noted π_x to specify the fuzzy variable of reference. x is a fuzzy variable of Θ, and θ is a particular value of Θ that x can take. $\pi(\theta) = \pi_x(\theta)$ is the possibility degree that $x = \theta$: $\pi_x(\theta) = 0$ means it is impossible for x to equal θ; $\pi_x(\theta) = 1$ means that nothing can prevent x from equaling θ.

Min/max rules: The possibility measures (respectively necessity) of the *union* (respectively the *intersection*) of two crisp sets A and B are given by

$$\Pi(A \cup B) = \max[\Pi(A), \Pi(B)] \tag{8.105}$$

and

$$N(A \cap B) = \min[N(A), N(B)] \tag{8.106}$$

if A and $B\Pi$ and N can be axiomatically defined from (8.105) and (8.106).

Joint and marginal possibility distributions: Let x and y be two (fuzzy) variables of Θ_x and Θ_y, respectively. As a joint probability distribution, a *joint possibility distribution* π_{xy} is defined on the Cartesian product $\Theta_x \times \Theta_y$:

$$\pi_{\Theta_x \Theta_y}(\theta_x, \theta_y) = \pi(\theta_x, \theta_y) \tag{8.107}$$

The *marginal distributions* are then projections π_x and π_y defined by

$$\pi_{\Theta_x}(\theta_x) = \max_{y \in \Theta_y} \pi(\theta_x, \theta_y) \tag{8.108}$$

$$\pi_{\Theta_y}(\theta_y) = \max_{x \in \Theta_x} \pi(\theta_x, \theta_y) \tag{8.109}$$

8.5.1 Other Formalizations of Possibility Theory

8.5.1.1 Possibility Theory Based on Fuzzy-Set Theory

Although the concept of possibility can be defined independently, Zadeh [33] defined a possibility distribution equal to a membership function of the corresponding fuzzy set:

$$\pi_A(\theta) = \mu_A(\theta) \tag{8.110}$$

Another notation eliminating ambiguity regarding the difference between both measures has been proposed by Dubois and Prade [28]:

$$\pi(\theta \mid A) = \mu(A \mid \theta) \tag{8.111}$$

$\pi(\theta \mid A)$ is the possibility that $\theta = x$, knowing $\theta \in A$ (A is a crisp set), whereas $\mu(A \mid \theta)$ is the degree of truth of $\theta \in A$ (A is a fuzzy set).

8.5.1.2 Possibility Theory as a Special Branch of Evidence Theory

When the focal elements of a belief function are nested ($A_1 \subseteq A_2 \subseteq \ldots \subseteq A_n = \Theta$), this belief function is called *consonant* (otherwise, it is *dissonant*) and defines a necessity function. In the same manner, a consonant plausibility function defines a possibility function. Thus, for a nested body of evidence ($\mathcal{B}_n$, m), the associated plausibility and belief functions correspond to possibility and necessity functions, respectively, and satisfy the properties (8.105) and (8.106), respectively. Moreover, every possibility measure Π on Θ is uniquely defined by a possibility distribution function π by (8.104) for all $A \subseteq \Theta$. So, if α_i is a fixed value of $\pi(\theta)$ (between 0 and 1), then let ${}^{\alpha}A_i$ denote its corresponding α-cut. Thus, it follows that

$$\pi(\theta) = \sum_{i=1,\, \theta \in {}^{\alpha}A_i}^{n} m({}^{\alpha}A_i) \tag{8.112}$$

if we set $m({}^{\alpha}A_i) = \alpha_i - \alpha_{I+1}$, $i = 1, \ldots, n$ and $\alpha_{n+1} = 0$. Then,

$$\pi(\theta) = (\mathcal{B}_n, m) = \{({}^{\alpha}A_i, m(A_i)) \mid i = 1, \ldots, n\} \tag{8.113}$$

is a nested body of evidence.

8.5.1.3 Likelihood

The possibility distribution can also be interpreted as a likelihood:

$$\Pi(\theta) = P(A \mid \theta) \tag{8.114}$$

is the probability that $x \in A$, knowing that $x = \theta$. For a better understanding of the difference between the concepts of probability, fuzzy sets, and possibility, see Figure 8.1.

8.5.2 Calculus and Reasoning (Aggregation/Fusion)

Noninteraction: Let $x = (x_1, x_2)$ be a binary fuzzy variable taking its values in $\Theta_1 \times \Theta_2$. Let π be a joint possibility distribution on the set $\Theta_1 \times \Theta_2$, and π_{x_1} and π_{x_2} be the corresponding marginals. The possibility measures associated with π_{x_1} and π_{x_2} are noninteractive if, and only if, [37],

$$\pi_x(\theta_1, \theta_2) = \min[\pi_{x_1}(\theta_1), \pi_{x_2}(\theta_2)] \tag{8.115}$$

where $\theta_1 \in \Theta_1$ and $\theta_2 \in \Theta_2$.

Independence: Two marginals' possibility measures are independent if, and only if, [37],

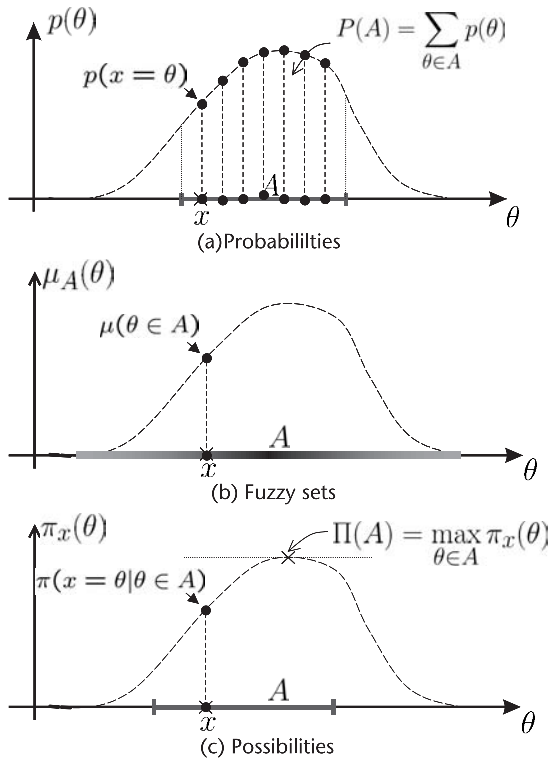


Figure 8.1 Differences between (a) probability, (b) fuzzy set, and (c) possibility concepts.

$$\pi_{x_1|x_2}(\theta_1 \mid \theta_2) = \pi_{x_1}(\theta_1) \tag{8.116}$$

$$\pi_{x_2|x_1}(\theta_2 \mid \theta_1) = \pi_{x_2}(\theta_2) \tag{8.117}$$

That is, the conditional possibilities are equal to the corresponding marginal possibilities. Contrary to their probabilistic counterparts, the concepts of noninteraction [(8.115)] and independence [(8.116) and (8.117)] are not equivalent [37]. In fact, possibilistic independence entails possibilistic noninteraction, but the reverse is false. Possibilistic independence is thus stronger than noninteraction.

Conjunctive rule of combination: Let $\pi_1(\theta)$ and $\pi_2(\theta)$ be two distinct possibility distribution functions given by two distinct sources. Thus, the resulting possibility distribution function obtained by a conjunctive rule is

$$\pi_\wedge(\theta) = T[\pi_1(\theta), \pi_2(\theta)] \tag{8.118}$$

where T is a t-norm.

Disjunctive rule of combination: Let $\pi_1(\theta)$ and $\pi_2(\theta)$ be two distinct possibility distribution functions given by two distinct sources. Thus, the resulting possibility distribution function obtained by a disjunctive rule is

$$\pi_\vee(\theta) = S[\pi_1(\theta), \pi_2(\theta)] \tag{8.119}$$

where S is a t-conorm, such as those defined for fuzzy sets.

Adaptative rule of combination: Let $\pi_1(\theta)$ and $\pi_2(\theta)$ be two distinct possibility distribution functions given by two distinct sources. Thus, the resulting possibility distribution function obtained by the adaptative rule proposed in [3] is

$$\pi_{AD}(\theta) = \max\left\{\frac{T[\pi_1(\theta),\ \pi_2(\theta)]}{1 - K(\pi_1,\ \pi_2)},\ \min[1 - K(\pi_1,\ \pi_2),\ S[\pi_1(\theta),\ \pi_2(\theta)]]\right\} \tag{8.120}$$

where $K(\pi_1,\ \pi_2)$ is a conflict factor equivalent to that defined for Dempster's rule of combination. This adaptative rule of combination allows one to take into account the reliability of the sources: If the two sources are reliable, conjunction is performed; otherwise, only one of them is reliable, and disjunction is used to avoid a big conflict. This is a way to wait for more reliable information without making the fusion algorithm converge toward an improbable solution.

8.5.3 Dealing with Uncertainty

8.5.3.1 Representation of Uncertainty

In possibility theory, uncertainty is represented by pairs of possibility and necessity measures, and imprecision is represented in terms of possibility distributions. $\pi(\theta) = 1$ for all $\theta \in \Theta$ corresponds to total ignorance (on the value of x). It is the less informing possibility distribution. Its counterpart is the maximal informing distribution, corresponding to $\pi(\theta) = 1$ for $x = \theta$ and 0 for all other $\theta \neq x$. $\Pi(A) = 1$ means A is fully possible, whereas $\Pi(A) = 0$ means A is fully impossible.

8.5.3.2 Measures of Uncertainty

Let π be a possibility distribution on Θ, and ${}^{\alpha}A_i$, $i = 1, \ldots, n$, with n α-cuts of π corresponding to n levels α_i such that $1 = \alpha_1 \leq \alpha_2 \leq \ldots \leq \alpha_n$. We let $m({}^{\alpha}A_i) = \alpha_i - \alpha_{i+1}$. Thus, different kinds of measures of uncertainty can be defined in the possibility theory framework.

Nonspecificity:
$$U(\pi) = \sum_{i=2}^{n} m({}^{\alpha}A_i)\log_2\left(\frac{|{}^{\alpha}A_i|}{|{}^{\alpha}A_{i-1}|}\right) \tag{8.121}$$

Conflict:
$$S(\pi) = \sum_{i=2}^{n} [m({}^{\alpha}A_i) - m({}^{\alpha}A_{i-1})]\log_2\left(\frac{|{}^{\alpha}A_i|}{\sum_{i=1}^{n} m({}^{\alpha}A_i)}\right) \tag{8.122}$$

Total uncertainty:

$$NS(\pi) = \sum_{i=2}^{n} [m({}^{\alpha}A_i) - m({}^{\alpha}A_{i-1})]\log_2\left(\frac{|{}^{\alpha}A_i|^2}{\sum_{i=1}^{n} m({}^{\alpha}A_i)}\right) \tag{8.123}$$

8.5.4 Final Remarks

Here are the final remarks for this section:

1. Possibility theory seems to be a nice framework for incomplete data processing.
2. Even if possibility theory is based on the concept of fuzzy sets, it significantly differs from fuzzy-set theory as it concerns uncertainty, partial ignorance, and the quality of being "not vague."
3. Possibility theory is way to generalize the error-interval notion: For example, an expert can give several error intervals around a particular value associated with confidence levels, the intervals being as small as possible.
4. In practice, it is used to model and combine expert opinions or in databases interrogation systems. Knowledge-based engineering systems are its main application area, measurements being taken from subjective evaluations. It has also been applied in pattern classification [38] or to multisource information-fusion for satellite image classification.
5. Other applications examples can be found in [38], such as the medical expert system DIABETO and the inference engine TAIGER.

8.6 Rough-Set Theory

In the early 1980s, Zdzislaw Pawlak introduced the rough-set theory as a "new mathematical tool to deal with vague concepts" and thus as an alternative to fuzzy-set theory [39]. The main purpose of rough-set theory is to replace uncertain and imprecise piece of information with two imprecise but certain pieces of information. Hence, a rough set is an approximate representation of a crisp set in terms of two other crisp sets, whereas a fuzzy set is defined by a membership function. For an introduction to rough-set theory, see [40–44]. Whereas Pawlak [39] pretends that rough sets are a general class of fuzzy sets, Dubois and Prade [45] think that fuzzy sets and rough sets concern different kinds of uncertainty: fuzziness for the former, and undiscernability (roughness) for the latter. These two aspects of vagueness often being present together, other kinds of sets have been proposed, namely rough fuzzy sets and fuzzy rough sets. The rough-set approach proposes a formal framework for the transformation of data into knowledge. Using concepts of certain and possible membership in a class, rules for classification may be created in various ways. An advantage of the rough-sets methodology over the Bayesian approach is that no assumptions about the independence of the attributes are necessary, nor is any background knowledge about the data.

Rough-set theory lies on the assumption that some knowledge about elements of the universe we are interested in is available, for instance, in the form of a database [46]. In the rough-set analysis, data from which information is retrieved is gathered in a table, being possibly of two kinds: an information system or a decision system.

An *information system* I_s, (or an approximation space) is represented by

$$I_s = (\Theta, \mathcal{A}, \{V_a\}, f_a) \tag{8.124}$$

where:

- Θ is a nonempty finite set of objects, called a *discourse*.
- $\mathcal{A}$ is the nonempty finite set of attributes representing the characteristics of each object.
- V_a is the domain of attribute a (the set of its values).
- f_a is an *information function* $f_a\colon \Theta \to V_a$ defined by each attribute $a \in \mathcal{A}$.

A subset A of Θ is called a *concept*. A *decision system* $\mathcal{D}_S$, is an information system for which the attributes of $\mathcal{A}$ can be classified into two disjoint sets: the *condition* attributes (dependent attributes) $\mathcal{C}$ and the *decision* attributes (independent attributes) $\mathcal{D}$. Thus, $\mathcal{A} = \mathcal{C} \cup \mathcal{D}$ and $\mathcal{C} \cap \mathcal{D} = \varnothing$. In most situations, $\mathcal{D}$ consists of only one element, $\mathcal{D} = d$. To convert an information system to a decision system, we must simply add a new attribute $d \in \mathcal{A}$.

Indiscernibility relation: An indiscernibility relation R_a is an *equivalence relation* (i.e., reflexive, transitive, and symmetric)[1] and is defined for all subsets a of $\mathcal{A}$ by

$$\theta_1 R_a \theta_2 \Leftrightarrow a(\theta_1) = a(\theta_2),\ \forall a \in a \subseteq \mathcal{A} \tag{8.125}$$

where $a(\theta)$ denotes the value of attribute a for object θ, $a(\theta) \in V_a$. Such a relation between two objects of Θ means that they cannot be distinguished (discerned) regarding all their considered attribute values and are thus identical. Therefore, an indiscernibility relation expresses the limitation of our knowledge about the elements of Θ. $\theta_1 R_a \theta_2$ means that "object θ_1 is indiscernible from object θ_2" ("θ_1 is too close [or too similar] to θ_2, so both elements are indiscernible"), with respect to the attributes in a [45]. For simplicity, we will drop the indice a of R_a when it is unnecessary. Of course, we must keep in mind that an indiscernibility relation is always defined for a subset of attributes a.

Equivalence class: Let R be an equivalence relation. An *equivalence class*, noted $R(\theta)$ or $[\theta]_R$, is the set of all the objects of Θ indiscernible from the object θ:

$$R(\theta) = \theta_i \in \Theta \mid \theta_i R \theta \tag{8.126}$$

$R(\theta)$ (or $[\theta]_R$) denotes the class of objects having the same description as θ in terms of attributes in $a \subseteq \mathcal{A}$. The equivalence classes are also called *elementary sets*. The family of equivalence classes defined by R is called the *quotient set* and is denoted by Θ/R. This family then forms a partition of Θ.

Lower and upper approximations: The two basic operations in rough-set theory are approximations of sets. For any subset (concept) A of Θ, a lower and upper approximation are defined with respect to R as follows:

$$\underline{R}A = \{[\theta]_R \mid [\theta]_R \subseteq A\} \tag{8.127}$$

1. R_a can also be a *tolerance relation* (i.e., reflexive and symmetric). If R_a is a tolerance relation, and $\theta_1 R_a \theta_2$, then θ_1 and θ_2 are called *similar* with respect to R_a, whereas they are referred as *indiscernible* if R_a is an equivalence relation.

$$\overline{R}A = \{[\theta]_R \mid [\theta]_R \cap A \neq \emptyset\} \tag{8.128}$$

Another possible notation is

$$R_*A = \{\theta \in \Theta \mid R(\theta) \subseteq A\} \tag{8.129}$$

$$R^*A = \{\theta \in \Theta \mid R(\theta) \cap A \neq \emptyset\} \tag{8.130}$$

Set $\underline{R}A$ consists of all the elements of Θ that can be *with certainty* classified to A (using the knowledge R), whereas set $\overline{R}A$ contains all the elements of Θ that can *possibly* be classified to A (using the knowledge R). This leads to the following definition [41]: *A set is rough with respect to R if its lower and upper approximations are different; otherwise, the set is exact (crisp).*

Boundary region: The boundary region of $A \subseteq \Theta$ is the difference between its upper and lower approximations:

$$BN_R(A) = \overline{R}A - \underline{R}A \tag{8.131}$$

$BN_R(A)$ contains all elements of A that cannot be classified either to A or to its complement $\overline{A}$ (using the knowledge R). If $BN_R(A) = \emptyset$, then A is a crisp set (with respect to R); otherwise, it is a rough set (with respect to R).

Rough membership functions: Several rough membership functions of object θ with respect to A can be defined. For example,

$$\mu_A^R(\theta) = \frac{|[\theta]_R \cap A|}{|[\theta]_R|} \tag{8.132}$$

or

$$\mu_A^R(\theta) = \begin{cases} 1 & \text{if } \theta \in \underline{R}A \\ 0.5 & \text{if } \theta \in [\overline{R}A - \underline{R}A] \\ 0 & \text{if } \theta \in \Theta - \overline{R}A \end{cases} \tag{8.133}$$

However, these membership functions cannot be extended to the union and intersection of sets like those defined for fuzzy sets [39].

Rough fuzzy sets and fuzzy rough sets: Because roughness and fuzziness are two aspects of vagueness (i.e., different types of uncertainty) that sometimes coexist, Pawlak defined two other kinds of sets, *rough fuzzy sets* and *fuzzy rough sets*, combining these two aspects. This idea was extended later by Dubois and Prade [45].

Fuzzy rough set: A fuzzy rough set is a rough set based on a fuzzy equivalence class. Let A be a crisp subset of a universe Θ, and $\tilde{R}$ be a fuzzy equivalence relation on Θ.[2] Thus, the fuzzy rough-set approximation of A is represented by each α-cuts of $\tilde{R}$:

2. We will note here $\tilde{R}$ to distinguish a fuzzy relation from a crisp one.

$$^{\alpha}\tilde{R}(A) = \left[{}^{\alpha}\underline{\tilde{R}}(A),\ {}^{\alpha}\overline{\tilde{R}}(A)\right] \tag{8.134}$$

Rough fuzzy set: A rough fuzzy set is a rough-set approximation of a fuzzy set. Let $\tilde{A}$ be a fuzzy set of Θ and R be a crisp equivalence relation on Θ.[3] Thus, the rough fuzzy set approximation of $\tilde{A}$ is represented by

$$R({}^{\alpha}\tilde{A}) = \left[\underline{\tilde{R}}({}^{\alpha}\tilde{A}),\ \overline{\tilde{R}}({}^{\alpha}\tilde{A})\right] \tag{8.135}$$

where

$$\underline{\tilde{R}}(A) = \mu_{\underline{\tilde{R}}(A)}(\theta) = \inf\{\mu(\theta) \mid [\theta]_R\} \tag{8.136}$$

$$\overline{\tilde{R}}(A) = \mu_{\overline{\tilde{R}}(A)}(\theta) = \sup\{\mu(\theta) \mid [\theta]_R\} \tag{8.137}$$

8.6.1 Calculus and Reasoning (Aggregation/Fusion)

Let $\mathcal{F} = A_1, \ldots, A_n$ be a family of concepts of Θ. A_i is said to be *dispensable* for $\mathcal{F}$ if

$$\bigcap_{j=1, j\neq i}^{n} A_j = \bigcap_{j=1}^{n} A_j \tag{8.138}$$

Otherwise, the concept A_i is *indispensable*. A family of concepts $\mathcal{F}$ is *independent* if all of its concepts A_i are indispensable for $\mathcal{F}$. Because rough-set theory replaces a vague concept (vague set) with two crisp sets, then the classical theory of sets can be applied for the aggregation or fusion.

Combination of concepts: Let Θ be the universe and let A and B be two subsets (concepts) of Θ. Given an indiscernibility relation R, the approximations of A and B are, respectively, the two intervals of sets $[\overline{R}A; \underline{R}A]$ and $[\overline{R}B; \underline{R}B]$. The union $A \cup B$ is consequently approximated by the two following bounds:

$$\underline{R}(A \cup B) \supseteq \underline{R}A \cup \underline{R}B \tag{8.139}$$

$$\overline{R}(A \cup B) = \overline{R}A \cup \overline{R}B \tag{8.140}$$

In the same way, the intersection $A \cap B$ is bounded by the two following sets:

$$\underline{R}(A \cap B) = \underline{R}A \cap \underline{R}B \tag{8.141}$$

$$\overline{R}(A \cap B) \subseteq \overline{R}A \cap \overline{R}B \tag{8.142}$$

Combination of knowledge: Let $\mathcal{R} = R_1, R_2, \ldots, R_m$ be a family of knowledge on Θ. An equivalent knowledge can be computed by taking the intersection of all the knowledge

3. $\tilde{A}$ will denote here a fuzzy set to distinguish it from a crisp one.

$$\mathrm{IND}(\mathcal{R}) = \bigcap_{i=1}^{m} R_i \tag{8.143}$$

$\mathrm{IND}(\mathcal{R})$ thus represents the set of equivalence classes of indiscernible objects.

8.6.2 Links to Other Theories

8.6.2.1 Fuzzy Sets

Pawlak [39] compares fuzzy sets and rough sets. He concludes that the rough set is a more general concept than the fuzzy set. If equality exists between the pair (8.139) and (8.142), then rough sets reduce to fuzzy sets. Moreover, he shows that the union and intersection of fuzzy sets (8.80) and (8.76) defined by Zadeh have no equivalent counterpart in rough-set theory. However, this vision is not shared by Dubois and Prade, who think that the two concepts address different kinds of uncertainty, fuzziness and roughness (indiscernibility). Their interpretation thus leads to the concepts of rough fuzzy sets and fuzzy rough sets.

8.6.2.2 Dempster-Shafer Theory

Skowron [47] establishes a bridge from rough sets to evidence theory. He shows that each problem represented in rough-set theory can be also represented in evidence theory. Let Θ be a universe of discourse, A a concept of Θ, and R an equivalence relation (knowledge) on Θ. Thus, belief and plausibility functions can be defined from the lower and upper approximations of A (with respect to R):

$$\underline{k}A = Bel(A) = \frac{|\underline{R}A|}{|\Theta|} \tag{8.144}$$

$$\overline{k}A = Pl(A) = \frac{|\overline{R}A|}{|\Theta|} \tag{8.145}$$

In the rough-set framework, $\underline{k}A$ and $\overline{k}A$ are called the *lower quality function* and *upper quality function*, respectively.

8.6.3 Dealing with Uncertainty

8.6.3.1 Representation of Uncertainty

In rough-set theory, knowledge is regarded as the ability to classify objects [48]. Pawlak [42] says that rough-set theory is a mathematical approach to imprecision, vagueness, and uncertainty. However, according the description of the concept of uncertainty in Chapter 6, we would say that rough-set theory mostly deals with indiscernibility (roughness, coarseness) (whereas fuzzy sets mostly deals with vagueness, probability theory with randomness, and so forth). Knowledge is represented by indiscernibility relations. If a knowledge R provides any information about the concept A, then total ignorance is represented by the lower and upper approximation sets, being $\varnothing$ and Θ, respectively.

8.6.3.2 Measures of Uncertainty

From our knowledge, in the rough-set framework, no measure of uncertainty has been defined, except those related to classical sets (the Hartley measure for nonspecificity) or to fuzzy sets (measure of fuzziness). It could be interesting, however, to define a measure of roughness.

8.6.4 Final Remarks

These remarks conclude this section on rough-set theory:

1. This theory has a strong qualitative analysis ability, but because of the additional quantifications, such as membership functions and its close links with quantitative approaches, it appears in this book under quantitative approaches.
2. Rough-set theory deals with roughness (or indiscernibility), a special aspect of vagueness, by approximating sets by setting upper and lower bounds.
3. However, it can deal only with discrete values as it uses the granularity structure of the given data.
4. Knowledge is then regarded as the ability to classify objects (depending on the granularity).
5. It is a method that avoids external parameters, using only internal knowledge, relying on no prior model assumptions (e.g., fuzzy sets or probabilities).
6. No assumptions about the independence of the attributes are necessary, nor is any background knowledge about the data.
7. A wide range of applications uses the ideas of the theory: Medical data analysis, aircraft-pilot performance evaluation, image processing, and voice recognition are a few examples. Rough-set theory is essentially used for data analysis. This theory seems to be specially appropriate for data reduction; discovering dependencies, similarities, differences, and patterns in data; and extraction of hierarchy rules [48], data mining, and approximate classification. This approach has been successfully implemented in real-life applications, such as engineering design, the analysis of hierarchy factors, approximate classification of patients, and so on. A detailed list of implementation examples can be found in Pawlak [41]. Moreover, the rough-set theory is used as the theoretical basis for problems in machine learning and has inspired a variety of logical research.

8.7 Conditional Event Theory

The study of conditional events (or conditional objects) was initiated in the thirties by De Finetti and others. Their idea was to give a mathematical and logical meaning to conditional relationships between two events (or logical formulae) in agreement with the theory of probability and, more specifically, conditional probability. More recently, CE theory has been revived by some authors [12, 35, 49]. The idea remains to define a mathematical object, called *a conditional event*, such that the set of these objects can use a Boolean algebra, in which conditional probabilities become "true" probability measures. The interest in developing conditional events algebras comes mainly from the fact that rules in natural language ("if B, then A") used in knowledge-based or expert systems are naturally modeled by conditional events ("A given B"). However, rules are generally modeled by the Boolean material implication, which is incompatible with conditional probabilities. Thus, the aim is to find a suitable mathematical framework in which (1) rules can be consistently modeled as conditional events, and (2) standard probability theory can be extended to these rules. A complete description of conditional event algebra and its potential use in information fusion is available by Goodman and Kramer [13] and by Goodman, Nguyen, and Walker [50].

8.7.1 Links to Other Theories

8.7.1.1 Probability Theory

The link between conditional event theory and probability has been already described since CE theory is developed to provide a meaning for the conditional object arbitrarily defined in probability theory. CE theory is an extension of probability theory around the concept of the conditional event. CE theory must thus reduce to standard probability when conditional events reduce to ordinary events.

8.7.1.2 Fuzzy-Set Theory

Goodman, Nguyen, and Walker [50] showed that conditional event $(A \mid B)$ can be uniquely represented as a three-valued membership function defined on Θ by

$$\mu_{(A \mid B)}(\theta) = \begin{cases} 1 & \text{if } \theta \in A \cap B \\ 0.5 & \text{if } \theta \in \overline{B} \\ 0 & \text{if } \theta \in \overline{A} \cap B \end{cases} \tag{8.146}$$

8.7.2 Calculus and Reasoning (Aggregation/Fusion)

Because CE algebra is an extension of probability theory, independence should be defined in the same way as in probability theory. The combination of information (rules) is done through the connectors "$\wedge$," "$\vee$," and "$\neg$." It is thus translated from one algebra to another. Here, we present three different CE algebras, defined by their three connectors.

Product space conditional event algebra (PSCEA): Let $(\Theta, \sigma_\Theta, P)$ be a probability space (of unconditional, i.e., ordinary, events). Thus, the PSCEA is the extension of $(\Theta, \sigma_\Theta, P)$ denoted by $(\hat{\Theta}, \hat{\sigma}_\Theta, \hat{P})$ where:

- $\hat{\Theta}$ is the infinite Cartesian product of the Θ,

$$\hat{\Theta} = \Theta \times \Theta \times \ldots \tag{8.147}$$

- $\hat{\sigma}_\Theta$ is the σ-algebra spanned by

$$\sigma_\Theta \times \sigma_\Theta \times \sigma_\Theta \times \ldots \tag{8.148}$$

- A conditional event of $\hat{\sigma}_\Theta$ can be written by

$$(A \mid B) = \bigcup_{n=1}^{\infty} \left[(\overline{B})^n \times A \cap B \right] \tag{8.149}$$

or with a recursive form

$$(A \mid B) = (A \cap B \mid \Theta) \cup (\overline{B})^n \times A \cap B \tag{8.150}$$

- $\hat{P}$ is the product measure on $(\hat{\Theta}, \hat{\sigma}_\Theta)$. It can be shown that within this algebra,

$$\hat{P}[(A \mid B)] = P(A \mid B) \tag{8.151}$$

SAC algebra: This algebra has been proposed by Schay [51]:

$$\neg_1 (A \mid B) = (\neg A \mid B) \tag{8.152}$$

$$(A \mid B) \wedge_1 (C \mid D) = (\neg B \vee A)(\neg D \vee C) \mid (B \vee D) \tag{8.153}$$

$$(A \mid B) \vee_1 (C \mid D) = (AB \vee CD) \mid (B \vee D) \tag{8.154}$$

GNW algebra [50]: Goodman, Ngyuen, and Walker developed a conditional event algebra based on the idea that a conditional event $(A \mid B)$ is the set of the solutions to the *modus ponens* equation:

$$(A \mid B) \wedge_2 B = B \cap A \tag{8.155}$$

The so-called Goodman-Nguyen-Walker (GNW) algebra has the following logical operations:

$$\neg_2 (A \mid B) = (\neg A \mid B) \tag{8.156}$$

$$(A \mid B) \wedge_2 (C \mid D) = (AC \mid \neg AB \vee \neg CD \vee BD) \tag{8.157}$$

$$(A \mid B) \vee_2 (C \mid D) = (A \vee C \mid AB \vee CD \vee BD) \tag{8.158}$$

The operators of this algebra satisfy properties of commutativity, associativity, distributivity, idempotence, the De Morgan Laws, the *modus ponens* equation, and transitive logical chaining. However, this logic is not fully compatible with conditional probability because the function $(A \mid B) \to P(A \mid B)$ is not a probability measure, and GNW is not a Boolean algebra.

8.7.3 Final Remarks

These are the final remarks for this section:

1. CE algebra gives a significance to the conditional event object in order to use this object to model rules in natural language and use probability measures for their quantification.
2. CE algebra is used to manipulate and evaluate rules in knowledge-based and expert systems, such that standard probability theory can be applied.
3. Conditional events lie at the core of the development of a computationally tractable theory of plausible reasoning.

8.8 Random-Set Theory

The concept of random sets was introduced in the early 1970s [52, 53] as a generalization of the random variable concept: Random sets are random elements whose values are sets, whereas random variables are random elements whose values are numbers. A simple example of a random set is a confidence interval. More recently, Goodman, Mahler, and Ngyuen [12] presented random-set theory as a unifying paradigm for most theory of uncertain reasoning. It appears that at least probability theory, Dempster-Shafer theory, possibility theory, fuzzy-set theory, and conditional events can be represented in the random-set framework. Hence, such a unification offers a systematic methodology for the fusion of information involving various types of uncertainty. On the other hand, they developed finite-set statistics (FISST) as a "version of random set theory specifically designed to multiple-targets multiple-sensors applications." Within this approach, they intend to put into a single probabilistic framework most of the different aspects of data fusion, allowing most of the classical rules of probability be used in a larger universe, the power set of this universe. Although this extension of classical point variable statistics and probability is not yet widely studied, it remains, we think, a very interesting candidate for a global assessment of the situation.

Random set: Let $(\Omega, \sigma_\Omega, \text{Prob})$ be a probability space, and let Θ be a finite discrete set (e.g., the universe, the frame of discernment) and $\mathbf{P}(\Theta) = 2^\Theta$ its power set. A random set of Θ is a set-valued random element χ from Ω to $\mathbf{P}(\Theta)$:

$$\chi : \Omega \to (\Theta) \tag{8.159}$$

$$\omega \to \chi(\omega) \tag{8.160}$$

A random set is thus a multivalued mapping from Ω to $\mathbf{P}(\Theta)$. Remember that a random variable X is a (single-valued) mapping from Ω to $\mathbb{R}$, such that $X(\omega) = x \in \mathbb{R}$. X is thus a special case of a random set.

Density function: A random set χ is completely defined by its density function, a function from $\mathbf{P}(\Theta)$ to [0, 1] defined by

$$f(A) = \text{Prob}(\{\omega \mid \chi(\omega) = A\}) = \text{Prob}(\chi = A) \tag{8.161}$$

and satisfying

$$\sum_{A \in \mathbf{P}(\Theta)} f(A) = 1 \tag{8.162}$$

$f(A)$ is thus the probability of the event $\{\chi = A\}$, that is, the probability that the random set χ takes the particular value (set) A of $\mathbf{P}(\Theta)$ ($\{\chi = A\}$ is the set of ω such that $\chi(\omega) = A$).

Distribution function: Equivalently to random variables, a distribution function of χ can be defined as

$$F(A) = \mathrm{Prob}(\chi \subseteq A) = \sum_{B \subseteq A} f(B) \tag{8.163}$$

$F(A)$ is also called a *belief function* as it satisfies the axioms of Shafer's belief functions.[4] More details on this link will be given bellow. Note that $\{\chi \subseteq A\} = \{\omega \mid \chi(\omega) \subseteq A\}$.[5]

Möebius transform: Let χ be a random set of Θ (finite) with F as distribution function. Then, the Möebius transform defines the density function of χ from F as

$$f(A) = \sum_{B \subseteq A} (-1)^{|A - B|} F(B) \tag{8.164}$$

This transformation represents the counterpart of derivatives in the random-set framework. Another approach consists of extending the concept of Radon-Nikodým derivatives to the case of nonadditive set functions.

Finite-set statistics: Finite-set statistics [54] are an extension of practical statistics to multisensor, multitarget data fusion. In this theoretical framework, basics concepts such as expectations, covariances, densities, and the like, are defined as direct analogs of the classical ones, except that they are defined on the power set of the universe instead of the universe itself. Moreover, extensions of classical estimators (e.g., maximum a posteriori, maximum likelihood, Bayesian) are built.

8.8.1 Links to Other Theories

8.8.1.1 Probability Theory

Random-set theory reduces to probability theory as soon as χ is a random variable X. Indeed, only one-element sets (singletons) have non-null probability.

8.8.1.2 The Dempster-Shafer Theory

The link between random sets and belief functions has been studied by many authors, especially by Nguyen [11]. It appears that random-set theory is the probabi-

4. Note that we should write $f_X(A)$ or $F_X(A)$, but the subscript will be avoided when no ambiguity exists.
5. In the following, we will simply note $\chi = A$ for $\{\omega \mid \chi(\omega) = A\}$, $\chi \subseteq A$ for $\{\omega \mid \chi(\omega) \subseteq A\}$ . . .

listic interpretation of the Dempster-Shafer theory. Let χ be a random set of the finite set Θ, and let m be a basic probability assignment from $\mathbf{P}(\Theta)$ to [0, 1]. Then, m defines a density function on $\mathbf{P}(\Theta)$:

$$m(A) = \text{Prob}(\chi = A) = f(A),\ \forall A \in \mathbf{P}(\Theta) \tag{8.165}$$

where f is introduced in (8.161). Under the closed-world assumption, m must also satisfy $\Sigma_{A \subseteq \Theta} m(A) = 1$ and $m(\emptyset) = 0$. It follows that the belief function is

$$Bel(A) = \text{Prob}(\chi \subseteq A) = F(A),\ \forall A \in \mathbf{P}(\Theta) \tag{8.166}$$

where F is introduced in (8.163), and the plausibility function is

$$Pl(A) = \text{Prob}(\chi \cap A \neq \emptyset),\ \forall A \in \mathbf{P}(\Theta) \tag{8.167}$$

Finally, if m_1 and m_2 are two density functions on Θ corresponding to the statistically independent random set χ_1 and χ_2, then Dempster's rule of combination can be formulated by

$$m_1 \oplus m_2(A) = \text{Prob}(\chi_1 \cap \chi_2 = A \mid \chi_1 \cap \chi_2 \neq \emptyset),\ \forall A \in \mathbf{P}(\Theta) \tag{8.168}$$

$\text{Prob}(\chi_1 \cap \chi_2 = \emptyset)$ is thus the conflict factor. The unnormalized Dempster's rule of combination corresponds to the intersection-independent, nonempty, random subsets. The normalization is then a restriction to the class of nonempty, random subsets. The only difference between the random set and Dempster-Shafer frameworks is that in the latter each expert's opinion is represented by a random set, whereas in the former, the entire set of knowledge is represented by a random set [55]. The mathematical analogy between random-set theory and the transferable belief model has been presented by Smets [10].

8.8.1.3 Fuzzy-Set Theory

Goodman [56] makes a formal connection between random sets and fuzzy sets. He shows how fuzzy sets can be considered equivalence classes of random sets. Let $(\Omega, \sigma_\Omega, \text{Prob})$ be a probability space and Θ be a finite space. Thus, with each random set χ from Ω to $\mathbf{P}(\Theta)$, we can associate a membership function $X: \Theta \to [0, 1]$ of a fuzzy set on Θ, such that

$$X(\theta) \triangleq \text{Prob}(\theta \in \chi),\ \forall \theta \in \Theta \tag{8.169}$$

$X(\theta)$ is the one-point covering function of χ. Conversely, if μ is a membership function of a fuzzy set on Θ, then there exists a random set

$$\chi_{\overline{X}}(\mu) = \{\theta \in \Theta \mid X(\theta) \leq \mu(\theta)\} \tag{8.170}$$

where X is a uniformly distributed r.v. on [0, 1]. Because X is uniform, it can be shown that

$$\mu_{\chi_X}(\mu) = \mu \tag{8.171}$$

This connection between both theories does not mean that randomness can capture fuzziness. However, it allows the theory of probability (through random-set theory) to be used to manipulate fuzzy sets. Because fuzzy sets are "one-point coverages" of random sets, the classical fuzzy operators have a set-theoretic counterpart in random-set theory [57].

Another discussion on this subject can be found in Orlov [58]. In this paper, Orlov uses the mathematically well-developed theory of random sets to support the lack of theorems in the fuzzy-set theory. He thinks that random-set theory will probably find new applications everywhere the fuzzy concept is used.

8.8.1.4 Possibility Theory

As a special case of the Dempster-Shafer theory, the possibility theory can be easily linked to random-set theory.

8.8.1.5 Conditional Event Algebra

Mahler [59, 60] shows that there is at least one way to represent knowledge-based rules in random set form. Let $(A \mid B)$ be a conditional event of the GNW algebra, $A, B \subseteq \Theta$ being a finite universe, and let $\mathcal{U}$ be a uniformly distributed random subset on Θ, that is, $\text{Prob}(u) = 1/2^N$, $\forall u \subseteq \Theta$, where $N = |\Theta|$. Thus, we define the random subset $\chi_U(A \mid B)$ associated to the CE $(A \mid B)$ by

$$\chi_U(A \mid B) \triangleq (A \cap B) \cup (\overline{B} \cup \nu) \tag{8.172}$$

Hence, the correspondence between the conditional event $(A \mid B)$ and its associated random set $\chi_U(A \mid B)$ is well defined:

$$(A \mid B) = (C \mid D) \Leftrightarrow \chi_U(A \mid B) = \chi_U(C \mid D) \tag{8.173}$$

Moreover, $\chi_U(A \mid \Theta) = A$, $\forall A \subseteq \Theta$.

8.8.2 Calculus and Reasoning (Aggregation/Fusion)

Two random subsets χ_1 and χ_2 are statistically independent if, and only if,

$$\text{Prob}(\chi_1 = A, \chi_2 = B) = \text{Prob}(\chi_1 = A)\text{Prob}(\chi_2 = B) \tag{8.174}$$

where Prob is a probability measure on Θ. This kind of independence corresponds to independent pieces of evidence, the condition required by Dempster's rule.

Operations on random sets: In general, usual properties, as well as usual operations, of classical sets stay valid for random sets. For example, for all $\omega \in \Omega$,

Random-set complement: $$\overline{\chi}(\omega) = \overline{\chi(\omega)} \tag{8.175}$$

Random-set intersection:

$$(\chi_1 \cap \chi_2)(\omega) = \chi_1(\omega) \cap \chi_2(\omega) \tag{8.176}$$

Random-set union: $$(\chi_1 \cup \chi_2)(\omega) = \chi_1(\omega) \cup \chi_2(\omega) \tag{8.177}$$

It follows, for example,

$$\text{Prob}(\chi_1 \cap \chi_2 = A) = \sum_{X_1 \cap X_2 = A} \text{Prob}(\chi_1 = X_1, \chi_2 = X_2) \tag{8.178}$$

and

$$F_{\chi_1 \cup \chi_2}(A) = F_{\chi_1}(A) + F_{\chi_2}(A) - F_{\chi_1}(A) F_{\chi_2}(A) \tag{8.179}$$

if χ_1 and χ_2 are independent.

Minkowski addition:

$$\chi_1 \oplus \chi_2 = \{x_1 + x_2 \mid x_1 \in \chi_1, x_2 \in \chi_2\} \tag{8.180}$$

Minkowski subtraction: $$\chi_1 \ominus \chi_2 = \{x_1 \mid x_1 + \chi_2 \subseteq \chi_1\} \tag{8.181}$$

Minkowski addition generalizes the addition in $\mathbb{R}^d$ (vectorial addition) and corresponds to dilatation. Minkowski subtraction corresponds to erosion.

8.8.3 Dealing with Uncertainty

Being a general theory, random-set theory is able to represent every kind of uncertainty (e.g., randomness, nonspecificity, vagueness, conflict), and the way this is accomplished is thus described in each specific theory. Moreover, the corresponding measures of uncertainty are defined in each theory.

8.8.4 Final Remarks

These are the final remarks on random sets:

1. Random sets provide a general framework for representing and manipulating different kinds of uncertainty and appears to be a unifying framework in which most theories of quantitative uncertain reasoning can be justified.
2. A unified framework is helpful in combining multiple formalisms possibly present in a large problem solver such as situation analysis [55].
3. Random-set formalism is, however, rarely used, probably due to the large memory size it requires ($2^N - 1$ values must be stored to completely describe a density function of a random set defined for a universe of N elements). For existing restrictions, we refer to the Dempster-Shafer theory, which faces exactly the same problem.
4. The random-set framework offers justifications of basic operations on fuzzy sets, which are mainly ad hoc [55, 58].
5. Finite-sets statistics is a tool for multiple-target, multiple-sensor tracking [11, 54].

References

[1] Klir, G. J., and T. A. Folger, *Fuzzy Sets, Uncertainty, and Information*, Upper Saddle River, NJ: Prentice-Hall, 1988.

[2] Poole, D., "Learning, Bayesian Probability, Graphical Models, and Abduction," in P. Flach and A. Kakas, (eds.), *Abduction and Induction: Essays on Their Relation and Integration*, Norwell, MA: Kluwer Academic Publishers, 1998.

[3] Dubois, D., and H. Prade, "La fusion d'informations imprecises," *Traitement du Signal*, Vol. 11, No. 6, 1994, pp. 447–458.

[4] Dempster, A., "Upper and Lower Probabilities Induced by Multivalued Mapping," *Ann. Math. Statist.*, Vol. 38, 1967, pp. 325–339.

[5] Shafer, G., *A Mathematical Theory of Evidence*, Princeton, NJ: Princeton University Press, 1976.

[6] Barnett, J. A., "Computational Methods for a Mathematical Theory of Evidence," *Proc. 7th Int'l Joint Conf. AI*, Vancouver, WA, 1981, pp. 868–875.

[7] Shafer, G., and R. Logan, "Implementing Dempster's Rule for Hierarchical Evidence," *Artificial Intelligence*, Vol. 33, 1987, pp. 271–298.

[8] Gordon, J., and E. H. Shortliffe, "A Method for Managing Evidential Reasoning in Hierarchical Hypothesis Spaces," *Artificial Intelligence*, Vol. 26, 1985, pp. 323–358.

[9] Smets, P., and R. Kennes, "The Transferable Belief Model," *Artificial Intelligence*, Vol. 66, 1994, pp. 191–234.

[10] Smets, P., "The Transferable Belief Model and Random Sets," *International Journal of Intelligent Systems*, Vol. 7, 1992, pp. 37–46.

[11] Nguyen, H. T., "On Random Sets and Belief Functions," *Journal of Mathematical Analysis and Applications*, Vol. 65, 1978, pp. 531–542.

[12] Goodman, I. R., R. P. S. Mahler, and H. T. Nguyen, "Mathematics of Data Fusion," in H. J. Skala, (ed.), *Series B: Mathematical and Statistical Methods,* Norwell, MA: Kluwer Academic Publishers, 1997, p. 507.

[13] Goodman, I. R., and G. F. Kramer, "Extension of Relational and Conditional Event Algebra to Random Sets with Application to Data Fusion," in J. Goustias, R. P. S. Mahler, and H. T. Nguyen, (eds.), *Random Sets: Theory and Applications*, New York: Springer-Verlag, 1997, pp. 209–242.

[14] Yaghlane, B., B. P. Smets, and K. Mellouli, "Independence Concepts for Belief Functions," *8th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Madrid, Spain, July 3–7, 2000.

[15] Yaghlane, B., B. P. Smets, and K. Mellouli, "Independence and Non-Interactivity in the Transferable Belief Model," in F. M. Studeny, (ed.), *Workshop on Conditional Independence Structures and Graphical Models*, Fields Institute, Toronto, September 29–October 1, 1999.

[16] Bauer, M., "Approximation Algorithms and Decision Making in the Dempster-Shafer Theory of Evidence—An Empirical Study," *International Journal of Approximate Reasoning*, Vol. 17, Nos. 2–3, 1997, pp. 217–237.

[17] Voorbraak, F., "On the Justification of Dempster's Rule of Combination," *Artificial Intelligence*, Vol. 48, 1991, pp. 171–197.

[18] Tessem, B., "Approximations for Efficient Computation in the Theory of Evidence," *Artificial Intelligence*, Vol. 61, No. 2, 1993, pp. 315–329.

[19] Voorbraak, F., "A Computationally Efficient Approximation of Dempster-Shafer Theory," *Int. J. Man-Machine Studies*, Vol. 30, 1989, pp. 525–536.

[20] Jousselme, A.-L., D. Grenier, and E. Bossé, "Analyzing Approximation Algorithms in the Theory of Evidence," *Proceedings of SPIE*, Vol. 4731, Orlando, FL, April 3–5, 2002, pp. 65–74.

[21] Smets, P., "Belief Functions: The Disjunctive Rule of Combination and the Generalized Bayesian Theorem," *International Journal of Approximate Reasoning*, Vol. 9, 1999, pp. 1–35.

[22] Walley, P., "Measures of Uncertainty in Expert Systems," *Artificial Intelligence*, Vol. 83, No. 1, 1996, pp. 1–58.

[23] Suppes, P., and M. Zanotti, "Necessary and Sufficient Conditions for Existence of a Unique Measure Strictly Agreeing with a Qualitative Probability Ordering," *Journal of Philosophical Logic*, Vol. 5, 1976, pp. 431–438.

[24] Liu, C., et al., "Measuring of Ambiguity in the Evidence Theory," *IEEE Transactions on Systems, Man and Cybernetics—Part A*, Vol. 36, No. 5, September 2006, pp. 890–903.

[25] Liu, C., et al., "Reducing Algorithm Complexity for Computing Aggregate Uncertainty," *IEEE Transactions on Systems, Man and Cybernetics*, accepted for publication, 2006.

[26] Smarandache, F., and J. Dezert, (eds.), *Applications and Advances of DSmT for Information Fusion*, Rehoboth, MA: American Research Press, 2004.

[27] Zadeh, L. A., "Fuzzy Sets," *Information and Control*, Vol. 8, 1965, pp. 338–353.

[28] Dubois, D., and H. Prade, "Fuzzy Sets and Probability: Misunderstandings, Bridges and Gaps," *Proceedings of the 2nd IEEE International Conference on Fuzzy Systems*, San Francisco, CA, March 28–April 1, 1993.

[29] Black, M., "Vagueness: An Exercise in Logical Analysis," *Philosophy of Science*, Vol. 4, 1937, pp. 427–455.

[30] Kosko, B., "Fuzziness vs. Probability," *International Journal of General Systems*, Vol. 17, 1990, pp. 211–340.

[31] Ma, M., A. Kandel, and M. Friedman, "Representing and Comparing Two Kinds of Fuzzy Numbers," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 28, No. 4, 1998, pp. 573–576.

[32] Parsons, S., and A. Hunter, "A Review of Uncertainty Handling Formalisms," in A. Hunter and S. Parsons, (eds.), *Applications of Uncertainty Formalisms*, New York: Springer, 1998.

[33] Zadeh, L. A., "Fuzzy Sets as a Basis for a Theory of Possibility," *Fuzzy Sets and Systems*, Vol. 1, 1978, pp. 3–28.

[34] Dubois, D., and H. Prade, "Possibility Theory in Information Fusion," *ISIF Proceedings of Fusion 2000*, Paris, France, July 10–13, 2000.

[35] Dubois, D., and H. Prade, "Conditional Objects as Nonmonotonic Consequence Relationships," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 24, No. 12, 1994, pp. 1724–1740.

[36] Dubois, D., and H. Prade, "An Introduction to Possibilistic and Fuzzy Logics," in P. Smets, et al., (eds.), *Non-Standard Logics for Automated Reasoning*, New York: Academic Press, 1988, pp. 1–40.

[37] Klir, G. J., and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Upper Saddle River, NJ: Prentice Hall International, 1995, p. 574.

[38] Drago, G. P., and S. Ridella, "Possibility and Necessity Pattern Classification Using an Interval Arithmetic Perceptron," *Neural Computing and Applications*, Vol. 8, 1999, pp. 40–52.

[39] Pawlak, Z., "Rough Sets and Fuzzy Sets," *Fuzzy Sets and Systems*, Vol. 17, 1985, pp. 99–102.

[40] Pawlak, Z., *Rough Sets—Theoretical Aspects of Reasoning About Data,* Norwell, MA: Kluwer, 1991.

[41] Pawlak, Z., "Rough Sets: A New Approach to Vagueness," in L. A. Zadeh and J. Kacprzyk, (eds.), *Fuzzy Logic for the Management of Uncertainty*, New York: John Wiley & Sons, 1992, pp. 105–111.

[42] Pawlak, Z., "Rough Set Approach to Knowledge-Based Decision Support," *14th European Conference on Operational Research*, Jerusalem, Israel, July 3–6, 1995.

[43] Parsons, S., M. Kubat, and M. Dohnal, "A Rough Sets Approach to Reasoning Under Uncertainty," *Journal of Experimental and Theoretical Artificial Intelligence*, Vol. 7, 1995, pp. 175–193.

[44] Polkowski, L., and A. Skowron, "Rough Mereology: A New Paradigm for Approximate Reasoning," *International Journal of Approximate Reasoning*, Vol. 15, 1996, pp. 333–365.

[45] Dubois, D., and H. Prade, "Rough Fuzzy Sets and Fuzzy Rough Sets," *International Journal of General Systems*, Vol. 17, No. 2–3, 1990.

[46] Pawlak, Z., "Vagueness—A Rough Set View," in J. Mycielski, G. Rozenberg, and A. Salomaa, (eds.), *Structures in Logic and Computer Science*, New York: Springer, 1997, pp. 106–117.

[47] Skowron, A., "The Relationship between the Rough Sets and the Evidence Theory," *Bulletin of the Polish Academy of Science*, Vol. 37, Nos. 1–2, 1989, pp. 87–90.

[48] Li, Y.-R., and Jiang, J.-P., "The Integrated Methodology of Rough Sets, Fuzzy Logic and Genetic Algorithms for Multisensor Fusion," *Proceedings of the American Control Conference*, Arlington, VA, June 25–27, 2001.

[49] Calabrese, P. G., "Deduction and Inference Using Conditional Logic and Probability," in I. R. Goodman, H. T. Nguyen, and G. S. Rogers, (eds.), *Conditional Logic in Expert Systems*, New York: Elsevier, 1991, pp. 71–100.

[50] Goodman, I. R., H. T. Nguyen, and E. A. Walker, (eds.), *Conditional Inference and Logic for Intelligent Systems*, New York: Elsevier, 1991.

[51] Schay, G., "An Algebra of Conditional Events," *Journal of Math. Anal. and Appl.*, Vol. 24, 1968, pp. 334–344.

[52] Kendall, D. G., "Foundations of a Theory of Random Sets," in E. F. Harding and D. G. Kendall, (eds.), *Stochastic Geometry*, New York: John Wiley & Sons, 1974, pp. 322–376.

[53] Matheron, G., *Random Sets and Integral Geometry*, New York: John Wiley & Sons, 1975.

[54] Mahler, R. P. S., *An Introduction to Multisource-Multitarget Statistics and Its Applications*, Lockheed-Martin Technical Report, 2000.

[55] Kreinovich, V., "Random Sets Unify, Explain, and Aid Known Uncertainty Methods in Expert Systems," in J. Goustias, R. P. S. Mahler, and H. T. Nguyen, (eds.), *Random Sets: Theory and Applications*, New York: Springer-Verlag, 1997, pp. 321–345.

[56] Goodman, I. R., "Fuzzy Sets as Equivalence Classes of Random Sets," in R. R. Yager, (ed.), *Fuzzy Set and Possibility Theory*, New York: Pergamon, 1982, pp. 327–342.

[57] Quinio, P., and T. Matsuyama, "Random Closed Sets: A Unified Approach to the Representation of Imprecision and Uncertainty," in R. K. P. Siegel, (ed.), *Symbolic and Quantitative Approaches to Uncertainty*, New York: Springer, 1991, pp. 282–286.

[58] Orlov, A. I., "Fuzzy and Random Sets," in *Prikladnoi Mnogomerni Statisticheskii Analys.*, 1978, pp. 262–280 (in Russian).

[59] Mahler, R. P. S., "Representing Rules as Random Sets, I: Statistical Correlations between Rules," *Information Sciences*, Vol. 88, 1996, pp. 47–68.

[60] Mahler, R. P. S., "Unified Data Fusion: Fuzzy Logic, Evidence and Rules," *SPIE Proceedings*, Vol. 2755, 1996, pp. 226–237.

CHAPTER 9

Hybrid and Graphical Approaches

Patrick Maupin, Anne-Laure Jousselme, and Éloi Bossé

9.1 Introduction

The previous two chapters addressed the qualitative and quantitative approaches available to model situation analysis and high-level data fusion. As already mentioned, qualitative approaches seem better suited to reasoning about knowledge, while quantitative approaches are better candidates for uncertainty representation and management. This chapter addresses hybrid approaches that can be used for a global modelization of a situation. Such approaches (quantitative logics, incidence calculus) mix quantified evaluations of uncertainty and high reasoning capabilities. This chapter also includes graphical or graph-based approaches (e.g., Bayesian networks) that support the graphical representation and propagation of knowledge.

9.2 Discussion of Quantitative Logics

Roughly speaking, logic is associated with qualitative approaches, whereas mathematics is associated with quantitative approaches. Although quantitative logics exist, being a sort of bridge between these two worlds, depending on the kind of approach used, distinct vocabulary and symbols are used to designate equivalent things. We think that a first step to giving a uniform overview of the theories is to make a correspondence between both worlds. On one hand, it is familiar for a logician to talk about a proposition (or more generally a formula) ϕ, which implies another proposition (or formula) ψ, without any reference to a particular universe of discourse. The fact that *it is raining* (ϕ) implies that *the ground is wet* (ψ), written $\phi \rightarrow \psi$. On the other hand, a mathematician will use the set-theoretic vocabulary and talk about *events*, which are subsets of a universe. From a mathematical (set-theoretic) point of view, the universe of discourse (frame of discernment) Θ is a set containing all the possible outcomes for a given experiment. An element of Θ is often called an *object* as it is characterized by a finite number of attributes. If $\Theta = \{\theta_1, \theta_2, \ldots,\}$, θ_i is characterized by m features f_j (θ_i) can be seen as a vector with m components $\theta_i = [f_1^i f_2^i \ldots f_n^i]^T$). f_1 could be the weather condition: $f_1 = 1$ if it is raining, $f_1 = 2$ if it is not raining, $f_1 = 3$ if it is sunny, and so on. Another characteristic, f_2, could be the state of the ground: f_2 = dry, f_2 = wet. From a set-theoretic point of view, *it is raining* refers to a subset A of Θ, and *the ground is wet* refers to another subset B of Θ: A is the set of elements of Θ having feature f_1 equal to 2; B is the set of elements of Θ having the feature f_2 equal to *wet*. A subset of Θ is called an *event*.

For a logician, Θ is the set of possible worlds. In general, it is referred to by $\mathbf{P} = \{\phi_1, \phi_2, \ldots, \phi_k\}$ as the set of formulae. In a qualitative approach, a world θ_i is not attached to a set of features but to a set of true formulae. A corresponds thus to the set of worlds in which ϕ is TRUE, and B is the set of worlds in which is TRUE. The logical implication $\phi \rightarrow \psi$ is written then as $A \subseteq B$, meaning that all the worlds θ_i, where ϕ = *rainy day* is ψ TRUE, also have a wet ground (ψ = *wet ground* is TRUE).

The extension of this example to other connectors (or set relations) is straightforward. Table 9.1 summarizes the main symbols and terms and their equivalences in both languages.

9.3 Probabilistic Logic

Probabilistic logic is a logic for reasoning about probabilities (i.e., the probability calculus on rules) of logical formulas. It has been introduced by Reichenbach [1] and followed by Carnap [2]. More recently, this theory has been revisited by Nilsson as a "semantical generalization of logic in which the truth values of sentences are probability values" [3]. The principle of probabilistic logic lies in the fact that the truth-values of propositions are their probability of occurrence. The purpose is then to deal with propositional probabilities (i.e., probabilities assigned to particular propositions or assertions). The theoretical basics remain those described in Chapters 7 and 8, except that the events are replaced by logical formulas, which we will denote by ϕ or ψ. Probabilistic logic then combines logic with probability theory and reduces to ordinary logic when the probabilities of all sentences are either 0 or 1. This approach is based in the possible-worlds semantic. Other approaches have been developed by Halpern [4] and Fagin, Halpern, and Megiddo [5].

The theoretical basics of probabilistic logic are those of probability theory on one hand and of classical logic on the other. So we refer the reader to previous

Table 9.1 Logical Versus Set-Theoretical Notions

Logical Notions	*Set-Theoretic Notions*
Propositions ϕ	Subsets A
Conjunction $\wedge$ or &	Intersection $\cap$
Disjunction $\vee$	Union $\cup$
Implication $\rightarrow$	Inclusion $\subset$, $\subseteq$
Are, Is	Belonging $\in$
Negation $\neg$ or	Complementation $\overline{A}$
Contradiction	Empty set $\emptyset$
Tautology T	Frame of discernment Θ
Knowledge Base	Network
Observations	Evidence

chapters. In this part, we will essentially give the differences in notations of both kinds of probabilities, propositional and statistical.

Propositional and statistical probabilities: We note the difference between propositional probabilities and statistical probabilities. Propositional probabilities are probabilities assigned to particular propositions or assertions, whereas statistical probabilities make assertions about the proportion of individuals from a particular set that are members of some other set, such as the proportion of individuals having a fitness equal to or higher than the average in the real population (out of all possible populations). We may also view this as attributing a property to a proportion of individuals in a set with a certain probability.

Random variable: A *random variable* is a term in a language that can take on different values [6]. In the propositional approach, a random variable takes its value in some set. This set of all the possible values a variable can take is called its *domain*. We keep the notation of X for the random variable, and we will denote by $\mathcal{D}_X$ its domain (e.g., $\mathcal{D}_X = \mathbb{R}$, but it can be something else). The possible values X can take are then $x_1, x_2, \ldots,$ elements of $\mathcal{D}_X$. A *Boolean r. v.* is one whose domain has only two values. In general, we note $\mathcal{D}$ = {TRUE,FALSE}, but also $\mathcal{D}$ = {YES,NO}, $\mathcal{D} = \{0, 1\}$, and so on. Rather than writing X = TRUE, we simply write X and also $\neg X$ for X = FALSE.

Possible world: Let $X_1, X_2, \ldots$ be a set of random variables, and let Θ be the set of possible worlds. A possible world assigns one value to each random variable:

$$\theta \vDash X = x \equiv X(\theta) = x \tag{9.1}$$

means that the variable X is assigned value x in the world $\theta \in \Theta$. The right-hand side of the equivalence corresponds to the notations of statistical probabilities.

Proposition and formula: A *proposition* is a Boolean formula made from assignments of values to variables. For example, $\phi = [X = x]$ is the proposition that variable X has value x. It can either be true or false. A proposition is equivalent to an event A for statistical probabilities.

Probability distribution and measure: A *probability distribution* is a mapping from Θ, the set of possible worlds, to $[0, 1]$, assigning to each world a measure between 0 and 1, such that

$$\sum_{\theta \in \Theta} p(\theta) = 1 \tag{9.2}$$

The *probability* of a proposition is then

$$P(\phi) = \sum_{\theta \vDash \phi} p(\theta) \tag{9.3}$$

which is the sum of all the measures of the worlds in which is TRUE.

Certainty: A formula ϕ is certain if its probability is 1.

The major difference compared with propositional probabilities is that the statistical probability operator must specify a set of placeholder variables—we are not talking about a particular individual but about a set of individuals.

Statistical probability: Let ϕ be a formula and x a vector of n objects. Then, $P(\phi)$ is the statistical probability of ϕ. We also need a measuring function, known in statistics and probability theory as a random variable (see any standard reference on probabilities). These variables are used to map individual objects (or properties of objects) to real numbers in order to discuss these objects or properties.

9.4 Fuzzy Logic

One of the first papers proposing fuzzy logic for natural-language modeling was written by Lakoff [7]. For a point of view arguing that fuzzy logic is nothing other than classical logic in disguise, see Elkan [8] and the well-argued answer of Klir and Yuan [9].

Degrees of memberships $\mu_A(\theta)$ are truth-values x.

Standard Lukasiewicz logic $L_{\aleph 0}$: This infinite-valued logic is isomorphic to fuzzy-set theory based on standard fuzzy operators. Let x and y be truth values of [0, 1]:

$$\neg x = 1 - x \tag{9.4}$$

$$x \wedge y = \min(x, y) \tag{9.5}$$

$$x \vee y = \max(x, y) \tag{9.6}$$

$$x \to y = \min(1, 1 + x - y) \tag{9.7}$$

$$x \leftrightarrow y = 1 - |x - y| \tag{9.8}$$

Fuzzy propositions: The fundamental difference between classical propositions and fuzzy propositions is in the range of their truth-values [10]. Let Θ be a universe, θ an element of Θ, and A and B two fuzzy sets of Θ (A, B are *fuzzy predicates*.). We can distinguish four types of fuzzy propositions:

1. *Unconditional and unqualified fuzzy propositions:* ϕ: [θ is A];
2. *Unconditional and qualified fuzzy propositions:* ϕ: Prob([θ is A]) is P;
3. *Conditional and unqualified fuzzy propositions:* ϕ: if [θ is A], then [ω is B];
4. *Conditional and qualified fuzzy propositions:* ϕ: Prob([θ is A] | [ω is B]) is P.

Fuzzy quantifiers: In general, these are fuzzy numbers that take part in fuzzy propositions [9]. They are of two kinds:

1. Defined on $\mathbb{R}$ and characterizing linguistic terms such as *about 10, much more than 100, at least about 5*, and so forth:
   a. ϕ: "There are k n's in N such that [$\theta(n)$ is A];"
   b. ϕ: "There are k n's in N such that [$\theta_1(n)$ is A_1] and [$\theta_2(n)$ is A_2];"

2. Defined on [0, 1] and characterizing linguistic terms such as *almost all, about half, most*, and so on:
   a. ϕ: "Among n's in N such that [$\theta_1(n)$ is A_1], there are k n's in N such that [$\theta_2(n)$ is A_2]."

Linguistic hedges: These are linguistic terms, such as *very, more or less, fairly, extremely*, and so on, used to modify fuzzy predicates, fuzzy truth-values, and fuzzy probabilities:

$$\phi\text{: } [\theta \text{ is } A] \rightarrow H\phi\text{: } [\theta \text{ is } HA]$$

9.4.1 Calculus and Reasoning

In fuzzy logic, we use the generalizations of classical inference rules, such as

Generalized modus ponens

$$\begin{array}{cc} \text{Rule:} & \phi_1\text{: "If } \theta_1 \text{ is } A\text{, then } \theta_2 \text{ is } B\text{"} \\ \text{Fact:} & \phi_2\text{: "}\theta_1 \text{ is } A_1\text{"} \\ \text{Conclusion:} & \Psi\text{: "}\theta_2 \text{ is } B_1\text{"} \end{array} \tag{9.9}$$

Generalized modus tollens

$$\begin{array}{cc} \text{Rule:} & \phi_1\text{: "If } \theta_1 \text{ is } A\text{, then } \theta_2 \text{ is } B\text{"} \\ \text{Fact:} & \phi_2\text{: "}\theta_2 \text{ is } B_1\text{"} \\ \text{Conclusion:} & \Psi\text{: "}\theta_1 \text{ is } A_1\text{"} \end{array} \tag{9.10}$$

Generalized hypothetical syllogism

$$\begin{array}{cc} \text{Rule:} & \phi_1\text{: "If } \theta_1 \text{ is } A\text{, then } \theta_2 \text{ is } B\text{"} \\ \text{Fact:} & \phi_2\text{: "}\theta_2 \text{ is } B\text{, then is } C\text{"} \\ \text{Conclusion:} & \Psi\text{: "}\theta_1 \text{ is } A\text{, then is } C\text{"} \end{array} \tag{9.11}$$

9.5 Possibility Logic

Let **A** be a set of axioms $\mathbf{A} = \{\phi_1, \phi_2, \ldots, \phi_n\}$. In possibilistic logic, a grade of possibility $\Pi(\phi_i)$ and a grade of necessity $N(\phi_i)$ that ϕ_i is true is assigned to each formula of **A**, $i = 1, \ldots, n$.

9.5.1 Calculus and Reasoning

Inference: Basic patterns of classical logic have been extended by possibilistic logic [11]:

$$\begin{array}{ll} \text{Modus ponens} & \Pi(\psi) \geq N(\psi) \geq \min(N(\phi), N(\phi \rightarrow \psi) \\ \text{Modus tollens} & \Pi(\phi) \leq N(\phi) \leq \max(\Pi(\psi), 1 - N(\phi \rightarrow \psi) \end{array} \tag{9.12}$$

9.5.2 Dealing with Uncertainty

The concept of uncertainty in possibility logic is handled this way:

$$\begin{gathered} N(\phi) = 1 \Rightarrow \phi \text{ is TRUE} \\ \Pi(\phi) = 0 \Rightarrow \phi \text{ is FALSE} \\ N(\phi) = 0 \text{ or } \Pi(\phi) = 1 \Rightarrow \text{Total uncertainty about the truth of } \phi \end{gathered} \tag{9.13}$$

Possibilistic logic offers an absolute reference point for expressing ignorance.

$$\begin{gathered} \text{Certainty: } N(\phi) = 1,\ N(\neg\, \phi) = 0 \\ \text{Ignorance: } N(\phi) = N(\neg\, \phi) = 0 \end{gathered}$$

Ignorance cannot be modeled in probability theory, where it is approximated by randomness. Possibility cannot model randomness.

Attach weights of uncertainty to rules "if ϕ, then ψ" in complete accordance with classical logic. The quantity $N(\phi \rightarrow \psi)$ is very close to conditional probability measure:

$$N(\phi \rightarrow \psi) = N(\psi \mid \phi) \triangleq 1 - \Pi(\neg\, \psi \mid \phi) \tag{9.14}$$

Possibilistic logic is a quasi-qualitative calculus where numbers are compared, not added or multiplied. Numbers are useful only to model grade, and no great precision is required.

9.6 Incidence Calculus

Incidence calculus has been proposed by Bundy [12, 13] as a probabilistic logic for reasoning under uncertainty. It is a method for managing uncertainty in a numerical way. Unlike in other numerical approaches, in incidence calculus probabilities are associated with a set of possible worlds rather directly with formulae. The probability of a formula is then calculated through the incidence set assigned to the formula. Incidence calculus itself appears to be a unification of symbolic and numerical approaches. It can therefore be regarded as a bridge between the two reasoning patterns [14]. Incidence calculus is used, for example, to represent default logic and to implement assumption-based truth maintenance systems (ATMS) [14, 15].

Possible worlds: Let Θ note the set of possible worlds. A *possible world* is a primitive object of incidence calculus. It can be understood as a partial interpretation of some logical formula.

Incidence calculus theory: An *incidence calculus theory* is a quintuple (Θ, p, **F**, **A**, i) where

- Θ is the finite set of the possible worlds, $\Theta = \{\theta_1, \theta_2, \ldots, \theta_N\}$;
- p is a probability distribution over Θ such that $p(\theta)$ is the probability of the world θ;

- **F** is the finite set of the propositions, $\mathbf{F} = \{p_1, \ldots, p_n\}$;
- **A** is the set of axioms;
- i is the *incidence function* assigning to each element of **A**, a subset of Θ.

The set of all the formulae is $\mathcal{L}(\mathbf{F})$, called the *language of* **F**. **A** is thus a subset of $\mathcal{L}(\mathbf{F})$.

Incidence function: An *incidence function i* is a mapping from to 2^{Θ} such that

$$i(\phi) = \{\theta \in \Theta \mid \theta \vDash \phi\} \tag{9.15}$$

satisfying two conditions:

$$i(\phi_1 \wedge \phi_1) = i(\phi_1) \cap i(\phi_2) \text{ and } i(\perp) = \varnothing \tag{9.16}$$

$i(\phi)$ is called the incidence set of ϕ and is thus the subset of Θ containing all the worlds θ in which ϕ is TRUE.

Lower and upper bounds: For any formula $\phi \in \mathcal{L}(\mathbf{F})$, $\phi \neq \mathbf{A}$, we can only get a lower bound of its incidence set

$$i_*(\phi) = \bigcup_{(\psi \to \phi) = T} i(\psi) \tag{9.17}$$

and an upper bound:

$$i^*(\phi) = \Theta - i_*(\neg \phi) \tag{9.18}$$

For any $\phi \in \mathbf{A}$, we have $i^*(\phi) = i(\phi)$.

$(\psi \to \phi) = T$ is the *semantical implication* also denoted by ψ *** ϕ, and in (9.15), it means that the disjunction is performed for all $\psi \in \mathcal{L}(\mathbf{F})$ such that $(\psi \to \phi)$ is TRUE, $\phi \in \mathbf{A}$. Note that $(\psi \to \phi) = T \Leftrightarrow \psi \wedge \phi = \psi \Leftrightarrow i(\psi \to \phi) = \Theta$.

Probability: Let p be a probability distribution over Θ. The *weighted probability* is the measure P such that

$$P(A) = \sum_{\theta \in A} p(\theta) \tag{9.19}$$

Equation (9.19) is thus a probability measure, and it follows that $P(\Theta) = 1$. The *probability of the formula* is thus

$$\text{Prob}(\phi) = P(i(\phi)) \tag{9.20}$$

Equivalently, if $\phi \in \mathcal{L}(\mathbf{F}) - \mathbf{A}$, then the probabilities of its lower and upper bound are, respectively,

$$\text{Prob}_*(\phi) = P(i_*(\phi)) \text{ and } \text{Prob}^*(\phi) = P(i^*(\phi)) \tag{9.21}$$

The probabilities of the lower and upper bounds are equivalent to the belief and plausibility functions in the Dempster-Shafer theory. The *conditional probability* is defined as

$$\text{Prob}(\phi \mid \psi) = \frac{\text{Prob}(\phi \wedge \psi)}{\text{Prob}(\psi)} \tag{9.22}$$

Basic incidence assignment: Given a set of axioms **A**, a *basic incidence assignment ii* is an incidence function defined on **A** which satisfies

$$ii(\phi) \cap ii(\psi) = \varnothing \textit{ if } \phi \neq \psi \tag{9.23}$$

and

$$ii(\perp) = \varnothing \textit{ and } ii(T) = \Theta - \bigcup_j ii(\phi_j) \tag{9.24}$$

It follows that the incidence function i can be defined from ii:

$$i(\phi) = \bigcup_{(\psi \to \phi) = T} ii(\psi) \tag{9.25}$$

It can be shown that, given an incidence calculus theory (Θ, p, **F**, **A**, i), there exists a basic incidence calculus assignment for I [14].

9.6.1 Calculus and Reasoning

Liu and Bundy [16] propose a method for combining different pieces of evidence in the incidence calculus framework as an alternative to Dempster's rule of combination. They show that their approach is more powerful than Dempster's rule in the sense that it can be use to combine non-DS-independent pieces of evidence. The combination rule involves two steps:

1. Construction of a joint space;
2. Propagation of the information in this space.

9.6.1.1 Step 1: Two Sets of Possible Worlds

Let (Θ_1, p_1, **F**, $\mathbf{A}_1$, i_1) and (Θ_2, p_2, **F**, $\mathbf{A}_2$, i_2) be two incidence calculus theories whose probability spaces are DS independent. Then, another two incidence calculus theories can be constructed from them, say (Θ, p, **F**, $\mathbf{A}_1$, i_1') and (Θ, p, **F**, $\mathbf{A}_2$, i_2') such that

- Θ is the *joint set* of possible worlds

$$\Theta = (\Theta_1 \times \Theta_2) - \Theta_0 \tag{9.26}$$

with

$$\Theta_0 = \bigcup_{(\phi_1 \wedge \phi_2) = \perp} i_1(\phi_1) \times i_2(\phi_2) \tag{9.27}$$

where "×" is the Cartesian product and "–" is the set subtraction operator. An element of Θ is thus a pair $\theta = (\theta_1, \theta_2)$.

- The new probability distribution on Θ is then

$$p(\theta) = p[(\theta_1, \theta_2)] = \frac{p_1(\theta_1)p_2(\theta_2)}{1 - \sum\limits_{\theta' \in \Theta_0} p_1(\theta_1')p_2(\theta_2')} \tag{9.28}$$

- The new incidence functions are defined by

$$i_1'(\phi_1) = (i_1(\phi_1) \times \Theta_2) - \Theta_0, \ \phi_1 \in \mathbf{A}_1 \tag{9.29}$$

$$i_2'(\phi_2) = (\Theta_1 \times i_2(\phi_2)) - \Theta_0, \ \phi_2 \in \mathbf{A}_2 \tag{9.30}$$

The principle of this step is that

if $\theta_1 \in \Theta_1$ makes $\phi_1 \in \mathbf{A}_1$ true,
and $\theta_2 \in \Theta_2$ makes $\phi_2 \in \mathbf{A}_2$ true,
then $(\theta_1, \theta_2) \in \Theta$ makes $\phi_1 \wedge \phi_2 \in \mathbf{A}$ true.

or, equivalently, if $\theta_1 \in i_1(\phi_1)$ and $\theta_2 \in i_2(\phi_2)$ then $(\theta_1, \theta_2) \in i(\phi_1 \wedge \phi_2)$. However, the elements of Θ that make TRUE must be taken out (these form the set Θ_0).

9.6.1.2 Step 2: One Set of Possible Worlds

Let $(\Theta, p, \mathbf{F}, \mathbf{A}_1, i_1)$ and $(\Theta, p, \mathbf{F}, \mathbf{A}_2, i_2)$ be two incidence calculus theories based on the same set of possible worlds Θ, with p being a probability distribution on Θ. Given two incidence functions i_1 and i_2 from two observations (and their associated sets of axioms), the *joint impact* of information carried by the two theories is represented by a quintuple $(\Theta, p, \mathbf{F}, \mathbf{A}, i)$, where

- $\mathbf{A}$ is the set of axioms comprising the (true) conjunctions of formulae of $\mathbf{A}_1$ and $\mathbf{A}_2$:

$$\mathbf{A} = \{\psi \mid \psi = \phi_1 \wedge \phi_2, \text{ where } \phi_1 \in \mathbf{A}_1, \ \phi_2 \in \mathbf{A}_2, \ \psi \neq \perp\} \tag{9.31}$$

- $i(\psi)$ is the incidence set of the formula $\psi \in \mathbf{A}$ equal to

$$i(\psi) = \bigcup_{(\phi_1 \wedge \phi_2 \rightarrow \psi) = T} i_1(\phi_1) \cap i_2(\phi_2), \ \psi \in \mathbf{A} \tag{9.32}$$

Also, let

$$i(\perp) = \varnothing \text{ and } i(T) = \Theta \tag{9.33}$$

Liu and Bundy have proved that (Θ, p, **F**, **A**, i) is also an incidence calculus theory [16] and that this rule obtained the same results as that of Dempster.

9.6.2 Dealing with Uncertainty

Pieces of evidence are represented by incidence calculus theories, comprising the five elements described above. $ii(T) = \Theta - \cup_j ii(\phi_j)$ is an alternative way to represent ignorance.

9.6.3 Final Remarks

Let us conclude this section with the following remarks:

1. Formal relations have been established with the Dempster-Shafer theory [17].
2. Links have been made with ATMS [14, 15]. We have proven: (a) that managing nodes in an ATMS is equivalent to producing incidence sets of these statements, (b) the equivalence between the justification set for a node and the implication relation set for this node, and (c) that incidence calculus provides a theoretical basis for constructing ATMS.

9.7 Introductory Discussion of Graph-Based Approaches

Bayesian networks appeared in the late 1970s. They were first developed to model distributed processing in reading comprehension. A few years later, Bayesian networks imposed themselves as a general representation scheme for uncertain knowledge. Pearl [18, 19] especially has shown that the marginal distribution for individual nodes can be obtained by using only local computation. It is the basis for graphical representation and propagation of knowledge. Hence, since 1986, several architectures for exact computation of marginals using local computation in uncertainty reasoning have emerged: In 1988, Lauritzen and Speigelhalter [20] gave an alternative architecture for computing marginals, based on junction trees. Jensen, Olesen, and Andersen [21, 22] modified the architecture of Lauritzen and Speigelhalter and then expanded the case of singly connected Bayesian networks considered by Pearl to multiple connected networks. This improved architecture led to more efficient computations. In parallel, other authors were interested in belief-function propagation (instead of probabilities) in graphical schemes, so similar techniques were developed for belief functions. Since 1985, the problem of propagating belief functions in a diagnosis tree has been posed by Gordon and Shortliffe [23]. Because their method involved an approximation, Shafer and Logan [24] proposed an exact implementation of Dempster's rule of combination; at the same time, Shenoy and Shafer [25] presented a general scheme for propagating belief functions with local computation. This last architecture is valid for certain kinds of trees that admit a transformation in qualitative Markov trees; it also generalizes the computational scheme of Shafer and Logan for diagnosis trees, as well as Pearl's scheme for Bayesian causal trees. Led by the idea of generalization, Shenoy and Shafer [25, 26] proposed an abstract framework for computing marginals in join tress. They called this framework a valuation-based system (VBS), in which different formalisms could be considered, such as Bayesian probabilities, belief functions, and possibilities. Some comparison of these different architectures has yet been done [27]. It appears that the most suitable one is that proposed by Shenoy and Shafer [28], which later became known as valuation networks (VNs) or VBSs [29]. An overview of graphical approaches has been recently presented by Pearl [30].

9.7.1 Basic Graph-Theoretic Notions

Here are some basic graph-theoretic notations.

Graphs: A *graph* (or *network*) is a pair $\mathcal{G} = (\mathbf{U}, \mathbf{L})$, where $\mathbf{U}$ is a finite set of *nodes*, and $\mathbf{L}$ is a finite set of *links* (or *edges*) between nodes.

Nodes: A *node* represents a random variable X, whose frame (domain or universe, that is, the set of all possible states of X) is denoted by Θ_X. Most of the time, a node and its corresponding variable are confused. In a graph, a node is represented by a circle. A sequence of nodes is called a *path*. A subset of nodes is said to be complete if there are links between all nodes of this subset. A subset that is maximal with this property is called a *clique* [20].

Links: A *link* (or *edge*) is an unordered pair of elements of a set of nodes $\mathbf{U}$. In a graph, a link is represented by an arrow or an arc (depending on whether the graph is direct or not). A direct link is often noted $A \to B$, for $A, B \in \mathbf{U}$. Direct links represent causal relationships.

Hypergraph: A *hypergraph* is a graph whose links connect two or more nodes. Its links are then called *hyperlinks* or *hyperedges*.

Different kinds of graphs: A graph can be any of the following:

A graph is *direct* (*undirected*) if its links have a direction (no direction). For example, A implies B, and B does not imply A (direct), or A implies B, and B implies A (undirected).

A graph is *simply* (*multiple*) connected if each node has only one (more than one) incoming.

A graph is *cyclic* (*acyclic*) if it contains (does not contain) a cycle. This characteristic is valid only for direct graphs.

Among these combinations, we note some particularly interesting and fully used graphs: direct acyclic graphs (DAGs), causal trees, and diagnosis trees.

Trees: A *tree* is a graph in which every node except the root has only one incoming link.

- A *join tree* is a tree whose nodes are subsets of variables such that if a variable is in two distinct nodes, then it is in every node on the path between the two nodes [27].
- A *binary join tree* is a join tree such that no node has more than three neighbors.

- A *junction tree* is a join tree whose nodes are the cliques of the triangular moral graph.
- A *triangulated graph* is a graph containing no cycles of length 4 or more, without a *chord*.
- A *moral graph* is built by making links between unconnected parents ("marrying" parents) of a common child and dropping its original directions.

Direct acyclic graphs: A DAG is a graph that is direct and contains no cycles. It can, however, be simply or multiple connected. A DAG is also be called a causal network or causal graph.

Parents-children: In a direct graph, nodes can be either parents or children, depending on the direction of the link. For example, if $A \to B$, A is the parent of B, and thus B is the child (sometimes called the son or daughter) of A. A node can have more than two parents! If every node in the graph has only one parent, then the graph is a tree.

Causal trees: A *causal tree* (or a direct tree) is a simply connected DAG. It is a tree whose links are direct, whose nodes each have only one incoming link, and where no cycles exist.

Local computation: The main advantage of network-based (or graph-based) approaches is the local-computational technique, which allows computing marginals without explicitly computing the joint.

9.8 Bayesian Networks

Bayesian networks, also called Bayesian belief networks (or simply belief networks) or causal probability networks, became popular at beginning of the 1990s, within the community working on the artificial intelligence probabilistic. This model of reasoning was first introduced by Pearl [19]. A Bayesian network is a graphical representation of the relations between the variable set used to represent the knowledge of a given domain. Mathematically, this structure type is called a DAG in which nodes represent the variables of interest, and links connecting nodes represent causal influences between variables (i.e., their conditional dependencies). It consists then of two parts: a qualitative part (the graph representing the dependencies between variables) and a quantitative part (the conditional probabilities associated with each variable). For tutorials or introductions on belief networks, see [21, 31–33].

A Bayesian network is a 3-tuple $\mathcal{G}_{BN} = (\mathbf{U}, \mathbf{L}, \mathbf{P})$, where

- $\mathbf{U} = \{X_1, \ldots, X_n$ is the set of (random) variables of interest, often called the universe. For $X \in \mathbf{U}$, Θ_X is the *frame (of discernment)* of X (i.e., the set of all possible values for X).
- $\mathbf{L}$ is the set of links (arcs) over $\mathbf{U}$ representing the conditional dependencies between variables, such that $(\mathbf{U}, \mathbf{L})$ is a DAG.
- $\mathbf{P}$ is the set of the corresponding conditional probabilities.

Each variable X of $\mathbf{U}$ has a finite set of mutually exclusive states (Θ_X) and a conditional probabilities table.

Product space: Given a nonempty set of variables $U \subseteq \mathbf{U}$, Θ_U denotes the Cartesian product of all the frames Θ_X such that $X \in U$, and is called the *frame for U*:

$$\Theta_U = \times \{\Theta_X \mid X \in U\} \tag{9.34}$$

where "×" is the Cartesian product.

Conditional probability table: The *conditional probability table (CPT)* is a fixed matrix quantifying a link between two nodes, $X \rightarrow Y$:

$$\mathbf{P}_{Y|X} = \{P_{Y|X}\}_{(i,j)} = \{P(Y = y_j \mid X = x_i\} \tag{9.35}$$

for $x_i \in \Theta_X$ and $y_j \in \Theta_Y$. Figure 9.1 is an example of a Bayesian network: Node D has nodes B and E as parents, and its CPT is $P(D \mid B, E)$. If X does not have any parent, then the table reduces to prior probabilities $P(X = x)$, $x \in \Theta_X$. This is the case for nodes A and E.

Joint probability: The generalization for n random variables leads to the *joint probability*:

$$P(\mathbf{U}) = P(X_1, \ldots, X_n) = \prod_{X_i \in \mathbf{U}} P(X_i \mid X_{i+1}, \ldots, X_n) \tag{9.36}$$

In a Bayesian network, because of the dependencies (independencies) between variables, (9.36) reduces to

$$P(\mathbf{U}) = \prod_{X_i \in \mathbf{U}} P\left(X_i \mid X_1^1, \ldots, X_k^1\right) \tag{9.37}$$

where $X_1^1, \ldots, X_k^1$ are the parents of X_i.

Marginal probability: The *marginal probability* for some subset U of variables of $\mathbf{U}$, $U \subset \mathbf{U}$

$$P(U) = P(\{X, X \in U\} = \sum_{y \in \Theta_y,\, Y \in \mathbf{U} - U} P(U, Y = y) \tag{9.38}$$

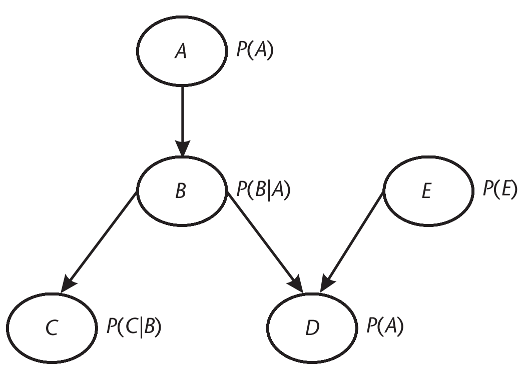


Figure 9.1 A Bayesian network.

where k is an order sum, and $k = |\mathbf{U} - U|$ represents the number of variables in $\mathbf{U} - U$.

Evidence: According to Pearl [19], the incoming information can be of two kinds:

1. *Specific evidence,* which includes direct observations that validate with certainty the values of some variables in the graph;
2. *Virtual evidence,* which are judgments based on undisclosed observations that affect the belief in some variable in the graph.

Belief: A new evidence implies then the instantiation of the variables corresponding to it, updating the conditional probabilities attached to the corresponding nodes. The resulting belief is then

$$Bel(a) \triangleq P(A = a \mid E = e) \tag{9.39}$$

$Bel(a)$ is the belief then accorded to proposition $A = a$, and E is the value combination of all instantiated variables.

9.8.1 Calculus and Reasoning

In a graph, a link represents the dependency between two variables. A missing link between two variables means that they are independent. Because a Bayesian network is based on probability theory, this kind of independence refers to statistical independence.

Inference: The objective is to know the marginal distributions (or marginals, for short) of some variables of interest, which will be computed from the conditional tables and prior probabilities, using Bayes's rule. Given H, a set of hypotheses, and E, a set of evidences (observations), we can determine $h \in H$ given $e \in E$:

$$P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)} \tag{9.40}$$

Local computation: If there are a lot of variables in the network, the computation of the joint probability becomes intractable. However, local computation is possible, enabling the computing of marginals without explicitly computing the joint probability.

Bayesian inference algorithms: Algorithms based on local computations have been proposed [20, 22]. We note two kinds of inference algorithms in Bayesian networks:

1. Exact algorithms;
2. Approximate algorithms.

9.8.2 Dealing with Uncertainty and Knowledge

As in all graph-based approaches, the knowledge is represented by the Bayesian network itself:

1. The graph represents the known dependencies between variables.
2. The CPTs and prior probabilities of variables must be known to initialize the network and compute the inferences.
3. An observation serves as an evidence to propagate the information into the network and update the marginal probabilities.

Because the theoretical framework of a Bayesian network is Bayesian probability theory, uncertainty is represented through probability measures.

9.8.3 Final Remarks

Here are the final remarks for this section:

1. The local-computation technique provides a solution to the problem of computational complexity involved in joint probability distributions.
2. Main applications of Bayesian networks concern medical diagnosis, as well as economics, genetics, statistics, and so forth. More recently, Bayesian networks have been applied to target identification [34, 35].

9.9 Valuation-Based Systems

A valuation-based system is a general framework for the graphical representation of systems reasoning under uncertainty. It enables the processing of uncertainty described by different formalisms, including the theory of probabilities, the theory of belief functions, the theory of possibilities, and so forth. This system was developed by Shenoy and Shafer [28] as a generalization of Bayesian networks and other hierarchical evidence-based algorithms. It involves exact computation of marginals using local computation.

A VBS consists of a 3-tuple $\{\mathbf{U}, \mathbf{\Theta}, \mathbf{V}\}$, where

- $\mathbf{U} = \{X_1, \ldots, X_n\}$ is the set of (random) variables of interest, often called the universe.
- $\mathbf{\Theta} = \{\Theta_X\}_{X \in \mathbf{U}}$ is a set of frames, where Θ_X is the frame (of discernment) of X (i.e., the set of all possible values for X).
- $\mathbf{V} = \{V_i\}_{i=1,\ldots,m}$ is a set of m valuations associated with each of the m links between variables.

A valuation network is an undirected graph built like any other graph, such as a Bayesian network. The nodes (circles) represent the variables of the problem, and the links denote influences between variables. The valuations are represented by diamond-shaped nodes. Figure 9.2 gives an example of a VN issued from the transformation of the Bayesian network of Figure 9.1.

Variables: Let us call U the set of all the variables (circular nodes) of the problem, and let X be one of these variables. We call the set Θ_X of all the possible values for X the frame for X.

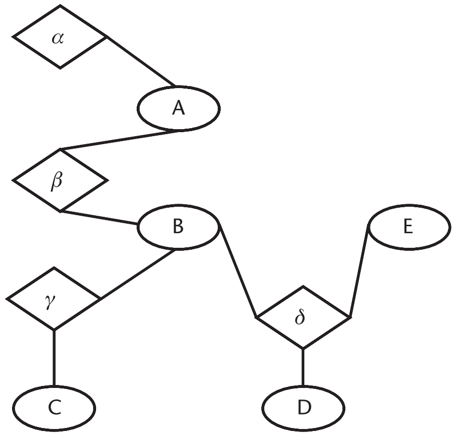


Figure 9.2 Example of a VN.

Configurations: Let U be a nonempty set of variables $U \subseteq \mathbf{U}$, and let Θ_U denotes its frame. The elements of Θ_U are called *configurations* of U. For example, if U is a set of two binary variables X_1 and X_2, then the configurations for U are

$$\Theta_U = \{(x_1, x_2), (x_1, \overline{x_2}), (\overline{x_1}, x_2), (\overline{x_1}, \overline{x_2})\} \tag{9.41}$$

Valuations: Given a subset of variables $U \subseteq \mathbf{U}$, a *valuation* for U represents some knowledge about the variables in U.

$$\mathbf{V} = \cup \{\mathbf{V}_U \mid U \subseteq \mathbf{U}\} \tag{9.42}$$

If $X \in U$, then we say that $\mathbf{V}_U$ bears on X. We denote two kinds of valuations: those defined on subsets of variables $U \subseteq \mathbf{U}$ and those defined for single variables $u \in \mathbf{U}$. Valuations are primitives in the abstract framework of VBS, and that's why they require no definition. Valuations can represent different formalisms, such as probability theories, Dempster-Shafer theory, possibility theory, and so forth. However, besides any interpretation, valuations can simply be seen as objects that can be marginalized and combined.

Joint valuation: The combination of all the valuations is called joint valuation, and the objective is then to compute the marginals of the joint valuation for all the variables of interest. Shenoy and Shafer have presented an algorithm for doing so without explicitly computing the joint valuation.

9.9.1 Calculus and Reasoning (Aggregation/Fusion)

In a graph, a link represents the dependency between two variables. A missing link between two variables means that they are independent. Because a Bayesian network is based on probability theory, this kind of independence refers to statistical independence.

In a VBS, as in a Bayesian network, the objective is to compute the marginals of the joint valuation for some variables of interest. Shenoy and Shafer have presented an algorithm for doing so, without explicitly computing the joint valuation (i.e., using local computations).

Combination: Let V_1 and V_2 be valuations for two nonempty subsets of **U**, U_1 and U_2. A *combination* is a mapping ⊗: $\mathbf{V} \times \mathbf{V} \to \mathbf{V}$, such that $V_1 \oplus V_2$ is a valuation for $U_1 \cup U_2$. The operation corresponds to the aggregation of knowledge. For example, the basic probability assignment (BPA) combination uses Dempster's rule of combination [26]. The combination of all the valuations is called *joint valuation*:

$$\otimes \{V \in \mathbf{V}\} \tag{9.43}$$

Marginalization: Let U_1 and U_2 be two subsets of **U** on which valuations are defined, such that $U_1 \subseteq U_2$. Thus, a marginalization to U_1 is a mapping

$$\downarrow U_1\colon \cup \left\{V_{U_1} \mid U_1 \subseteq U_2\right\} \to V_{U_1} \tag{9.44}$$

such that $V^{\downarrow U_1}$ is a valuation for U_1 if V_{U_2} is a valuation for U_2. Marginalization corresponds to a coarsening of a knowledge by deleting variables. If V_U is a valuation for U (representing then some knowledge about the variables in U), and if $X \in U$ is a variable of U, then the marginalization from U to $U - \{X\}$ (U pruned from the single variable X), noted as $V^{(U\{X\})}$, represents the knowledge about remaining variables in $U - \{X\}$ implied by V_U, disregarding the variable X. In Dempster-Shafer theory, this corresponds to minimization over Θ_X.

Axioms for combination and marginalization: The operations of combination and marginalization must satisfy three axioms to ensure that the algorithm gives correct answers:

1. Order of deletion does not matter.
2. Combination is commutative and associative.
3. Marginalization is distributive over combination.

Axiom 3 makes local computation possible because it states that it is not necessary to compute $\oplus\{V \mid V \in \mathbf{V}\}$.

Local computation: If a network has N variables with only two states in their respective frames, there will be 2^N configurations for the entire set of variables. Hence, joint valuation with a high number of variables is not easy to compute. Local computation is then a solution to avoid explicit computation of the joint valuation, while allowing the computation of the desired marginals. The algorithm has been proposed by Shenoy [26]. Shenoy shows first that the algorithm must satisfy the three above axioms. The basic idea of this fusion algorithm is to delete successively all variables but X, the variable for which we want to compute the marginal. Details can be found in [26].

9.9.2 Dealing with Uncertainty and Knowledge

In a VBS, knowledge is represented by the graph itself, with its nodes (variables of interest and their possible values) and its links corresponding to dependencies between variables. Moreover, the knowledge is represented by the valuations associated with the links. Two kinds of knowledge can be distinguished:

1. Generic knowledge is represented by valuations bearing on sets of variables. In general, this kind of knowledge will not change, being provided by experts (equivalent to a knowledge base in expert systems).
2. Factual knowledge is represented by valuations bearing on single variables. This kind of knowledge is subject to changes according to the problem (equivalent to a database in the expert system).

Because VBS is a general framework allowing the use of different formalisms dealing with uncertainty, uncertainty is thus represented and managed following the corresponding theories. Valuations thus become the carriers of uncertainty quantifications and representations:

1. In probability theory, valuations are called *probability potentials*.[1]
2. In the Dempster-Shafer theory, valuations are called *BPA potentials*.[2]
3. In possibility theory, valuations are called *possibility potentials*.

According to Shenoy, using VBS for decision problems is a more efficient method than using decision trees and influence diagrams.

References

[1] Reichenbach, H., *The Theory of Probability*, Berkeley, CA: University of California Press, 1949.

[2] Carnap, R., *Logical Foundations of Probability*, Chicago, IL: University of Chicago Press, 1950.

[3] Nilsson, N., "Probabilistic Logic," *Artificial Intelligence*, Vol. 28, No. 1, 1986, pp. 71–87.

[4] Halpern, J. Y., "An Analysis of First-Order Logics of Probability," *Artificial Intelligence*, Vol. 46, 1990, pp. 311–350.

[5] Fagin, R., J. P. Halpern, and N. Megiddo, "A Logic for Reasoning About Probability," *Information and Computation,* Vol. 87, Nos. 1–2, 1990, pp. 78–128.

[6] Poole, D., "Learning, Bayesian Probability, Graphical Models, and Abduction," in P. Flach and A. Kakas, (eds.), *Abduction and Induction: Essays on Their Relation and Integration*, Norwell, MA: Kluwer Academic Publishers, 1998.

[7] Lakoff, G., "Hedges: A Study in Meaning Criteria and the Logic of Fuzzy Concepts," *Journal of Philosophical Logic*, Vol. 2, 1973, pp. 458–508.

[8] Elkan, C., "The Paradoxical Success of Fuzzy Logic," *IEEE Expert—Intelligent Systems and Their Applications*, Vol. 9, No. 4, 1994, pp. 3–8.

[9] Klir, G., and B. Yuan, "Elkan Goes Wrong—Again," *IEEE Expert—Intelligent Systems and Their Applications*, Vol. 9, No. 4, 1994, pp. 25–26.

[10] Klir, G. J., and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Upper Saddle River, NJ: Prentice-Hall, 1995, p. 574.

1. Probability potentials are unnormalized probability distributions.
2. BPA potentials are unnormalized probability distributions.

[11] Dubois, D., and H. Prade, "An Introduction to Possibilistic and Fuzzy Logics," in P. Smets, et al., (eds.), *Non-Standard Logics for Automated Reasoning*, New York: Academic Press, 1988, pp. 1–40.

[12] Bundy, A., "Incidence Calculus: A Mechanism for Probabilistic Reasoning," *Journal of Automated Reasoning*, Vol. 1, 1985, pp. 263–283.

[13] Bundy, A., "Incidence Calculus," in S. Shapiro, (ed.), *The Encyclopedia of AI*, New York: John Wiley & Sons, 1992, pp. 663–668.

[14] Liu, W., and A. Bundy, "On the Relations Between Incidence Calculus and ATMS," *Proceedings of the 2nd European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU93)*, Granada, Spain: Springer, 1993.

[15] Liu, W., and A. Bundy, "Constructing Probabilistic ATMS Using Extended Incidence Calculus," *International Journal of Approximate Reasoning*, Vol. 11, 1994, pp. 121–158.

[16] Liu, W., and A. Bundy, *The Combination of Different Pieces of Evidence Using Incidence Calculus*, Department of Artificial Intelligence, University of Edinburgh, 1992.

[17] Correa da Silva, F., and A. Bundy, "On Some Equivalence Relations Between Incidence Calculus and Dempster-Shafer's Theory of Evidence," *Proceedings of the 6th Workshop of Uncertainty in Artificial Intelligence*, 1990.

[18] Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, San Mateo, CA: Morgan Kaufmann Publishers, 1988.

[19] Pearl, J., "Fusion, Propagation and Structuring in Belief Networks," *Artificial Intelligence*, Vol. 29, 1986, pp. 241–288.

[20] Lauritzen, S. L., and D. J. Speigelhalter, "Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems (with discussion)," *Journal of Royal Statistical Society*, Series B, Vol. 50, No. 2, 1988, pp. 157–224.

[21] Jensen, F. V., *An Introduction to Bayesian Network*, New York: Springer, 1996.

[22] Jensen, F. V., K. G. Olesen, and S. K. Andersen, "An Algebra of Bayesian Belief Universes for Knowledge-Based Systems," *Networks*, Vol. 20, No. 5, 1990, pp. 637–659.

[23] Gordon, J., and E. H. Shortliffe, "A Method for Managing Evidential Reasoning in Hierarchical Hypothesis Spaces," *Artificial Intelligence*, Vol. 26, 1985, pp. 323–358.

[24] Shafer, G., and R. Logan, "Implementing Dempster's Rule for Hierarchical Evidence," *Artificial Intelligence*, Vol. 33, 1987, pp. 271–298.

[25] Shenoy, P. P., and G. Shafer, "Axioms for Probability and Belief Functions Propagation," in R. D. Shachter, J. F. Lemmer and L. N. Kanal, (eds.), *Uncertainty in Artificial Intelligence*, Amsterdam: Elsevier, 1990, pp. 169–198.

[26] Shenoy, P. P., "Valuation-Based Systems: A Framework for Managing Uncertainty in Expert Systems," in L. A. Zadeh and J. Kacprzyk, (eds.), *Fuzzy Logic for the Management of Uncertainty*, New York: John Wiley & Sons, 1992, pp. 83–104.

[27] Lepar, V., and P. P. Shenoy, "A Comparison of Lauritzen-Spiegelhalter, Hugin and Shenoy-Shafer Architectures for Computing Marginals of Probability Distributions," in G. S. Moral, (ed.), *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)*, 1998, pp. 328–337.

[28] Shenoy, P. P., and Shafer, G., "Propagating Belief Functions with Local Computations" *IEEE Expert*, Vol. 1, No. 3, 1986, pp. 43–52.

[29] Shenoy, P. P., "Binary Join Trees for Computing Marginals in the Shenoy-Shafer Architecture," *International Journal of Approximate Reasoning*, Vol. 17, 1997, pp. 239–263.

[30] Pearl, J., *Causality: Models, Reasoning, and Inference*, Cambridge, U.K.: Cambridge University Press, 2000, p. 384.

[31] Charniak, E., "Bayesian Networks Without Tears," *AI Magazine*, Vol. 12, No. 4, 1991, pp. 50–63.

[32] Pearl, J., "Bayesian Networks," in R. A. Wilson and F. C. Keil, (eds.), *MIT Encyclopedia of Cognitive Sciences*, Cambridge, MA: MIT Press, 1999.

[33] Huang, Y. S., and C. Y. Suen, "A Method of Combining Multiple Experts for the Recognition of Unconstrained Handwritten Numerals," *IEEE Transactions on Pattern Recognition and Artificial Intelligence*, Vol. 17, No. 1, 1995, pp. 90–94.

[34] Chang, K.-C., and R. Fung, "Target Identification with Bayesian Networks in a Multiple Hypothesis Tracking System," *SPIE*, Vol. 36, 1997, pp. 684–691.

[35] Ménard, É., et al., "Application of Valuation-Based Systems to Object Recognition Problem," *ISIF Proceedings of 4th Annual Conference on Information Fusion*, Montreal, Quebec, Vol. 2, August 7–10, 2001, pp. FrC2-17–24.

CHAPTER 10

Computational Aspects of Information Fusion

Steve Wark and Jean Roy

10.1 Introduction

As has been discussed, data and information fusion plays a critical role in achieving situational awareness for future command-and-control systems. Information fusion (IF) allows the commander to cope with the complexity and tempo of operations in the modern dynamic battle space and serves an important role in asymmetric conflict. Data and information fusion draws together concepts from a wide range of fields: psychology, human factors, knowledge representation, artificial intelligence, mathematical logic, and signal processing. The complex nature of IF environments, IF sources, and IF processes is carried over into the challenging requirements for IF systems and the computational systems needed to implement them.

This and the following chapters review the computational issues surrounding the development of IF systems, with a focus on the higher levels of information fusion and issues relating to an integrated IF environment rather than the specialized issues related to multisensor fusion and control. Some of the considerations for an integrated IF environment are:

- What are the key characteristics of the IF domain and the performance requirements that they impose on IF systems?
- What are the key elements of computational infrastructure relevant to the design and performance of IF systems? What are suitable system architectures, computer networks, middleware, information sources, and human-computer interfaces?
- What key concepts in knowledge-based and artificial intelligence systems impact upon higher-level fusion processes?
- What technologies are appropriate for engineering IF systems? Are reactive systems like subsumption architectures, neural networks, rule-based systems, logically based systems, or case-based systems become?
- What software architectures are appropriate for IF systems? Are systems such as blackboard architectures and multiagent systems suitable?

Moving beyond national-defense systems, a coalition information system incorporating information sources and systems from diverse agencies, organizations, sectors, or nations, will almost certainly require a heterogeneous distributed architecture. What middleware is appropriate for this, especially given that coalitions and organizational alliances come and go and information dissemination between participants often needs to be restricted. These issues are explored in the remaining chapters of this book.

10.2 Information-Fusion Domain Characteristics

IF technology can be applied to information-processing tasks in any complex, dynamic domain where information is derived from a number of sources; large volumes of information need to be processed; tight time constraints apply; the dimensionality of the information is high; or adaptive information collection is required. In these cases, the goal of IF technology is to reduce the complexity to a level manageable by a human analyst or operator. This can be achieved by, for example, correlating information from multiple sources, extracting key information from "noise" or clutter, automating the analysis processes, aggregating and abstracting information to a higher level, or optimizing information collection. IF technologies can be applied to diverse domains, including command and control, intelligence analysis, strategic analysis, counterterrorism, and homeland security.

This section discusses the main characteristics of the IF domain and its consequent performance requirements in order to elicit the key computational requirements for IF systems. The focus of this discussion will be on information fusion for situation awareness and decision-making in the coalition command-and-control domain, but it should be noted that many of the issues and characteristics discussed also apply to other application domains.

Previous chapters have introduced the concept of *situation analysis* as a unifying framework for situation awareness and data fusion in the command-and-control domain, as illustrated in Figure 10.1. The key elements of this model are the characteristics of the environment in which the situation exists, the characteristics of the information sources and processes used for situation acquisition, and the characteristics of the situation-analysis applications (SAAP) that support the establishment and maintenance of situation awareness for the decision-maker.

10.2.1 The Environment

IF technologies are applied to environments that are complex (hence, the need for these technologies) and share five dominant characteristics:

- *Feature rich/dense:* This is true in both the spatial and temporal domains. Large quantities and a variety of features in the environment may include environmental clutter, routine traffic, and unintentional interference, as well as target of interest (TOIs).[1] This leads to a high frequency of events that need to be analyzed or processed.

1. Here the term *TOI* is used to represent physical entities, as well as more abstract features (e.g. patterns of activities, relationships, threats or opportunities) that may be of interest to the analyst or operator.

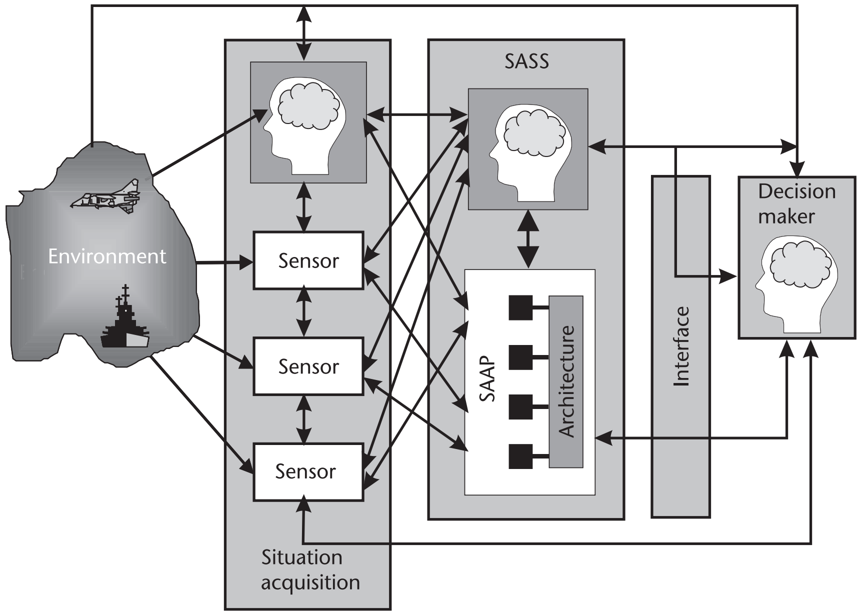


Figure 10.1 Elements of the situation-analysis framework.

- *Highly dynamic:* Environmental clutter may change dramatically, and threats or opportunities may emerge suddenly. The world state can change while the IF system is processing events. Emerging TOIs need to be identified and responded to before they are realized or other operational deadlines are reached. TOIs may seek (or be generated) to minimize the response time available to IF systems via countermeasures or deceptive behavior or both.
- *Unpredictable:* Due to nondeterministic clutter, anomalous traffic, results of own actions, or hostile intent, the environment may not be suitable to formal description, and TOIs may have nondeterministic properties (because of intelligent behavior or responses, such as deception). Behaviors of entities in the environment may not be rational due to conflicting goals or appear not to be rational because of unknown or mistaken goals.
- *Deceptive:* Driven by an organized, intelligent adversary, TOIs may actively attempt (or be generated) to exploit IF vulnerabilities, or may they adapt (or be adapted) to counter IF-system capabilities. This is notionally different from domain unpredictability, which may occur independently of IF-system capabilities. IF systems need to be aware of own vulnerabilities and seek to refine processes dynamically to counter or minimize these vulnerabilities and their consequences.
- *High risk:* Due to the nature of military domains, TOIs may suddenly appear, leaving little time to respond (e.g., as result of deception or countermeasures). Threats posed or the consequences of errors may be disastrous or life threatening.

This environment is important at all echelons: strategic, operational, and tactical.

Table 10.1 lists examples of the drivers for these characteristics that are applicable to the tactical command-and-control environment.

10.2.2 The Information Sources

The information sources and processes required for situation acquisition also contribute to the characteristics of the IF domain. The dominant characteristics are:

- *Multisource:* Accessing multiple similar or dissimilar sources, including sensors and databases. IF systems attempt to maximize utilization of information obtained from multiple sources and must identify and deal with problems such as incestuous fusion and conflicting information from multiple sources.
- *Uncertain:* Because of stochastic detection processes (noise), measurement uncertainty, biases, incomplete or inconsistent information, IF processes must seek to extract TOIs and identify relationships of interest under these conditions.

Table 10.1 Main Characteristics of the IF Environment

Characteristics	*Example in Tactical C2 Domains*
Feature rich/dense	Clutter (e.g., terrain, sea state, weather, wildlife) Countermeasures (e.g., decoys, jamming, chaff) RF interference (e.g., data links, radio, radar emitters) Huge surveillance volume of interest (e.g., search radars, satellites) High density of entities/activity (RF signals, sea/air traffic) Wide variety of entities/activity (e.g., RF emitters, sea vessels) Mixed military/civilian environment (commercial shipping/air lanes)
Highly dynamic	High speed threats (supersonic missiles) Stealthy, low-observable threats (stand-off weapons, fighters) Close proximity of threats (littoral operations, rules of engagement) Pop-up threats (e.g., sea-skimming missiles, cruise missiles) Mobile targets (e.g., scud missiles) Multiple threats (uncoordinated strikes) Simultaneous convergence of threats (coordinated strikes)
Unpredictable	Clutter Mixed military/civilian environment Advanced threats (new technologies) Intelligent, adaptive threats/targets (new doctrines) Maneuvering threats/targets (evasion) Ill-structured problems (no formal description available) Competing or conflicting goals (own and threat systems)
Deceptive	Innovative operational concepts Camouflage (passive) Spoofing (false information/identification) Countermeasures (e.g., false targets) Exploitation of critical vulnerabilities (intelligent threats) Deceptive maneuvering Exploitation of rules of engagement
High risk	Pop-up threats (short reaction times available) Critical vulnerabilities Risk of collateral damage Risk of fratricide Political consequences

- *Delayed:* Collection latencies may be introduced by acquisition, processing, or network delays. Delayed information may delay processing or confuse temporal relationships. IF processes must seek to reconstruct temporal relationships.
- *Dynamic:* Adaptive control of collection and analysis processes is needed to respond to dynamic environments.
- *Resource limited:* Sensors and sources may have resource limitations such as CPU, network, energy, or timing limits. IF systems may need to schedule resource usage to remain within these limits.

Examples of these characteristics are shown in Table 10.2.

10.2.3 The Fusion Process

The situation-analysis/IF process has characteristics that allow it to deal with the environmental and situation-acquisition processes. The main characteristics of the analysis processes required for situation analysis and information fusion are

- *Complex:* The scale, depth, and interdependencies of analysis problems may be beyond human capabilities (given time constraints).
- *Heterogeneous:* The system needs to be able to deal with heterogeneous concepts, data, and TOIs.
- *Robust:* The system needs to be resilient in the face of a wide range of conditions and factors.
- *Intelligent:* The system needs to be adaptable to changing or evolving environments.

Table 10.2 Characteristics of Situation Acquisition in IF Domain

Characteristics	*Examples in Tactical C2 Domains*
Multisource	Shipboard radar/IFF/IRST/ESM Sonar arrays Cooperative Engagement Capability Networkcentric warfare Electronic-warfare databases
Uncertain	Thermal noise (RF/IR) Registration biases (over-the-horizon radars) 2-D versus 3-D radars Terrain masking
Delayed	Search radar (rotation rate = update rate) UGS networks Network delays (out-of-sequence reports) Processing delays
Dynamic	Sensor cuing Multifunction phased array radars (MFAR) Kalman filters
Resource limited	Beam scheduling (MFAR) CPU limits Tactical digital information links (TADIL) bandwidth limits

Table 10.3 presents these characteristics and the factors that contribute to them.

10.2.4 The IF Application Perspective

The performance requirements for IF systems are determined by the characteristics of the IF domain and the operational requirements of the application. The relative

Table 10.3 Main Characteristics of the Information Fusion Process

Characteristics	*Requirements/Contributing Factors*
Complex	Often have ill-structured problem not amenable to formal representation Must deal with large-scale problems with large amounts of information to process Possess large solution space—no clearly optimal solution available Present multiple alternatives—no unique solution path/line of reasoning Must handle asynchronous data and information Computational complexity of algorithms needs to be managed to ensure timely solutions May need to deal with multiple users Often have distributed fusion nodes—distributed problem solving Have distributed databases—legacy and proprietary systems Must handle interoperability with services and coalition partners Must deal with dynamic coalition formation
Heterogeneous	Need to integrate/manipulate a variety of data Need to deal with noncommensurate data and information types Have a mix of diverse qualitative/quantitative information Need to deal with multiple levels of abstraction May need multiple interdependent analyses Need for many independent/semidependent pieces of knowledge to cooperate in forming a solution Systems (and requirements) may be data driven and/or goal driven Open system that incorporates human operators or "fusion nodes"—user-in-the-loop analyses Need to deal with legacy and proprietary systems
Robust	Need guaranteed ("anytime") response Need to perform when have unknown/unpredictable solution path Need to be scalable to allow addition of new nodes, deal with bigger problems Need to evolve to adapt to changing force structures and requirements Need intelligent failure management (graceful degradation) Need to address security requirements
Intelligent	May require hierarchical series of inferences Will need to use multiple types of reasoning (e.g., spatial, temporal) May need to use expert systems/rule-based systems/case-based reasoning/blackboards May need to employ explanation-based reasoning May need to apply inductive or deductive reasoning May require heuristic methods for ill-formed problems May incorporated knowledge based systems May need to use very abstract, symbolic, problem-solving approaches Should cope with many diverse, specialized knowledge representations Should be able to interface with a priori knowledge databases Should be adaptive to deal with dynamic environment/situation acquisition Should learn from experience

importance of these requirements in a particular application domain dictates the design of a suitable IF/situation-analysis architecture.

Table 10.4 lists the relative importance of requirements and the associated criteria relevant to the performance of IF applications, which will guide the development of a suitable IF architecture.

10.2.5 The Life-Cycle Support Perspective

For any operational system, life-cycle support and subsequent costs, whether financial or labor related, are of major importance. The key elements of life-cycle support arise from the need to meet and maintain the performance criteria discussed in the previous section—in particular, heterogeneity, scalability, robustness, and flexibility. Additionally, incremental growth and refinement of the IF system will be needed to meet evolving requirements so ongoing system-development issues are also important.

Table 10.5 lists some requirements relevant to life-cycle support for IF systems.

10.2.6 IF System Design

The development of an IF system needs to be grounded in the application domain to which it applies. From the preceding discussion, however, it is clear that there are some common principles of IF system design that should be applied to IF systems in order to meet the requirements above. The system must do the following:

- Use a component-based software architecture to support scalability, ongoing development, and system evolution;
- Incorporate distributed components, even if this only includes distributed information sources connected to a centralized processing facility;
- Integrate heterogeneous components, including new and legacy systems;
- Be standards based to support integration, ongoing development, and system evolution;
- Incorporate robust failure- and error-handling mechanisms;
- Incorporate mechanisms to manage information security and assurance.

IF systems tailored to particular application domains can then be built on top of this framework, incorporating their particular functional requirements. Using this approach, new applications can be developed rapidly by extending existing systems.

10.3 System Architectures

The architectures needed for IF systems are shaped by the underlying computational infrastructure in which they are implemented. As discussed in previous chapters, IF systems can be modeled using the Joint Directors of Laboratories (JDL) model. In this model [1], the level 0, 1, and 4 processes represent elements of the situation-

Table 10.4 IF Performance Requirements/Criteria

Importance	*Requirements/Criteria*
High	*Timeliness of solution*:* This is of paramount importance in the tactical domain. In other domains, quality of the solution may be more important than timeliness, but in all cases some measure of timeliness appropriate to the domain will apply. *Real-time performance*:* This includes factors such as timeliness and guaranteed performance and may be of high importance to tactical domains. In other domains, timeliness of the solution may be the dominant requirement. *Efficient dynamic scheduling/control mechanisms*:* In the tactical domain this has high importance to ensure the timeliness of the solution in dynamic environments, but it may be less important in other IF domains. *Efficient data/knowledge representation:* This applies both to tactical domains, where efficient computational mechanisms are required to ensure timeliness, and to other IF domains where efficient algorithms are required to manage computational complexity. *Scalability to large-scale problems:* This will always be of high importance as one of the main drivers for IF systems is to manage "information overload" in human analysts. *Capability to deal with multiple interdependent processes:* This is a similar issue to scalability but is particular to managing multiple concurrent collection and analysis processes, as well as the complexity of real-life problem domains (as opposed to simplified models). *Interoperability of multiple heterogeneous technologies:* Very different computational technologies are needed to handle the various stages of the IF/situation-analysis process, so any approach will require the integration of different technologies. *Robustness:* In all domains, failure modes must not be catastrophic and should allow human intervention or provide controlled degradation of performance in overload conditions and with system failure. *Trustworthiness:* Operator confidence in any IF system is crucial if the technology is to be applied effectively. This will often require transparency of system operations to operators so that they can "drill down" into the system to validate its performance against known measures. *Information security:* This is always of high importance, particularly in a coalition C2 environment where conventional security compartmentalization may not be appropriate.
Medium	*Flexible prioritization, redistribution, and control of processing tasks:* Given the capability to deal with multiple interdependent processes, efficient and flexible control of processing tasks is needed to optimize IF system performance in dynamic environments. This is not considered a high-priority requirement here because a human operator may usurp this functionality if no automated mechanism is available. *Incremental refinement of solutions*:* To provide robustness with partial information or under time pressure, a partial solution may be required as quickly as possible. Incremental refinement of the solution then allows further deliberation/processing to improve the quality of the solution when time is opportunistically available. In other domains where time pressure is not so critical, the highest-quality solution may always be desired, and so greater emphasis may be placed on criteria that contribute to an optimal solution. *Integration/manipulation of varied data types/formats/sources:* Related to the interoperability of heterogeneous technologies, IF systems will need to cope with varied data and data sources. Standardization of data formats to achieve the primary goal of interoperability of heterogeneous technologies may resolve this issue. Use of standardized/common ontologies and automated translation mechanisms will provide more scalable and dynamic interoperability of heterogeneous data, which is particularly important in coalition environments. *Secure information distribution policies and management:* In coalition C2 environments, conventional techniques may seriously impact system performance. Metatagging of information and dynamic control of information distribution policies between organizational, functional, and administrative domains through automated policies would be much more appropriate to efficient operation in a coalition IF system.

Table 10.4 (continued)

Importance	*Requirements/Criteria*
Lower	*Efficient management of a priori data and knowledge bases:* IF systems in most domains will generally require access to many diverse, and often large, data/knowledge bases holding legacy and a priori information. Efficient IF system performance will require intelligent management, control, and revision of these data/knowledge bases. *Opportunistic problem-solving ability:* When time is available, the system should make good use of it. This may, for example, allow refinement of partial solutions discussed above or allow the updating of system databases from observed data. *Graceful degradation of system performance:* This is related to IF system robustness, but rather than requiring controlled degradation, which could be as simple as switching to a backup system, graceful degradation of performance with system failure or under overload conditions would provide greater assurance of IF products.

*Particularly in tactical C2 domains.

acquisition process of the situation-analysis model described earlier involving multisensor integration and adaptive sensor control (in the tactical C2 domain). There are many specialized technologies relating to these processes, but this discussion will be confined largely to general computational issues for information fusion—those issues specific to levels 2 and 3 of information fusion, and how they integrate with levels 0 and 1 (via level 4)—rather than attempt an in-depth analysis of computational issues associated with levels 0 and 1.

In general (although not universally), at levels 0 and 1, and to some extent level 4, fusion in the JDL model primarily involves numerical processing to extract estimates of object characteristics and to apply parametric control signals. In contrast, at levels 2 and 3, fusion primarily involves symbolic processing to extract situation and impact representations, although some work has applied numerical estimation techniques to cluster objects using a priori templates, which is often referred to as a "lower" level 2 function. The difference in the types of processing required for the different fusion levels in the JDL model leads to significant differences in computational requirements throughout the system and across the many different functional components.

As discussed above, IF systems may consist of many functional components, and the architecture used to integrate these components depends on a number of factors in the application domain. There are two fundamental models that can be applied to IF system architecture—centralized and distributed systems, as illustrated in Figure 10.2.

10.3.1 Centralized Systems

Centralized systems are based around a central node that handles all levels of processing in the IF system. A centralized system can potentially provide optimal system performance since all of the information relevant to all of the processing tasks is available to it. However, in practice, centralized systems suffer from several limitations, including the following:

Table 10.5 IF Life-Cycle Support Requirements and Criteria

Importance	*Requirements/Criteria*
High	*Modular, component based architecture:* This is to allow incremental development, refinement, and growth of an IF system. This is particularly important for IF systems as the technology is still in the early stages of development and often requires application- and domain-specific processing of some sort. *Flexibility:* The IF system should not be hardwired to a particular domain or situation but should allow customization to tune applications to particular domains, or even to particular situations in a domain, by the user or human subject matter expert. This allows the IF system to be adapted to changing or similar domains. *Interoperability with heterogeneous, including legacy, applications:* The IF system should make use of existing tools when appropriate and allow integration with third-party applications, particularly in coalition C2 environments. *Standardized interfaces:* This an important requirement to support coalition interoperability and through-life development. Provided suitable interfaces have been developed, many system-development requirements may be considered less important. *Embedded performance-evaluation tools:* These will allow the user to monitor, evaluate, validate, analyze, and diagnose the behavior, dynamics, and performance of the IF system. *Predictability:* This is related to performance validation; the IF systems should behave predictably in well-defined situations. *Documentation:* Comprehensive documentation of the IF system will be required to support the user and allow incremental refinement. *Embedded training/simulation tools:* These will leverage off of embedded performance-evaluation tools *Robustness:* The IF systems should be robust to component failure, which may arise from hardware failure, software defects, or incompatibility of new or upgraded components. *Comprehensive licensing and support agreements:* These will provide vendor support to IF components and infrastructure, including validation of component upgrades before integration with the IF system.
Medium	*Embedded self-diagnosis tools:* These will allow the IF system to monitor its own performance and alert the operator to any potential problems. *Ongoing ability to draw on a commercial skill base:* This will simplify operator training. *Availability of source code:* The will allow refinement of individual components and includes access to translations of any foreign-language code, particularly in coalition operations *CASE/LCS tools for source code:* These will allow effective use of source code when available, including tools such as compilers, debuggers, run-time profilers, and configuration-management and bug-tracking tools. *Code reuse/modularity:* Source code should be designed to allow for easy extension or modification (e.g., object-oriented code) *Cross-platform support:* Development tools should support the ability to move components to different operating systems. In the case of source languages and the like, appropriate tools should be available on other platforms, and components should not be developed in a proprietary language. *Efficient communication protocol(s):* This aspect may be at odds with the requirement for standardized interfaces (where efficiency may be sacrificed for greater expressibility) but may be of higher importance in domains with bandwidth or timeliness limitations.
Lower	*Costs:* Must be affordable in terms of maintenance and support. *Platform independence:* Components should be able to run on any available platform. *Archival support:* This should include datasets, data-base updates, and so forth.

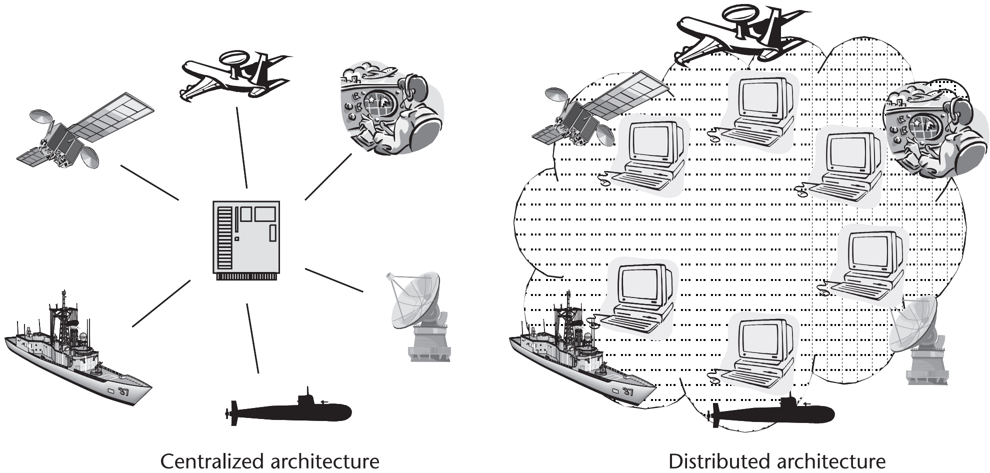


Figure 10.2 Centralized and distributed IF architectures.

- Developing a centralized system for large complex problems, such as an IF system, is extremely difficult [2].
- Centralized systems represent potential bottlenecks [3]. A single system that must perform all tasks can easily be overloaded and may slow operations dramatically. Hence, centralized systems may not operate well in dynamic domains.
- Centralized systems have a single point of failure [2, 3].
- Centralized systems are critically dependent on the communication infrastructure between the information sources or sensors and central processing system. Network traffic can also form a bottleneck in the system. Centralized systems are not suited to domains that have geographically distributed information sources or sensors because the communication overhead can be extreme if raw data is sent to the central system for processing [3].
- Centralized systems are typically not readily scalable; it is often difficult to add, remove, or alter components in the system [2].

The advantages of centralized systems include the following:

- Centralized systems are well suited to domains where the system tasks are highly dependent on each other as the processing tasks cannot easily be decomposed into subtasks across multiple systems.
- Centralized systems provide a single point of maintenance for large, complex systems, allowing their components to be readily maintained, repaired, and updated. They also provide a single release authority for information, simplifying control of proprietary and classified information.

IF systems can use a centralized architecture when the number of sources is tightly constrained, and all sources and processing must reside on a single platform, such as multisensor fusion systems for operationally deployed platforms.

10.3.2 Distributed Systems

Distributed systems consist of a collection of individual systems with some level of processing and control devolved to each component. The IF system is the collection of these components, connected by a network of some sort. Typically, distributed systems have multiple network-connection paths between the component systems.

A distributed system can potentially provide locally optimal performance for its component processing subtasks but may not be able to maintain an up-to-date global view of the IF system. Communication lags between components mean that, in a dynamic environment, no component can know the current state of any other component with certainty. This can lead to inconsistencies between estimates of component states, which potentially prevent global optimality. There are a number of models for distributed system architectures that mitigate this problem by imposing organizational and management systems that seek to maintain a global view of the IF system and coordinate activities between the components.

The advantages of distributed systems include the following:

- Development of large, complex systems is simplified by decomposing them into smaller, distributed tasks.
- Distributed systems allow concurrent processing of multiple elements of a task, eliminating the processing bottleneck of centralized systems.
- Distributed systems do not necessarily have a single point of failure. This depends on the model used to manage the "global" view of the IF system.
- By processing data from information sources or sensors locally, distributed systems can greatly reduce the amount of data that needs to be communicated between components. This reduces the load on the communication network, removing potential network bottlenecks and allowing geographically distributed information sources or sensors to process and transmit data efficiently.
- Distributed systems are readily scalable.

Issues with distributed systems include the following:

- Distributed systems are not suited to domains where the tasks are highly dependent on each other as it becomes difficult to decompose them into smaller, distributed tasks.
- Distributed systems that duplicate a priori information or databases can make it difficult to maintain, repair, upgrade, and control this information. As the distributed model allows varying degrees of component distribution, this can be managed by ensuring that such information is encapsulated in system components.

The distributed system architecture is well suited to IF systems that incorporate many heterogeneous sources and processing components—such as a system that spans all levels of the JDL model—where decomposition into subtasks (for each level or aspect of processing) is easily achieved.

10.4 Computer Systems

Steady advances in fabrication technologies, leading to increases in processing speed, memory, and nonvolatile storage, have given rise to an increasing trend in computational power. This is generally referred to as Moore's Law [4, 5], and the typical rate cited is that computer power doubles every 18 months. This trend is expected to increase until at least 2016 [6], when physical limits on the transistor density of silicon devices are expected to be reached.

10.4.1 Processing Speed

IF processes that use fast heuristic solutions will benefit greatly from predicted increases in processing speed, allowing more to be done when time is limited. For these types of systems, tasks that currently may not be feasible because of time constraints may become feasible in the near future. On the other hand, while IF tasks involving symbolic reasoning (levels 2 and 3) can benefit from increased processing speed, the computational complexity of the problem is often the limiting constraint, and an N-P (nondeterministic polynomial-time) hard problem with exponentially increasing compute time will not necessarily benefit significantly from speed increases.

10.4.2 Memory and Storage

In the past, limits on available nonvolatile storage on computers (e.g., hard disk capacity) have meant that massive data and knowledge bases, such as geospatial and imagery databases, were necessarily kept in large central repositories. This simplifies maintenance and management of the information in the repository, but access may be limited by network bandwidth and latencies. Centralized processing of these datasets reduces network loading but suffers from the limitations discussed above.

Increases in nonvolatile storage capacities have now made it feasible to store massive amounts of information locally, allowing decentralized processing of this information, which may include databases or a priori knowledge and rules for how to process the data. In both cases, decentralization allows faster processing and visualization "on demand" of large datasets but also makes it more difficult to update and maintain these distributed data and knowledge bases. For relatively static information, such as geospatial databases, it may not be necessary to keep the databases updated in real time. In these cases, it may be sufficient to disseminate updates to the databases, sourced from a central repository, as lower-priority background processes or during system downtime.

In other cases, where information is rapidly changing or database management is of paramount importance, a central repository of information remains necessary and Web-portal approaches may be needed to reduce the processing footprint on the local machine. In these cases, a simple Web interface could be used to access information stored and processed in a central repository.

Volatile computer memory (e.g., RAM) can be a limiting factor in computer algorithms designed for symbolic reasoning, such as automated theorem provers

[7], which may exhaustively explore the alternatives to a problem. These systems need to store a combinatorially large number of alternatives—either in volatile or nonvolatile memory. Access to volatile memory is generally much faster than it is for nonvolatile memory, so the compute time for these systems may benefit greatly from increases in available memory.

10.4.3 Operating Systems

In a tactical C2 environment, multisensor integration (levels 0 and 1) and control (level 4) often require proprietary, real-time operating systems that are designed specifically for the particular sensor platforms deployed. In these systems, the primary considerations are guaranteed behavior, system stability, and processing speed for specialized applications. Other system requirements, such as general system capabilities and system scalability, are less important in this context and are often sacrificed to meet these considerations.

In other, less specialized IF systems, more generally capable, but perhaps less stable and slower, operating systems are needed. Coping strategies such as human intervention (e.g., restarting the application, rebooting the computer) can mitigate losses in stability and guaranteed performance. Commercial off-the-shelf (COTS) development tools, applications, and middleware available on common commercial and open-source operating systems provide an important base on which to build and extend IF functionality. The use of common operating systems also provides a greater pool of experienced developers and users.

Specialized, niche operating systems may also be appropriate when an IF application is hardware dependent, such as some visualization systems. In these cases, it is desirable to ensure that appropriate software-development tools are available to allow expansion and refinement of the IF application. Interpreted programming languages, such as Java, that are available on a wide range of systems, provide an alternative to common operating systems, provided that a suitable interpreter (or virtual machine) exists for the platform. Operating system functions may then be supplanted by programming constructs, and IF applications developed in these languages can be run on any capable platform, giving system portability.

10.5 Networks

Computer networks are vital to modern computer systems and the IF system architectures shown in Figure 10.2. The implementation details of the computer network infrastructure are generally handled by middleware layers or the computer operating system and hidden from the IF system user—and to some extent from the developer. Only high-level network performance measures, such as network speed (bandwidth) and network latency, should need to be considered in an IF system. Network speed measures the total amount of data that can be transmitted across the network, while network latency measures the time delays produced by the network in getting data from source to target.

Computer networks connect multiple computers, each with one or more network interface devices, via various network media. Network media provide the communications links between computers on a network and can be electrical wires, optical fiber, or wireless (IR, RF, and microwave) transceivers connected to the network interface devices. The type of network media used places physical limits on the transmission bandwidth, thus on network speed. The network media used can also introduce network latencies, from signal propagation speeds (e.g., for satellite and transcontinental links), data loss due to poor signal-noise ratios (which may require data to be retransmitted), and network throughput limitations imposed by the media speed.

Network protocols define all operations in a network, such as how the hardware accesses the network to send and receive data, how one computer addresses data to be sent to another, what error-correction and handshaking mechanisms are used for data transmission, and how data is structured when it is transmitted. Overheads are associated with network protocols that reduce network speed and introduce latencies into data transmission.

Network routing devices allow data to be sent from one network to another, including networks that may use different architectures, media access models, or data formats. It recognizes whether a message is meant for a machine in a local group or somewhere else and intelligently decides how to forward a message to its destination. This allows efficient transmission of data packets to their destination using a distributed architecture that avoids the problems associated with centralization. However, this does add extra network latencies, as the hardware needs to read the data packets and transform them in some way. Networks can be as simple as local clusters of machines in a local-area network (LAN) or as complex as the Internet. Wide-area networks (WANs) connect physically distributed LANs via dedicated infrastructure or using virtual private networks (VPNs) across the Internet.

10.5.1 The Internet

The Internet is a global network of many smaller networks that have agreed to communicate. It began as a U.S. Advanced Research Project Agency (ARPA) research project to create a decentralized C2 system that would be robust enough to continue functioning even if most of the network were destroyed. The Transmission Control Protocol/Internet Protocol (TCP/IP) grew out of this project to allow data to be broken down into packets and sent by multiple redundant routes to their destination with no centralized control point.

Because of its ubiquity, the Internet is an attractive mechanism for providing network connectivity between geographically distributed components of IF systems or for allowing interoperability between coalition partners. However, data packets on the Internet travel over many redundant paths, can be routed through many network nodes, and may travel vast distances before arriving at their destination. This means that network latency can be significant, so any IF system using the Internet must have mechanisms in place to deal with this. Also, despite its military origins, one of the biggest challenges to the military use (including information fusion) of the Internet is assurance of information security since data packets are routed through many uncontrolled computers before arriving at their destination and it is not possible to track the flow of data.

10.5.2 Virtual Private Networking

Virtual private networks allow two computers or networks to talk to each other securely over an insecure transport media, such as the Internet (see Figure 10.3). To do this, VPNs use a computer at each of the points connecting to the transport media, called a point of presence (POP). Each POP encrypts a data packet and encapsulates it in a new data packet to be sent over the transport media using a technique called tunneling. The new packet is addressed to the appropriate POP on the remote network and sent via the transport media. At arrival, the data packet is authenticated, and the encapsulated packet is extracted and decrypted by the remote POP, then forwarded to the appropriate machine on the remote network.

VPN tunneling and security protocols are used to ensure that this all happens smoothly. VPN security protocols add complexity to VPN as they must ensure authentication, confidentiality, data integrity, and authorized access control. Data encryption, typically requiring a decryption key, provides VPN security. Key management and user authentication are then central to maintaining information security with VPNs.

10.5.3 Theatre Broadcast System

The Australian Theatre Broadcast System (TBS), U.S. Global Broadcast System (GBS), and U.K. Direct Broadcast System (DBS) are military networks designed to provide network connectivity to multiple deployed operational platforms over a wide operating area. In these systems, a primary high-bandwidth, one-way data feed is broadcast from a satellite and simultaneously received by multiple receive stations. A low-bandwidth connection is then used as a back-channel to request information to be added to the broadcast. These systems do not generally provide the full flow-control and error-correction capabilities of full-duplex communication, but simultaneous access by multiple receiver nodes has operational advantages when information such as a common operating picture, or COP, needs to be transmitted to multiple sites.

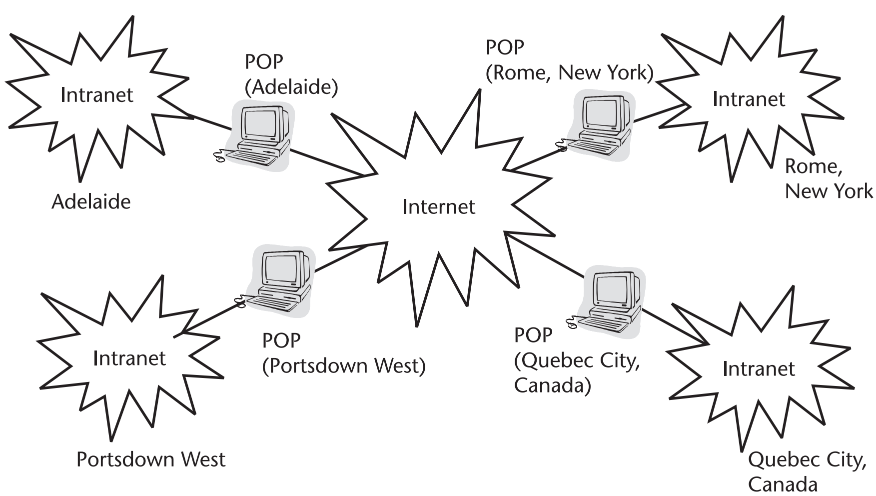


Figure 10.3 VPN between four remote sites via the Internet.

The configuration of the Australian TBS [8] is shown in Figure 10.4. Encrypted information is transmitted to a commercial satellite from the primary injection point and broadcast throughout the region covered by the satellite's footprint at high data rates. Platforms and users within this region can receive and decrypt this broadcast and use low-bandwidth wireless to request information. The requested information is then scheduled into the broadcast stream. This system allows command push of information to users over the broadcast link and allows user pull of information from theater information sources and archives. In other similar systems, multiple dedicated satellites are used to give increased coverage and access.

Deployed platforms may also use lower-bandwidth, end-to-end communication networks, such as tactical data links, to add information to the theater information sources and archives. For IF systems, this makes it possible to use TBS to broadcast shared information (e.g., COP) to distributed IF systems, which fuse this with information from local sensors or sources and transmit the IF products back to the theater C2 system.

10.5.4 Tactical Digital Information Links

Tactical digital information links (TADILs) are standardized military communication links used to transmit tactical data between assets or units. They typically rely on line-of-sight between military platforms or relay stations, although some TADILs, such ask TADIL-A/Link-11, that use high frequencies can allow communication over the horizon. TADILs are used for transmission of digital and, in some cases, voice information. Tactical track-data and platform-status messages can be shared using TADIL systems. TADIL-J/Link-16 offers the greatest promise for coalition interoperability as it is to be deployed in Australia, the United States, and NATO countries. Link-16 is a secure, jam-resistant link that supports voice

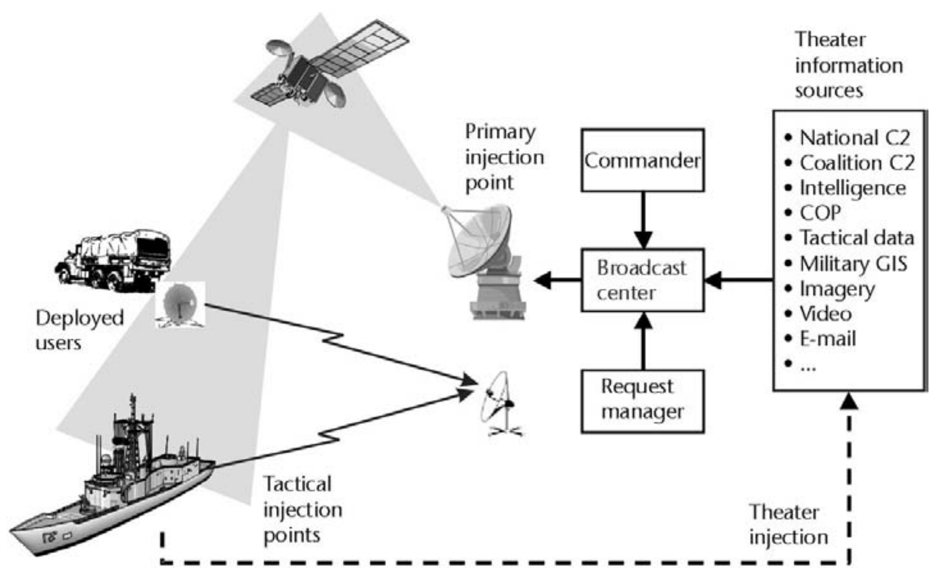


Figure 10.4 Australian TBS.

communications and exchange of air, ground, and maritime surveillance, as well as electronic warfare, intelligence, and platform-status data. Link-16 and other tactical data links use Time Division Multiple Access to provide access to the network—each participating unit is assigned regular time intervals at which to transmit on the network.

In general, TADILs do not provide a high-bandwidth link between distributed platforms for information fusion. Even with notionally high bandwidth of TADILs such as Link-16, access is limited by the time slices assigned to the participating units to meet the operational demands of distribution to many platforms.

Track data on Link-16 and other TADILs is transmitted with a track-quality measure that represents the uncertainty associated with the track—no other information is provided to address issues such as detection uncertainties, sensor biases, and incestuous fusion. Fusion of tracks over TADILs is also limited by messaging specifications—only the track with the best track quality can be transmitted over a TADIL. There is no native format in TADILs to convey higher-level fusion information or products; however, most TADILs employ a “free-text” format that can be used as a channel for this information if a suitable protocol is applied.

10.5.4.1 Cooperative Engagement Capability

The Cooperative Engagement Capability (CEC) is a system of software and hardware that allows U.S. Navy ships and aircraft to share fire-control-quality radar data on air targets and use this data to engage targets cooperatively beyond an individual ship’s sensor envelope. Each participating platform uses identical data-processing hardware, algorithms, and tactical displays and transmits data to other platforms using a dedicated high-bandwidth line-of-sight data-distribution system. This shared sensor data is fused to produce a single integrated air picture, providing improved situation awareness and sensor management.

The CEC system relies on common hardware and algorithms to achieve multisensor fusion for the tactical air-defense environment. In a coalition C2 environment, this may not always be feasible, and greater benefits may be achieved by fusing sensor data from a heterogeneous mix of sensors. Additionally, the CEC system does not address other tactical environments or higher-level information fusion. However, the high-bandwidth communications technology used in CEC and subsequent systems may be applicable in other domains, particularly those driven by the requirements for networkcentric operations.

10.5.5 Network-Centric Warfare

Network-centric warfare [9, 10] is an increasingly important operational concept in U.S. and coalition operations. Networkcentric warfare requires an architecture that enables close coupling between three key elements:

- A *sensor grid* of sensors deployed in the air, sea, ground, space, and cyberspace environments, as well as other information sources required by the task at hand. The sensor grid is intended to provide a high degree of awareness of friendly, neutral, and enemy forces and the environment in the battle

space. This is a transient grid that is formed only for specific tasks, although the components may well be permanently deployed.
- An *information grid* that provides the infrastructure for networkcentric computing and communications, as well as the means to receive, process, transport, store, and protect information. The information grid requires embedded capabilities for information assurance to prevent attack and mitigate failures. The information grid should permit "plug and play" of sensors and shooters—and IF systems. This grid should be permanently available.
- An *engagement (or transaction) grid* that allows the commander to plan and execute operations in a timely and adaptive manner to achieve "lock-out" of enemy operations. The engagement grid tasks assets to create the necessary effect in the battle space and dynamically retasks them as necessary.

The engagement grid is transient and formed for specific tasks.

IF systems both enable and exploit the infrastructure required for networkcentric warfare [11]. The operational architecture of the sensor grid should increase battle-space awareness and synchronize it with military operations. Improvements in operational performance are achieved through a combination of dynamic sensor tasking, data fusion, and effective distribution of information over the information grid. The sensor grid must provide dynamic sensor tasking to allow the commander to have the operational flexibility to synchronize battle-space sensors with the timing and tempo of operations and employ available sensors and resources in multiple modes. To achieve this, the sensor grid must employ information fusion to rapidly enable the commander's situation awareness. By enabling coordination of multiple sensors, the sensor resources can be optimized.

The information grid, as well as the flexibility required for "plug-and-play" operation, will also provide an enabling environment for IF systems. The ongoing development of technologies to support networkcentric warfare should provide a rich resource to feed into the development and operational deployment of IF systems in the future.

10.6 Middleware

Middleware fulfils an essential function in distributed systems by providing a common interface layer between various heterogeneous applications. Middleware also increases portability of application code by hiding proprietary operating system and hardware implementation details from the applications, allowing them to be built independent of a target operating system or hardware. This section discusses some middleware softwares that have been demonstrated in various command-and-control and IF systems and that are likely to be important, if not central, to future coalition IF systems.

10.6.1 CoABS/ISL

The Defense Advanced Research Projects Agency (DARPA) Control of Agent-Based Systems (CoABS) Grid [12, 13] is middleware that enables the integration of distributed, heterogeneous, agent-based, object-based, and legacy systems. It has been commercialized and offered as a COTS product called the Intelligent Services Layer (ISL) by Global InfoTek. It includes a method-based application programming interface (API) that allows agents and services to register, advertise their capabilities, discover other Grid agents based on their capabilities, and send and receive messages.

The mission of the DARPA CoABS program was [12]:

> To develop and demonstrate techniques to safely control, coordinate and manage large systems of autonomous software agents. The Control of Agent-Based Systems (CoABS) program will develop and evaluate a wide variety of alternative agent control and coordination strategies to determine the most effective strategies for achieving the benefits of agent-based systems, while assuring that self-organizing agent systems will maintain acceptable performance and security protections.

There was also a strong focus on the continued use of current object-oriented and procedural software within a new agent-driven framework. The Grid was designed to provide a base level of standardization that enables these heterogeneous systems to communicate. To provide this level of flexibility and generality, it is simple to implement, adapt, and extend and nonprescriptive about higher-level design and implementation issues. The Grid was also designed to be scalable in terms of the number of agents on it, the number of types of agents, and the number of different actions each agent can perform [14]. Experiments on the scalability of the Grid [15] have verified that up to 10,000 agents can exist simultaneously on the Grid without significantly degrading lookup performance.

The Grid is implemented in Java and extends and adds to Sun Microsystems' Jini classes. Grid helper classes can be used to hide the complexity of the underlying Jini classes, or the Jini classes can be used directly if more flexibility or control is required [14]. Grid wrapper classes are used to quickly make existing systems Grid aware. All objects on the Grid implement the service class, which enables other services to remotely call certain of its methods. Lookup services (LUSs) act as Grid directories. There must be at least one LUS running for services to communicate. In order to provide robustness and scalability, multiple LUSs can exist, in which case they share information in order to maintain equivalence.

To use the Grid, a service must register with an LUS. Helper classes simplify this process and make it host transparent by using multicast and unicast techniques to discover LUSs automatically. Services must also advertise their capabilities with an LUS to make them known to all. Capabilities are represented by entry objects, which can be any serializable object with public fields. With this implementation, services can advertise almost any of their features, such as name, languages and ontologies supported, architecture, physical location, goals, methods, and so forth.

The Grid enables heterogeneous systems to communicate and cooperate over a network by building standard interfaces around new and existing code. Much emphasis has been put on simplifying the process of making existing stand-alone systems Grid aware. Higher-level systems are then built on the underlying Grid infrastructure to provide more tools and resources. One extension to the Grid is an implementation of the Knowledgeable Agent-Oriented System (KAoS) [16–19].

Other Grid extensions, developed by various organizations, include a publish/subscribe mechanism, advanced visualization tools, proxy interfaces that enable the use of programming languages other than Java, GUI applications for user input and workflow control, and a system that enables agent mobility.

The effectiveness of the CoABS grid has been demonstrated in a number of coalition command-and-control experiments, including the Coalition Agents Experiments (CoAX) [20–23] demonstrations, the Expeditionary Sensor Grid Enabling Experiments (EEE), and the Fleet Battle Experiments (FBE).

10.6.2 Cougaar

DARPA's Cognitive Agent Architecture (Cougaar) [24] is a software architecture that enables distributed agent-based applications. It was developed for the DARPA Advanced Logistics Project (ALP), and this work has been continued under the DARPA Ultra*Log program [25]. The focus of the ALP and Ultra*Log programs has been to develop techniques for the planning and execution of U.S. military logistics. This is a tremendously complex planning problem, with millions of different objects, using tens of thousands of different business processes, involving thousands of different organizations with their own constraints and user requirements, over a thousand different legacy databases and systems with different data models and protocols. The challenge for Cougaar was to integrate these systems while providing a robust, secure, and scaleable environment.

There are several fundamental principles underlying the Cougaar architecture:

- *Composability:* This entails decomposing complex problems into smaller, maintainable components. The behavior of the aggregate entity is then the emergent behavior of its components.
- *Information hiding and encapsulation:* A given component should have access to all the data it needs and no more.
- *Time-phasing:* All aspects of a Cougaar problem are expected to vary over time, as the requirements change and as a solution is executed. All information about physical entities within Cougaar are time phased, meaning that the timewise history of values associated with that information is maintained.
- *Dynamic replanning and execution monitoring:* A key operating mode of Cougaar is planning and storing that plan in a distributed fashion throughout all the agents. The plan is built on continual dynamic negotiation between agents to attempt to generate a feasible and ultimately optimized cooperative solution. As the world state changes, a solution or plan may become stale, and Cougaar forces replanning to determine how to adjust the plan to compensate for the changes, if possible. Further, Cougaar continually monitors the plan as it is executed and forces replanning as assumptions are modified in real time.
- *Security:* Cougaar is designed to contain significant commercial-grade security mechanisms to ensure that all interagent communications are assured to be snoop- and tamperproof. The infrastructure core software, plug-in modules, and configuration information are designed to be certifiably intact and secure.

- *Robustness:* Applications are designed to be long-lived (running continuously 24/7/365), so many aspects of Cougaar are designed to allow the society to survive the temporary outage of a single agent. The internal state of agents can persist and be stored, to be resumed when the agents are restarted. Agent communities can be defined and changed without impacting other components, allowing dynamic reconfiguration.
- *Scalability:* Encapsulation, data hiding, and fine-grained information management limit the information passed between agents to the bare minimum, supporting massive scalability. Peer-to-peer interagent communication avoids exponential growth of interdependencies and interactions among different agents. The plug-in architecture allows easy integration of large legacy software systems and their representation by agents.

Cougaar is a large-scale workflow engine built upon a component-based, distributed-agent architecture. Agents communicate with one another by a built-in, asynchronous, peer-to-peer, message-passing protocol. Cougaar agents cooperate with one another to solve a particular problem, storing the shared solution in a distributed fashion across the agents. Cougaar agents are composed of related functional modules, which are expected to rework the solution dynamically and continuously as the problem's parameters, constraints, or execution environment change. Agents can access other services, databases, applications, and legacy systems. Plug-in interfaces include SQL, JDBC, XML, Java JNI, screen scraping, and DLL invocations, and many are available as part of the Cougaar release [26].

Cougaar is designed to stay permanently operational once invoked, handling an incoming stream of requirements, continually processing and trying to find better solutions to given problems, and continually reacting to changes in resources, requirements, and events or stimuli. With the introduction of new requirements or stimuli, Cougaar initiates dynamic planning and execution monitoring. Tasks are decomposed and assigned to other processing units, either in the same agent or another agent. Downstream processing results are passed back up to higher-level processing units, which can then aggregate or summarize this information or react by replanning. Each task creates a channel of information flowing through agent communities for requirements passing down and responses flowing back up. At each point, the execution of the planned requirements is monitored, and replanning may occur if a significant discrepancy is detected between the planned and observed operations. Throughout this flow of information across the system, there may be many negotiations among different Cougaar components to work toward optimally satisfying aggregate requirements.

10.6.3 CORBA

The Common Object Request Broker Architecture (CORBA) [27–30] specifies a standardized system that provides interoperability between objects in a heterogeneous, distributed environment and in a way that is transparent to the programmer. Its design is based on the Object Management Group (OMG) [28] Object Management Architecture (OMA). CORBA automates many common network-programming tasks, such as object registration, location, and activation; request demultiplexing; framing and error-handling; parameter marshalling and demarshalling; and operation dispatching. CORBA handles object references within a distributed environment and uses complex mechanisms to provide standardized interfaces to application programs. CORBA has been widely used in distributed environments, including the ARFL Adaptive Sensor Fusion System.

The CORBA object model defines how objects distributed across a heterogeneous environment can be described. The object model defines common object semantics for specifying the externally visible characteristics of encapsulated objects in a standard and implementation-independent way. In this model, objects provide services to *clients* that can only be accessed through well-defined *interfaces* specified in the OMG Interface Definition Language (IDL). A client accesses an object by issuing a *request* to the object to perform services on its behalf. The implementation and location of each object are hidden from the requesting client.

The object request broker (ORB) is central to the CORBA OMA, as everything else depends on it. The CORBA specification [28] details the interfaces and characteristics of this component. There are many different ORB products currently available, provided by different vendors or geared to different operational environments. The ORB interoperability architecture [27] is designed to allow different ORBs to interoperate with each other, as well as with other middleware systems that are not CORBA compliant. In addition, separate ORBs may be desired for administrative reasons, such as enforcement of security, or to provide a protected test environment for product development.

CORBA introduces the higher-level concept of a *domain* to handle this partitioning, which is essentially a set of objects separated from all other objects. Objects from different domains need a bridging mechanism to map between domains so that they can interact. The bridging mechanism needs to take into account any policies in force on communication between the domains. The general ORB interoperability architecture is based on the General Inter-ORB Protocol (GIOP), which specifies transfer syntax and a standard set of message formats for ORB interoperability over any connection-oriented transport mechanism. GIOP is designed to be simple and easy to implement, while still allowing reasonable scalability and performance. The Internet Inter-ORB Protocol (IIOP) specifies how GIOP is built over TCP/IP transports. The ORB interoperability architecture also provides for other environment-specific inter-ORB protocols (ESIOPs), which allow ORBs to be built for interoperability with other distributed computing infrastructures.

In addition to standard interoperability protocols, standard object reference formats are also needed for ORB interoperability. While object references are hidden from applications, ORBs use them to help determine how to direct requests to objects. CORBA specifies a standard object-reference format, called the Interoperable Object Reference (IOR). An IOR stores information needed to locate and communicate with an object over one or more protocols.

Most commercially available ORBs support IIOP and IORs and have been tested to ensure interoperability. This facilitates interoperability between systems using different ORBs but still requires that specific ESIOPs be available for interoperability with other non-CORBA distributed environments. If no ESIOP is available for the non-CORBA environment, an interface object can be implemented in the ORB. As this object will deliberately not have access to the details of any clients or servers in the ORB, it will need to incorporate an internal bridging and brokering mechanism to forward requests between the CORBA and non-CORBA environments.

10.6.4 KAoS

The Knowledgeable Agent-oriented System (KAoS) [16–19] provides management services to ensure that agent systems from diverse sources can be used safely in operational environments. Bounds on agent behavior are defined by policies expressed in DARPA Agent Markup Language (DAML) or Ontology Web Language (OWL). These are declarative constraints on one or more agents that can regulate registration, access, encryption, resource use, agent mobility, agent obligations, and agent conversations. With the appropriate semantics, agent conversation policies can control the types, encryption, and even content of messages exchanged between agents, providing a much needed security layer.

KAoS is based on Sun's Java Agent Services [31] and is compatible with several agent and middleware frameworks, including Nomads [32], the CoABS Grid [12], Cougaar [24], and CORBA [28]. Although originally designed for agent systems, KAoS services can also be applied to general-purpose grid environments and Web Services.

The KAoS domain-management services allow agents to be grouped into logical domains and subdomains to facilitate agent-to-agent collaboration and external policy administration. Domains may represent any sort of group imaginable, from functional, organizational, and administrative structures to dynamic task-oriented teams with continually changing memberships. Domains can be nested indefinitely, and depending on whether policy allows, agents may belong to more than one domain at a time.

Policies can be applied to individual agents, agent classes, agent domains, agents on particular hosts or Java virtual machines (JVMs), or to all agents on the network. KAoS policy services allow specification, management, conflict resolution, and enforcement of policies for agents. The KAoS Policy Ontologies (KPO) distinguish between *authorizations* (constraints that permit or forbid some action) and *obligations* (constraints requiring that some action be performed or else waiving such requirements). The DAML ontologies used to represent KAoS policies enable runtime extensibility and adaptability of the system, as well as the ability to analyze policies relating to entities described at different levels of abstraction. This representation facilitates careful reasoning about policy disclosure, conflict detection and harmonization, domain structure, and concepts. The representation of classes in ontologies allows the effects of policies to be extended automatically through their subsumption into new classes of objects defined at a later time.

KAoS defines basic ontologies for actions, actors, groups, places, various entities related to actions (e.g., computing resources), and policies. For a given application, the ontologies can be further extended with additional classes, individuals, and rules. Actors and groups of actors or other entities in KAoS may be distinguished through explicit enumeration in some kind of registry (extensionally) or by virtue of some common property, such as a joint goal or a given location (intentionally). Through various property restrictions, a given policy can be variously scoped—

for example, to individual agents, agents of a given class, agents belonging to a particular group, or agents running in a particular computational environment (host or JVM).

KAoS allows an administrator to browse and load ontologies, to define, deconflict, and commit new policies, and to modify or delete existing policies. Groups of interdependent policies can be composed into *policy sets*. A generic policy editor gives knowledgeable administrators a tool that allows fine-grained control over any aspect of policy specification. It provides the user with a list of choices narrowed to only those appropriate in the context of the other current selections. Other custom editors tailored to particular kinds of policies can be incorporated into KAoS.

The KAoS Policy Ontologies can be used for a variety of purposes, including policy disclosure management, reasoning about future actions based on policies in force, and assisting users of policy specification tools in understanding the implications of new policies given the current context and set of policies already in force. Logical inference is required to determine which policies are in conflict and how to resolve these conflicts. Given two policy dimensions of authorization and obligation in KAoS, three types of conflicts are handled:

- Being simultaneously permitted and forbidden from performing some action;
- Being both required and forbidden to perform some action;
- Being both required and not-required to perform some action.

Policy deconfliction and harmonization algorithms within KAoS detect and resolve these conflicts. Policy precedence conditions based on a numeric priority are used to determine which of the two policies in conflict are most important, allowing the conflict to be resolved automatically in favor of the most important policy. Alternatively, conflicts can be brought to the attention of a human administrator for manual resolution.

Major components of the KAoS policy and domain services architecture are

- *Domain managers (DMs):* These manage domain membership and are responsible for maintaining policy consistency. They store policies in the directory service and distribute policies to guards as appropriate. Domain managers work together with the job transfer program (JTP) and the directory service to ensure policy consistency at all levels of the domain hierarchy. They handle persistence and queries to the directory service. Domain managers are stateless, so one DM instance may serve multiple domains, or, conversely, a single large domain may require several instances of the DM to achieve scalable performance.
- *Policy directory service (PD):* This acts as a secure repository for policies. It can respond to a variety of queries from the domain manager and other trusted entities in accordance with policy disclosure strategies. Policies in the directory service are expressed declaratively so that some forms of analysis and verification can be performed in advance and offline, permitting execution mechanisms to be as efficient as possible.

- *Guards:* These receive policies from the domain manager and interpret and enforce them with appropriate native mechanisms within the bounds of a computational environment. The DM maintains a mapping of guards and the policies for which they are responsible.
- *Enforcers:* These components are capable of enforcing particular types of policies on agents. New types of enforcers may be added based on the capabilities of the underlying execution and agent platforms.

While other components of KAoS policy and domain services are generic, enforcement mechanisms are necessarily platform specific. Enforcement mechanisms built into the execution environment (e.g., operating system or JVM) are the most powerful as they can generally be used to assure policy compliance for any agent or program running in that environment, regardless of how the agent or program was written. The Java Authentication and Authorization Service (JAAS) provides a mechanism that ties access control to authentication in the JVM. Other virtual machines (VMs), such as the Aroma VM [32], extend the JVM and provide other resource-control mechanisms. Another enforcement mechanism takes the form of extensions to particular multiagent infrastructures. Agents that use the default classes do not participate in domains; as a result, they are typically granted only very limited permissions in their interactions with domain-enabled agents.

Obligation policies required still another kind of enforcement mechanism. Because obligations cannot be enforced through preventative mechanisms, enforcers can only monitor agent behavior and determine after the fact whether a policy has been followed. Two sorts of enforcers can be used for obligation policies: *monitors* and *enablers*. Monitors simply monitor the state of an agent and either try to diagnose and fix any problems or alternatively levy appropriate sanctions against the agent. Enablers go beyond monitoring to facilitate or perform obligations proactively on behalf of the agent.

The DARPA CoAX [21–23] modeled coalition military operations and implemented agent-based systems to mirror coalition command-and-control structures, policies, and doctrine. The project aimed to show that agent-based computing offers a promising approach to dealing with issues like the interoperability of new and legacy systems, the interoperability of coalition policies, security, recovery from attack, system failure, and service withdrawal. KAoS provided the mechanism for overall management of coalition organizational structures represented as domains and policies. The CoAX experiments demonstrated how KAoS policies, in conjunction with the Aroma VM [32], could be used to manage agent resources via policy control and protect the coalition command-and-control system from denial-of-service attacks. Further, they demonstrated that KAoS policies could be used to transform and filter information released to other domains in line with security policies [33].

Within the DARPA Ultra*Log [25] program, KAoS is being used to provide agent policy and domain services to assure the robustness and survivability of logistics functionality in the face of information-warfare attacks or severely constrained or compromised computing and network resources. KAoS is also used within the NASA Cross-Enterprise and Intelligent Systems Programs, where policy-based models are used to drive human-robotic teamwork and adjustable autonomy for highly interactive autonomous systems, such as the Personal Satellite Assistant [34].

10.6.5 Joint Battlespace Infosphere

The Joint Battlespace Infosphere (JBI) differs from other middleware technologies in that it is an information-management framework rather than an agent framework or a mechanism for portable remote procedure calls. The difference is largely one of perspective. While the other middleware frameworks focus primarily on the functions or services provided by the computational components within a system, JBI focuses on the discrete pieces of information or the information objects they share. The JBI provides a flexible means to encapsulate, share, and find these objects. The JBI vision is that by focusing on the information, rather than services, open and extensible systems will naturally evolve.

Service-based architectures use a service to exchange information between a provider and consumer. The consumer must, either implicitly or dynamically, discover the appropriate information provider, agree to an exchange protocol with the provider, establish that the credentials of the consumer permit transfer, and establish the semantics of the content that is transferred. Not only must the format and meaning of content be established among components sharing the information, but often the meaning of the services themselves must be agreed upon. Furthermore, it is difficult to capture the side effects (intended and unintended) of a service invocation in a manner that is amenable to automated reasoning. Therefore, applications must generally rely upon compile-time human understanding. Not only is this error-prone among loosely coupled communities of users, but it is brittle in the face of changing service specifications.

JBI addresses these issues by establishing a general set of exchange mechanisms based on a publish-and-subscribe paradigm. Therefore, the mechanisms themselves have simple semantic interpretations and are static. Applications need only concentrate upon the format and interpretation of the information that is to be exchanged. There are few restrictions on the richness of content that can be shared via the publish-and-subscribe mechanisms of JBI.

The information published to JBI forms an information space that is managed in accordance with information-management policy; this includes access-control policy, prioritization and resource allocation policy, and information life cycle policy (e.g., persistence, auditing, and destruction). Because the information space is distinct from the applications that interact with it, third parties can interact with the space to find information without the knowledge of the provider, subject to the access-control policy of the space for that type of information. The JBI information space is populated by information objects. Each object is of a specific type, and its contents are described by metadata of a structure unique to that type. The content of the information object, the *payload*, may be any finite object. The information object is the fundamental unit of management; the JBI does not manage below this level. Applications that interact with the JBI (called *clients*) use the metadata to find information of interest. Specifically, they subscribe or query for information with predicates over the metadata of information objects in the information space. If a published information object's metadata matches the predicate of a subscriber, the information object is forwarded to the subscriber. Aside from a few metadata fields of "universal" applicability (e.g., publisher identity, publication time, and information object type), metadata specific to a type of information is not dictated by a platform. The community of JBI users determines the metadata structure for the information object types they will exchange. The JBI specifies a mechanism for extension that allows information object types to extend other existing types.

To manage information objects effectively, JBI requires that each object be of a specific type. Objects of the same type share the same metadata format and generally describe similar things. Information-management policy, such as access control, is generally determined based upon the type of information object. For example, the set of clients that can publish and subscribe to an air tasking order (ATO) is encoded as a policy over objects of type mil.af.ato.

Clients written to interact with the JBI are not bound to a specific implementation of JBI. One implementation of the JBI core services can be replaced with another without modifying the clients that interact with them. To accomplish this, applications interact with the JBI by invoking a common client application programming interface (CAPI). There may be many different implementations of the CAPI using different technology foundations. It is anticipated that different implementations will be better suited to different deployment situations. The CAPI defines the permissible interactions a client may have with the platform. Interactions between the JBI core services and clients outside the CAPI are not permitted since they would reduce implementation independence. The CAPI is a set of interfaces that control how a client finds a JBI implementation, connects to it, authenticates, creates information objects, and publishes and subscribes. In addition, it has methods to allow a client to browse, create, and update the universe of known information object types.

Through the CAPI, clients specify the versions of the information objects they understand. This is important since information object schema may evolve over time, so that clients may gradually fall out of date. To handle this situation, the platform will not deliver to a client an object of a higher version than that which the client requested, but the JBI core services may automatically translate a newer version of an object into an older format to deliver to an older client.

The JBI relieves clients of many responsibilities that they currently shoulder: implementation of dissemination mechanisms, access control, auditing, resource allocation and prioritization, version control, and archiving. These core information-management functions are consistently implemented and enforced by the JBI core services. Since these functions, inconsistently implemented by applications today, cause the bulk of interoperability problems, it is envisioned that a JBI will dramatically improve the ability to exploit existing applications, develop new ones, and gradually evolve both the information space and the clients that interact with it.

10.7 Information Sources

The information sources available to an IF system are critical to its effectiveness. These may include real-time sensor feeds, dynamic databases of multisensor data or fusion products, static databases of the geographic information system (GIS) and other a priori information, knowledge bases, Web resources, and so forth. In the intelligence, counterterrorism, law-enforcement, and business domains, information fusion allows the decision-maker to discover and explore patterns of behavior or activity. In these domains, information sources such as search engines, Web pages, and other Web resources may be just as, if not more, important than real-time sensor feeds.

An IF system, as shown in Figure 10.5, needs to be able to access a heterogeneous mix of databases as they cannot rely on a monolithic database or homogeneous database schemas if they need to access operational, legacy, proprietary, or coalition data stores. The ability to access and analyze dynamic Web resources is also critical. Key enabling technologies for this domain are:

- Data access and topic filtering;
- Information extraction;
- Data mining and model generation;
- Model-analysis tools.

The dynamic database (DDB) program discussed below demonstrates a large-scale system for storing and accessing large volumes of sensor data, making it

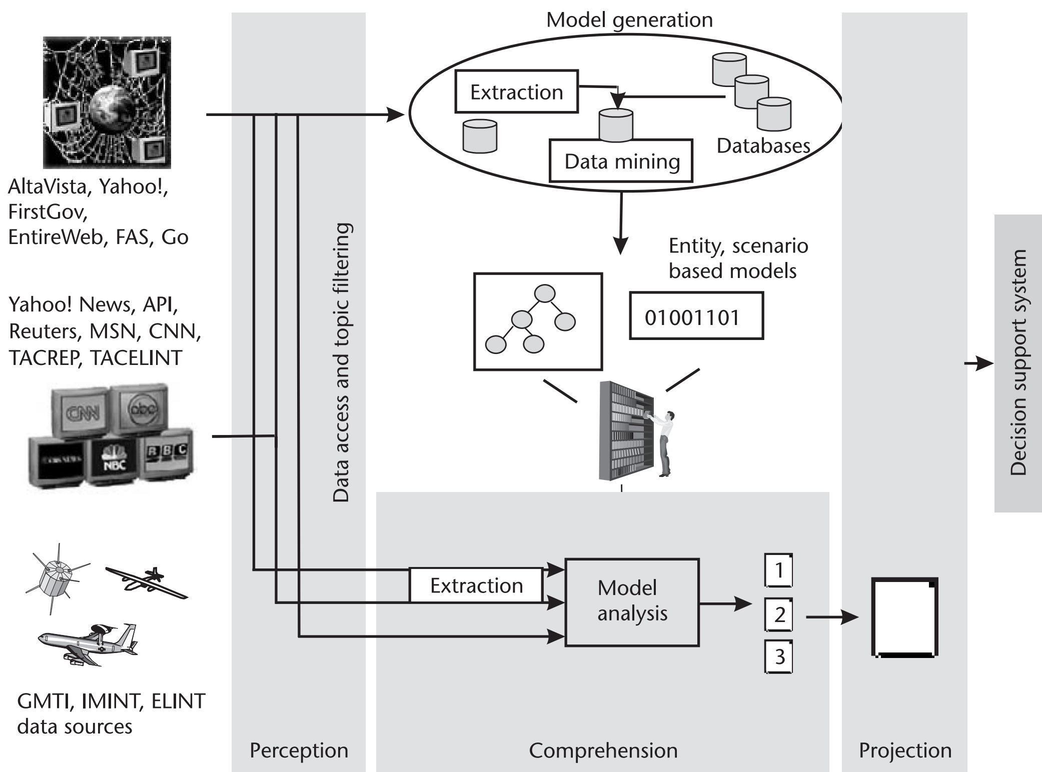


Figure 10.5 An IF model suitable for intelligence analysis.

available to military commanders. Such technology can be applied to other information sources and could be important in future IF systems.

To assist with automated access to databases by IF systems, information in different databases may need to be tagged with metadata that describes the data, or the databases themselves may need to be tagged with metadata that tells the IF system how to access and process records from the database. For this to be effective, a common, or at least systematic, metatagging scheme needs to be applied.

Similar issues and requirements arise when IF systems need to access information from Web resources, which are often poorly controlled and have little (or rapidly evolving) standardization. The Semantic Web attempts to resolve these issues through standardized representation of the content of Web resources, while other technologies, such as "Buddy," from the U.S. Air Force Research Laboratory (AFRL), seek to deal with unstructured data and to learn adaptively about changing Web resources.

Knowledge bases are also important to IF systems as they store semantic information that allow automated IF systems to represent and reason about the information from the sources appropriately. IF systems should be able to deal with heterogeneous knowledge bases and, where appropriate, automatically map between equivalent ontologies. The CoAX experiments [22, 23] demonstrated how, in a coalition command-and-control environment, semantic interoperability can be maintained between different coalition systems, enabling the discovery and utilization of distributed services [21] and, in particular, how sensor data from disparate sources can be translated and fused using DAML [35] ontologies.

10.7.1 The Semantic Web

The goal of the Semantic Web is to provide meaning and structure to information on the Web so that it is machine readable. This will facilitate automation of the Web (agents searching for and processing information for the user), improve searching and resource discovery, facilitate knowledge sharing, and improve the answering of high-level questions [20, 36]. Using this technology, IF systems can dynamically locate and access information sources on the Web to obtain up-to-date, relevant, or validated information from information warehouses.

To realize the Semantic Web, information on the Web needs to be structured. A suitable (and accepted) technology to achieve this is the Extensible Markup Language (XML), which can tag the information on the Web with metadata [36–40]. Meaning is provided through ontologies and a representation language to express them. The ontology offers a set of concepts (or classes), as well as their properties and relationships, relevant to the domain (or Web document). They describe elements within the Web document that will allow a machine to reason about its contents.

The concepts, properties, and relationships (terms) are identified by a universal resource identifier (URI) [38], which is a generic set of names and addresses of "things" (e.g., objects and relationships) of interest (the universal resource locator, or URL, is the most common type of URI [38]). A new term can be created by defining a URI for it. Ontology editing environments, such as Protégé [41] and OilEd [42], can assist in developing appropriate ontologies.

Representation languages describe how the ontologies and their instances are represented. The choice of representation languages must be made as a trade-off between expressiveness and computational properties (when inferencing). The most popular representation languages for the Semantic Web are Resource Description Framework (RDF) [43], its extension Ontology Interchange Language (OIL) [44, 45], and its successors DAML + OIL [46, 47] and OWL [48]. Their syntax is based on XML, thus eliminating the need for a specialized application parser [37].

RDF statements are sets of triples, comprising a subject, a predicate and an object [43]. For example, to represent "Ora Lassila is the creator of the resource www.w3.org/Home/Lassila" in RDF, the subject (resource) is "www.w3.org/Home/Lassila," the predicate (property) is "creator," and the object (literal) is "Ora Lassila" [43].

As discussed above, markup languages enable the creation of arbitrary domain ontologies that support the unambiguous description of Web content. Web Services [49] not only provide information to the user or autonomous system but also enable them to post information for access by other systems of users. Languages such as the Web Service Description Language (WSDL) [50] provide a low-level description of the messages and protocols used by Web Services. To complement this, semantic markup languages such as DAML-S and OWL-S [51, 52] are being developed that sit at the application level above WSDL, describing what is being sent and why, not just how it is being sent.

DAML-S and OWL-S make Web Services machine interpretable by enabling the following tasks [51]:

- *Discovery:* Locating Web Services, typically through a registry service, that provide a particular service and adhere to specified constraints;
- *Invocation:* Execution of an identified service by a user, agent, or other service;
- *Interoperation:* Breaking down interoperability barriers through semantics and automatically inserting message-parameter translations between clients and services;
- *Composition:* Composing new services through automatic selection, composition, and interoperation of existing services;
- *Verification:* Verifying service properties;
- *Execution monitoring:* Tracking the execution of complex or composite tasks performed by a service or set of services, thus identifying failure cases or providing explanations of different execution traces.

To achieve this, the DAML-S/OWL-S ontology consists of separate parts for:

- *Service profiles* that describe what the service provides and what it requires from the clients. Services advertise their profiles with networkwide *discovery services* that match service requests against the advertised profiles and identify which services provide the best match.
- *Service process models* that describe the workflow and possible execution paths of the service, including semantic descriptions of what are essentially

messaging APIs. Processes can be *atomic, simple*, or *composite* and have defined inputs and outputs that can be connected to other processes. Atomic processes are invokable in a single step, while simple and composite processes are noninvokable abstractions of atomic processes or combinations of processes.

- *Service groundings* that describe how atomic processes are to be mapped into various messaging formats, such as those provided by the Simple Object Access Protocol (SOAP) [53], Hypertext Markup Language, CoABS Grid [12], and so forth. The central function of the OWL-S grounding process is to show how the (abstract) inputs and outputs of atomic processes are to be realized as messages in some transmittable format. Service grounding is critical to the successful deployment of OWL-S since it provides the connection between abstract concepts in OWL-S and implementation standards such as WSDL.

DAML-S and OWL-S are designed to be complementary to WSDL: OWL-S provides abstract specifications to the developer with the benefits of OWL-S's process model and OWL's expressiveness, while also providing the extensibility of WSDL and its software support for message exchanges. The CoAX experiments have demonstrated how OWL and OWL-S can be used to transform and filter semantic data in coalition command-and-control systems [33].

SOAP [53] is another useful messaging protocol for IF systems, which, as discussed above, can be used as a grounding for OWL-S Web Services. SOAP is a lightweight protocol for the exchange of information in a decentralized, distributed environment. It is an XML-based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined data types, and a convention for representing remote procedure calls and responses. SOAP can potentially be used in combination with a variety of other protocols, but typical bindings for SOAP are currently with the Hypertext Transfer Protocol (HTTP) and HTTP Extension Framework.

10.7.2 Buddy

The Semantic Web is an attempt to impose structure and standardized protocols on the Web to facilitate access by automated processes. However, most resources available on the Web today do not follow any particular standard or set of standards; rather, they are ad hoc and designed solely for human access. This means that most Web sites are poorly structured for machine readability. Additionally, Web sites may be poorly maintained, may contain out-of-date hyperlinks, or may change rapidly.

This poses a challenge for automated access and analysis of these information sources. The AFRL Buddy system was developed as a meta–search engine that can access these Web resources. Current meta–search engines key on the common search engines, which only accessed around 60% of the Web in 2000 and around 40% in 2001 [54], and exclude specialty or regional search engines. They may rely heavily on Web scraping, which is very brittle to changes in the Web pages searched, and send the same request to multiple engines irrespective of the nature of the request. The AFRL Buddy search engine [55] is designed to improve information collection and to access a greater portion of the Web by accessing appropriate specialist or regional Web sites. It uses intelligent parsing of pages to extract information from unstructured documents to reduce brittleness to changes and to target requests to appropriate sites.

Buddy can simultaneously query and access multiple Web-based resources and can request multiple topics at a time using topic trees to allow a hierarchical query that refines the search space, as shown in Figure 10.6. It uses "self-healing" Web adapters to access Web sites and returns ranked responses using a unique ranking algorithm. As well as resource discovery, Buddy can also be used for document retrieval and has an API that allows its search kernel to be reused in other applications. Buddy can extract data from free-text and semistructured documents using fixed keywords or tokens, entities (e.g., names, places), or events (e.g., who, what, where, when).

Buddy has been used in several U.S. programs, including an AFRL program to build and maintain an equipment database automatically and another program geared toward the automatic classification and electronic archiving of free-text (paper) documents.

10.7.3 Dynamic Database

DARPA initiated the dynamic database program in 1998 to address the escalating problem of making the massive amounts of intelligence, surveillance, and reconnais-

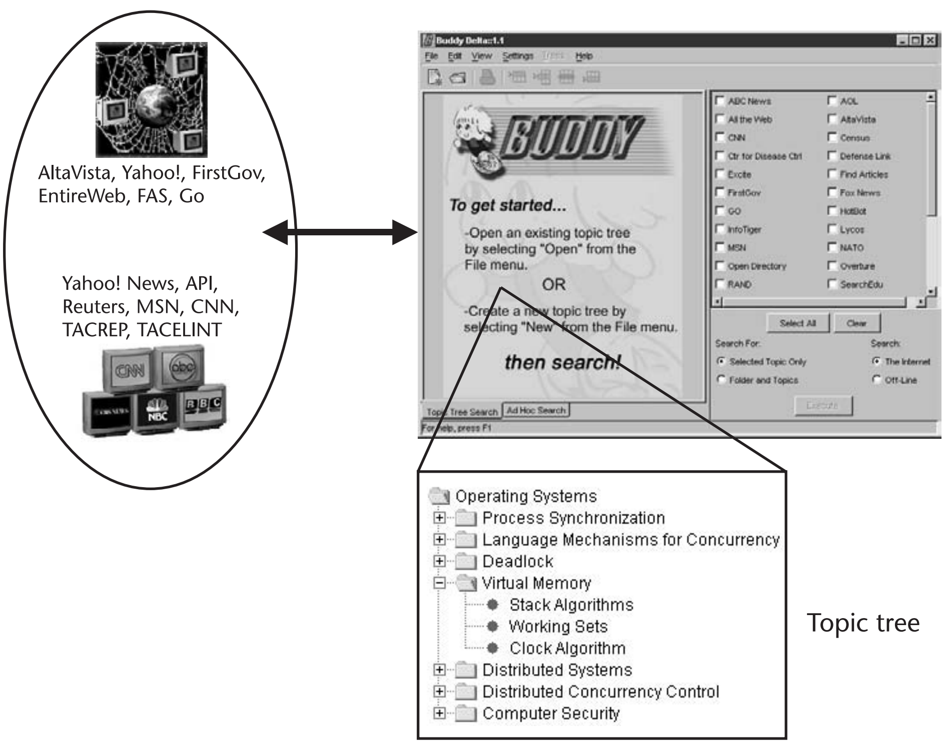


Figure 10.6 Buddy meta–search engine uses topic trees for queries.

sance data available to operational forces in actionable form. The system needed to manage the large volume of available data from diverse sensors and sources and to provide the multi-intelligence (INT) fusion necessary to convert sensor data into actionable information within the time demands required to respond to emerging threats within the targeting decision cycle.

A core element of the DDB was the database framework, known as the High Performance Data Store (HPDS), which allowed multisensor data and National Imagery and Mapping Agency terrain products to be ingested, automatically co-registered, and stored in logically separate sensor history databases. This enabled the DDB to rapidly store, index, retrieve, and share massive quantities of data among DDB components [56]. This data includes not only sensor and foundation data but also results of the detection, fusion, and reasoning algorithms. Therefore, a critical part of the DDB system is a rich common object-oriented schema that includes the raw sensor data, registration data, features, associations such as tracks, and models of real-world entities and their environments. The DDB application components reside on the DDB framework and conduct data mining of the co-registered data. The architecture for the DDB, showing how multi-INT inputs are processed to provide the information needed by the commander. is illustrated in Figure 10.7.

Typical commercial database-management systems, used as-is, could not meet the requirements of DDB to keep up with the constant stream of incoming sensor data; nor were they able to index it so that spatially and temporally coincident

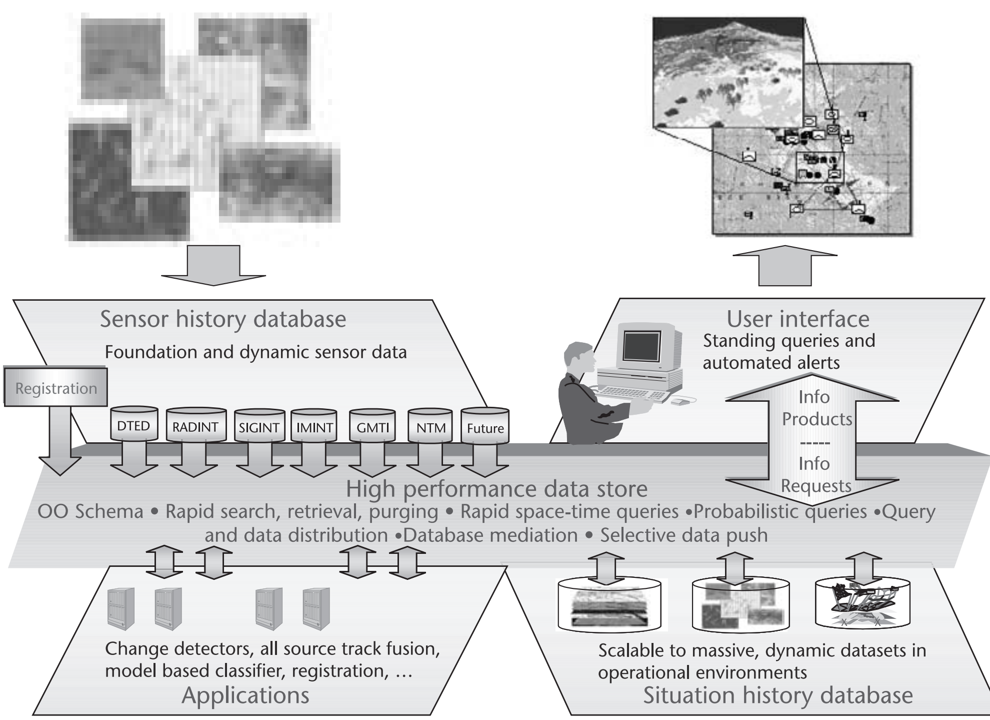


Figure 10.7 Architecture for DDB.

data could be efficiently retrieved. The HPDS was built by leveraging a commercial object database but was extended with hierarchical spatial and temporal indexes. To make room for new data, obsolete data is efficiently purged as necessary without significantly impacting performance.

Sensor errors, registration errors, association errors, and possible errors in identification all contribute to uncertainty in DDB results. A rigorous representation of uncertainty is being added to the schema, and eventually to the database query engine itself, that will allow algorithms to reason on and combine uncertain information correctly.

In addition, HPDS provides active database mechanisms that can trigger an algorithm to run on new data, notify a user of a significant event, or determine what computations must be run to answer a user's ad hoc query. The system can be "loosely" distributed across multiple locations, with data and queries selectively replicated to support an operational environment with possibly unreliable communications networks.

Registration of many disparate data types is accomplished by using processes and strategies that register all incoming data to a common fiducial, provide a common geospatial coordinate system for all applications, and perform common registration processes in a single application [57]. All DDB objects are geolocated in four dimensions: space (latitude, longitude, and elevation) and time. Cross-platform systematic errors are automatically removed or reduced during "background" processing. A registration success metric provides operators with a measure of registration accuracy. Position knowledge of DDB objects and reference data is continuously refined and improved over time through the use of salient features that are observable across the sensors. Features are automatically defined and refined as precision georegistered invariant features, which occur across multi-INT data (e.g., road intersections, stationary rotators) and are used to make associations across multi-INT data for error propagation and correction.

10.8 Human-Computer Interfaces

Humans are an essential component of the situation-analysis process. The goal of IF systems is to enable situation awareness for the decision-maker, as exemplified by the situation-analysis paradigm for information fusion. Ultimately, this requires human-computer interfaces that effectively convey the IF products to the decision-maker in a manner that supports human cognitive processes. Human-computer interfaces are thus an important component of IF systems.

A major goal of IF systems is to manage massive volumes of information so as to prevent "information overload" in an operator or commander. Information overload is not just the problem of presenting the user with large volumes of information. Humans have evolved perceptual mechanisms to deal with an environment that routinely presents massive volumes of sensory information and in which dangers and opportunities arise rapidly. Information overload can even occur from seemingly manageable volumes of relevant information. Studies with command-and-control systems have shown that the performance of commanders can actually decrease when too much apparently relevant information is available to them [58].

The role of IF systems is then not just to reduce a flood of information to the human but also to select the appropriate information and transform it so that it is presented in a form that is tuned to human perceptual and cognitive processes. This allows the IF system to complement human cognitive abilities, such as the ability to see patterns and their implications, and to make rapid decisions based on real-time analysis of rapidly changing information. IF systems should attempt to complement this human ability with the computer's ability to perform some aspects of mathematical, statistical, and logical analysis reliably and orders of magnitude faster than humans. This human-machine partnership requires good displays and interaction techniques.

Trust is central to military command-and-control systems. Human participation in the IF process is essential to allow the user to build trust in the process, particularly when "intelligent" systems attempt to present abstract concepts and infer users' intent rather than simply present raw data and execute tightly defined commands. The appropriate use of technology to enable or reinforce trust is thus essential for military command-and-control systems, which include IF systems. There are also other social dimensions that need to be considered in interfaces to IF systems (e.g., how does the user interact and collaborate with others? How does the task physically affect the user? How does the task affect users' morale, hence operational effectiveness?).

10.8.1 Visualization

The NATO Research and Technology Organization's IST-013/RTG-002 technical team describes visualization as follows [59]:

> Visualisation is something humans do. . . . Visualisation is not a data display, however ingenious. It is one route to understanding, another route being logical analysis. Complicated displays, such as virtual reality displays, can help visualisation, but humans can easily visualise situations and events when reading the text of a well-written novel that has no pictures at all. The nature of the display is not irrelevant, but it is not the whole story.

The team goes on to further describe visualization as the formation of an internal picture of the environment, or at least the part of it that is currently important, so as to recognize dangers and opportunities and act effectively in it. Using this description, situation awareness can be considered the desired end-state of visualization and the "other routes to understanding" mentioned above. Any of these routes can be used to achieve situation awareness, but visualization can be an effective route to the understanding of massive amounts of (perhaps abstract) data by matching its representation to human perceptual and cognitive processes.

10.8.2 Human Factors

The human factors associated with human-computer interaction will be essential to providing an effective IF system. These factors address the following questions:

- How should the information be fused and presented so that the human can visualize it effectively?
- How can the human control the fusion and display processes to accommodate a dynamic environment?
- How do IF systems affect the roles of humans in a C2 system?
- What are the personnel selection and training requirements for different IF systems and visualization schemes?
- What implications might there be on the health of the users. For instance, what are the long-term effects of immersive 3-D displays on user performance?
- How do particular visualization schemes perform for a user under stress?

The IST-013/RTG-002 team identifies four main classes of goals for computerized visualization [59], which represent different modes of operation of the human user and computer system. Each of these modes has different implications for IF systems.

10.8.2.1 Monitoring and Control

In this mode, the user attempts to keep track of some aspect of the data space that varies over time and to influence the data space through interface devices. The IF system therefore needs to extract the appropriate features from the data space and reliably present them so that the user can readily identify changes and the appropriate responses. The user must be able to specify to the IF system the features to be monitored, which may be an abstract property of the data space, such as enemy intent. The input interfaces must also allow the user to control the visualization and IF processes. Monitoring and control is aimed at achieving situation awareness of some aspect of the data space and when it is visualized.

10.8.2.2 Alerting

In this mode, autonomous systems need to monitor the data space for the occurrence of any of a number of possible trigger conditions. When these conditions are met, the IF system needs to suppress temporarily whatever is currently being visualized (and is considered less important) and present the alert conditions. The user needs to be able to define what needs to be monitored and the conditions under which to raise an alert. The display systems need to be able to show the user that an alert condition has occurred, together with its context, without interfering with whatever the user is currently monitoring. The user must then be able to dismiss or defer the alert and continue with the task at hand or to shift focus from the current task and handle the alert. Alerting is aimed at achieving situation awareness about aspects of the data space other than that currently being visualized, as the user is aware that there are no alert conditions currently extant.

10.8.2.3 Searching

This mode is used when some aspect of the data space being monitored has uncertainty associated with it, which may be resolved by additional information that is not readily apparent. To accommodate searching, the IF system must indicate what other aspects or features of the data space are available. Displays for searching therefore need to show how the data space is organized and how the user can access unseen parts of the data space. Searching is aimed at improving the accuracy of the user's situation awareness for the currently monitored aspect of the data space.

10.8.2.4 Exploring

This mode is used when the user wishes to learn the features of the data space to support future monitoring and control tasks. Exploration is aimed at identifying aspects of the data space that are likely to remain unchanged so that they can be accessed when they are needed in the future. Exploring is aimed at improving users' situation awareness by increasing their knowledge of the context of information visualized in the data space.

Displays need not only to discriminate patterns and highlight relationships in the data space, but they must also evoke some useful conceptualization. When useful relationships can map onto topological and geometric properties, the corresponding relationships can be represented in a 2-D or 3-D display space. Similarly, some properties can readily be mapped to color and texture.

However, abstract information does not necessarily map cleanly to human perceptual models. Visual metaphors are often used to represent and interact with data in a familiar environment, such as the common desktop metaphor or a 3-D metaphor. Abstract concepts may not be suited to visual metaphors and may be more effectively represented symbolically or linguistically. For example, free-text descriptions of real or fictional scenarios and events can allow the user to visualize rich and complex worlds, potentially allowing a one-to-many mapping of the text to the visualization. A movie generated from the same description presents a one-to-one mapping of spatial and physical features, but the user must infer and visualize any abstract relationships, such as personalities or political and cultural associations. The visualization of the abstract components may in fact be more difficult once a spatial visualization has been provided.

10.8.3 Cognitive Factors

Visualization for IF systems should take advantage of human cognitive processes and should allow for, and perhaps supplement, human cognitive limitations. For example, human attention is time limited; a human cannot easily comprehend relationships among more than a few things at a time; short term memory is capacity limited; concepts once formed are hard to correct with counterevidence; a display metaphor may be misleading when carried to an extreme.

An important issue in dynamic IF (and C2) systems is displaying what the user wants to see in such a way so as to prime a rapid, correct understanding of new material and, at the same time, jog the user out of persisting with false interpretations. A related issue is how to display alerts that possibly have different contexts than the current task. The display must present the alert and context in a form that users can readily comprehend, given that they were just involved in some other task prior to the alert, and it must discourage users from forming an incorrect interpretation of the alert, thereby perhaps giving the alert unwarranted attention or inappropriately dismissing it.

10.8.4 Presentation Systems

Presentation systems act as the interface between users and IF systems. They must not only display information effectively but also allow the user to navigate through the data space. They should support both visualization and logical analysis of the data. This leads to two apparently competing factors in the design of presentation systems—logical analysis is best done with only a few entities or relationships considered at a time, but visualization usually requires that an extended context be displayed to the user and is not as effective with a small number of entities or relationships. Presentations systems that present many entities or relationships and context for effective visualization therefore need to highlight to the user which few are appropriate for analysis.

There are some common presentation modes available that can be used in isolation or in combination.

Text displays can present symbolic, labeled, and linguistic data. They can be either line based or table based, using a "screen" partitioned into a fixed number of lines that are a fixed number of characters wide. This allows low-resolution symbolic maps to be displayed on these systems. They are commonly used to display linguistic information or as a command-line interface to applications. Input for these systems is via a keyboard, although speech-enabled systems have also been developed.

Speech-recognition systems allow the user to interact with the visualization system using speech—either commands in a structured command language or less structured conversational or extemporaneous speech. Speech-recognition systems can be speaker dependent or speaker independent. Current speaker-dependent systems require users to train the system to recognize their speech, but they use very general language models in order to recognize a wide domain of spoken language. Studies have shown that these systems perform satisfactorily when the user dictates prepared material but less well when attempting to recognize conversational or extemporaneous speech [60]. Speaker-independent systems do not require training of the system but are generally constrained to specific, task-oriented language models in order to improve recognition accuracy. This is sufficient for issuing commands or phrases using a formal command language but, again, is not suitable for conversational or extemporaneous speech.

Text-to-speech (TTS) systems allow the automatic generation of speech from text. This can provide an aural input channel to the user but is demonstrably slower than reading written text (this is left as an exercise to the reader or speaker). However, additional information can be transferred via speech through emotional cues and word stress. Speech recognition and TTS systems allow the display of, and interaction with, symbolic, linguistic, and labeled data.

"Display" systems can also use other audio output—the pitch, amplitude, and modulation of a tone or melody—to convey analogue, symbolic, and labeled data to the user. Typically, audio signals have been used for alarms or alerts, but much richer information can potentially be conveyed. The main disadvantages of audio "displays" are that they interfere with user-to-user interaction, and unlike the visual systems, the audio channel is not something that the human can easily "switch off."

Audio spatialization can also be used to "display" located data in 3-D [61]. This form of audio display is often associated with virtual reality displays. 2-D screen displays can present all commonly used data types, including 3-D data, by using projections onto the 2-D plane of the display. They allow the display of text, symbols, and images. Interaction with these displays typically combines use of a keyboard and a 2-D pointing device—such as a mouse or trackball—that controls the position of a pointer on the 2-D screen.

Desktop and windows metaphors are typically used to interact in these systems, with selection controlled by pop-up or drop-down menus and dialog (text) boxes. Selection of windows of interest on the screen can be done using a pointing device or keyboard "hot-keys." 2-D displays can be speech enabled to allow interaction with selected windows. Selection of windows of interest is harder with speech devices and relies on keyboard or pointer emulation or verbal "hot-keys." Finer selection of located data within a 2-D window is problematic with keyboard or speech interfaces and usually relies on 2-D pointing devices.

Humans have evolved to cope with a 3-D world, and have developed perceptual mechanisms to prevent "information overload" and resolve features of interest from background clutter using depth cues (such as stereoscopic vision, object occlusions, and head motion). Studies have shown that, in some domains, stereoscopic displays can be used to show around 1.6 times as much information as 2-D displays, and up to 3 times as much information if simulated head motion is included [62].

Virtual reality (VR) displays use left-eye/right-eye (L/R) pairs of stereoscopic images, each generated from a slightly different perspective, to generate depth in the fused images perceived by the user. VR displays can present all of the data types suitable for 2-D displays, with the advantage of stereoscopic cues that can reduce visual clutter. This allows much more effective presentation of 3-D analogue data than 2-D displays (that must rely on 2-D projection with only perspective and motion depth cues).

The stereoscopic systems used in VR displays only produce the images from a fixed perspective—users cannot "look around" the image unless the display device generates new stereoscopic images in response to a viewer's change in head position—in which case only a single user would perceive the effect correctly. In multiuser environments this means that the 3-D effect seen by observers at other positions will appear skewed, an effect that increases with the distance from the design point.

VR systems can be designed to be either immersive, where users appear to be "inside" the display environment and the viewpoint is controlled "naturally" by head-tracking systems; semiimmersive, where users appear to be inside the display but must control the viewpoint artificially; or nonimmersive, where they appear to be viewing the display from outside.

A number of visual display technologies are used for VR systems. Passive and active VR systems can be used in either desktop or large-screen projection formats that allow large fields of view. Desktop systems generally allow only a single user to interact with the system, while large-screen projection systems allow multiple users. VR visual displays are often coupled with 3-D spatialized audio systems. This can be used to enhance the perception of depth for the representation of 3-D entities or to provide an additional sensory channel for data display [61].

VR displays can use the range of interaction devices and paradigms available to 2-D displays but also need to include a mechanism for depth selection and navigation. Conventional 2-D pointing devices can be used to select data intersecting with the "line of sight" of the pointing device but will not be able to select data behind the first intersection. With sparse datasets, users can manipulate their viewpoints to "see around" any obstructing data.

3-D pointing devices allow the user to select any point in 3-D space directly without needing to navigate to the required viewpoint. These devices either track motion and orientation in 3-D space, emulating this in the virtual environment, or use a stationary input device with 6° of freedom. Gesture-recognition and hand-tracking systems, which use image-processing techniques [63] to allow 3-D navigation without the need to manipulate a device, promise a natural 3-D interaction paradigm.

One issue with 3-D pointing devices is that they allow the user to navigate with 6° of freedom in 3-D space, which is an unfamiliar environment for humans who have evolved to navigate on an essentially 2-D plane. This can make navigation difficult, and the degrees of freedom available to the user are often constrained to cope with this.

10.8.5 FOCAL

The Future Operations Centre Analysis Laboratory (FOCAL), shown in Figure 10.8, is a Defence Science and Technology Organisation program to explore new paradigms in situation awareness for the Australian Defense Organisation. FOCAL is designed to be a multiuser collaborative environment, where command teams can share context and interact with the display, as well as with each other. FOCAL is a semiimmersive VR system with a large, 150°, field-of-view, spherical screen, using two sets (L/R) of three passive stereo LCD projectors with an SGI Onyx3400 providing image-generation capabilities. Thus, a large amount of screen real estate can be used to display contextual information to users, while providing the advantages of a 3-D display.

The visualization challenges faced by FOCAL include what, how, and when information should be presented to the users to achieve effective situation awareness; how a team of users can interact effectively with the display; and what the cognitive and physiological factors associated with this environment when used in a C2 role are. FOCAL must also address the technological challenges of how to design suitable visualization and IF engines.

FOCAL implements a multiagent architecture based on the DARPA CoABS [12] middleware as a model for information retrieval, processing, and fusion. For information visualization, a model for situation awareness based on television news services is being explored. There are three key elements to these services:

Figure 10.8 FOCAL.

- *Advisers:* These provide expert commentary that conveys situation awareness through explanation.
- *Maps and diagrams:* These are visual props to aid in explanation (e.g., weather maps, stock market charts).
- *Photographs/video footage:* These convey context and provide experiential content.

FOCAL is investigating the effectiveness of this paradigm as a mechanism for achieving situation awareness in the military domain by essentially providing these services as software. They will be portable, accessible, and interactive—the commanders can then access them wherever and whenever they want to, and they can interact with them to explore the situation at the level of detail or abstraction they require.

As discussed above, effective interaction by a team of users with the FOCAL system is key to its effectiveness as a collaborative C2 environment. A number of interaction mechanisms are being explored for FOCAL:

- Traditional keyboard and 2-D pointing devices to allow users to interact with a suite of COTS applications;

- 3-D pointing devices, such as wands and space balls, to allow navigation in 3-D displays;
- Speaker-identification, speech-recognition, and TTS dialogue systems;
- Gaze and gesture tracking to provide a "natural" interface with the FOCAL systems.

FOCAL will support a number of metaphors for user interaction and presentation, including the standard desktop and X-windows metaphors, natural-language dialogue, and immersion in a virtual 3-D world. The latter is currently being explored using a prototype 3-D application dubbed the Virtual Planning Room (ViPR), as shown in Figure 10.9. This environment will allow users to immerse themselves in the planning environment and to monitor, search, and explore the situation data space. This will allow display of the information and context associated with situations in virtual "situation rooms" and facilitate the development of military courses of action.

Users will dialogue with the FOCAL systems using various mechanisms. Integration and management of interaction with multiple users will be context sensitive and dependent on user selection, preferences, and dialogue history. Selection of presentation format for visualization will likewise depend on context and user selection, preferences, and dialogue history. Dialogue and presentation management will use software agents and user input.

10.8.5.1 Virtual Advisers

One of the key elements of FOCAL is the virtual adviser [64–66]—a real-time, animated agent (see Figure 10.10) that will take on the role of television news advisers, and dialogue with users through spoken natural language and TTS systems. Virtual advisers could brief the command team on a developing situation, point out significant events for further attention, and suggest alternative courses of action.

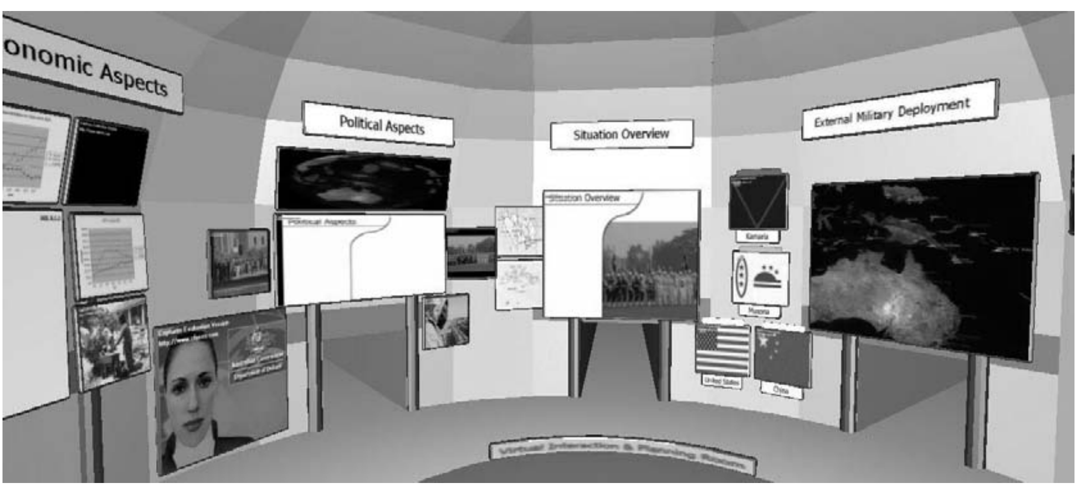


Figure 10.9 Prototype ViPR as a 3-D planning environment in FOCAL.

Figure 10.10 FOCAL virtual adviser used in CoAX 2002 demonstration.

Virtual advisers can access information resources on request, and by utilizing text, images, multimedia, and other presentation systems in FOCAL, they can present the results to users in preferred or selected formats. By combining facial gestures and emotional cues, virtual advisers can also convey the appropriate levels of trust and context that are normally associated only with face-to-face interaction between humans. These systems have been used effectively in pedagogical systems [67, 68], but exactly what the appropriate levels of trust are in a command-and-control environment, and what gestures and emotional cues are needed to convey this, are ongoing research topics.

Virtual advisers allow users to search for and explore the information data space through natural-language dialogue and also provide an alerting function rich in context via briefings on developing situations. In combination with interaction channels, such as gaze and gesture tracking, virtual advisers can provide a very familiar interface to users.

10.8.5.2 Virtual Video

The software equivalent of television news video footage in FOCAL is the virtual video system. This is still under development, but the intent is to generate animated (re)constructions automatically (see Figure 10.11) of emerging or developing situations or the potential consequences of alternate courses of action. This program

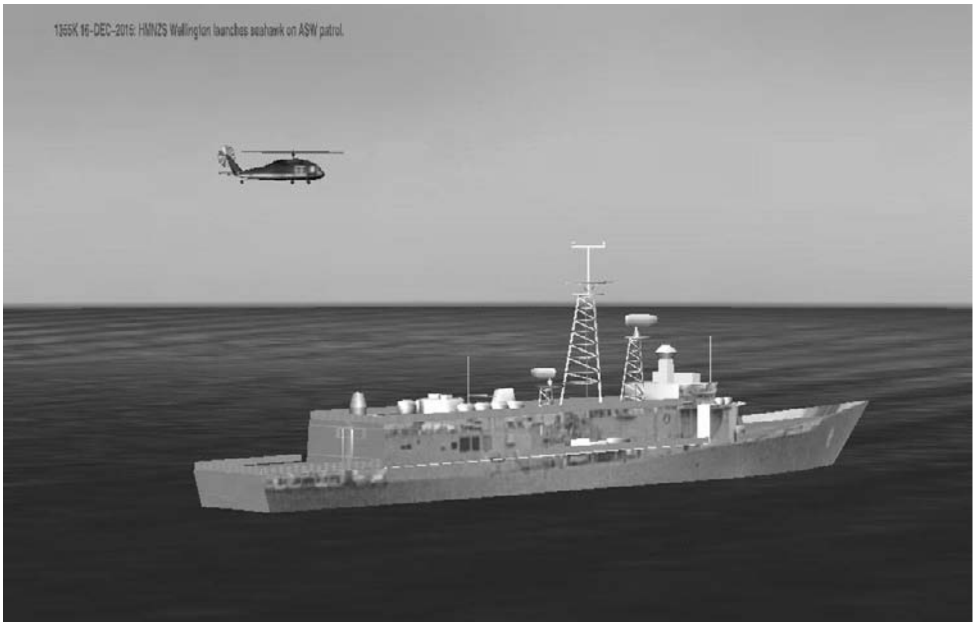


Figure 10.11 Animation generated by virtual video system to represent hypothetical situation.

will involve research into a number of domains, including automatic fact extraction, formal theories of knowledge representation [69], and automated selection and depiction of key events in a format not only suitable but interesting to users—in essence, attempting to codify at some level the "director's art."

The goal of this work is to convey contextual and experiential information to users and to investigate the effects of immersion in this way on their situation awareness. Key to the effective use of this tool is research into how to convey the appropriate level of trust (or distrust) in the information presented in this way, and how to retract information that is later identified as erroneous or misleading.

References

[1] Waltz, E., and J. Llinas, *Multisensor Data Fusion*, Norwood, MA: Artech House, 1990.

[2] Hannebauer, M., and R. Kühnel, "Dynamic Reconfiguration in Collaborative Problem Solving," *Proceedings of the Workshop on Concurrency, Specification and Programming (CS&P-99)*, Warsaw, Poland, 1999.

[3] Durfee, E. H., and V. R. Lesser, "Partial Global Planning: A Coordination Framework for Distributed Hypothesis Formation," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 21, No. 5, 1991, pp. 1167–1183.

[4] Moore, G. E., "Cramming More Components onto Integrated Circuits," *Electronics*, Vol. 38, No. 8, 1965, pp. 114–117.

[5] Moore, G. E., "An Update on Moore's Law," *Intel Developer Forum*, 1997, http://www.intel.com.

[6] ITRS, "International Technology Roadmap for Semiconductors," http://public.itrs.net 2001.

[7] Bundy, A., "A Survey of Automated Deduction," Division of Informatics, University of Edinburgh, Edinburgh, U.K., 1999.

[8] Cocks, C., et al., "The Theatre Broadcast System," DSTO-TN-0287, Defence Science and Technology Organisation, Edinburgh, South Australia, 2000.

[9] Alberts, D. S., J. J. Garstka, and F. P. Stein, *Network-Centric Warfare: Developing and Leveraging Information Superiority*, 2nd ed., Washington, D.C.: National Defense University Press, CCRP Publication Series, 1999, Ch. 1, http://dodccrp.org.

[10] Cebrowski, A. K., and J. J. Garstka, "Network-Centric Warfare: Its Origin and Future," *Proceedings of the U.S. Naval Institute*, JCS J-6, January 1998.

[11] Stein, F. P., "Observations on the Emergence of Network-Centric Warfare," *ICCRTS Proceedings*, June 1998, http://www.dodccrp.org/files/stein_observations/steinncw.htm, accessed November 11, 2006.

[12] DARPA, CoABS program Web page, http://www.darpa.mil/ipto/research/coabs/2003.

[13] Global InfoTek, CoABS Web site, 2003, http://coabs.globalinfotek.com.

[14] Kahn, M., et al., "DARPA CoABS Grid Users Manual," http://coabs.globalinfotek.com/public/downloads/Grid/documents/GridUsersManual.v4-draft.doc 2002.

[15] Kahn, M. L., and C. Della Torre Cicalese, "CoABS Grid Scalability Experiments," *Second International Workshop on Infrastructure for Agents, MAS, and Scalable MAS, Autonomous Agents 2001 Conference*, May 29, 2001.

[16] Bradshaw, J. M., et al., "Making Agents Acceptable to People," in N. Zhong and J. Liu, (eds.), *Handbook of Intelligent Information Technology*, Amsterdam: IOS Press, 2003.

[17] Bradshaw, J., et al., "Representation and Reasoning for DAML-Based Policy and Domain Services in KAoS and Nomads," *Autonomous Agents and Multi-Agents Systems 2003 (AAMAS'03)*, Melbourne, Australia, 2003.

[18] Bradshaw, J. M., "Terraforming Cyberspace: Toward a Policy-Based Grid Infrastructure for Secure, Scalable, and Robust Execution of Java-Based Multi-Agent Systems," *Computer*, Vol. 34, No. 7, 2001, pp. 49–56.

[19] Bradshaw, J. M., et al., "KAoS: Toward an Industrial-Strength Generic Agent Architecture," in J. M. Bradshaw, (ed.), *Software Agents*, Cambridge, MA: AAAI Press/MIT Press, 1997, pp. 375–418.

[20] Hendler, J., T. Berners-Lee, and E. Miller, "Integrating Applications on the Semantic Web," *Journal of the Institute of Electrical Engineers of Japan*, Vol. 122, No. 10, 2002, pp. 676–680.

[21] Allsopp, D. N., et al., "Toward Semantic Interoperability in Agent-Based Coalition Command Systems," *Proceedings of the 1st International Semantic Web Working Symposium (SWWS'01)*, http://www.semanticweb.org/SWWS, 2001.

[22] Allsopp, D. N., et al., "Coalition Agents Experiment: Multiagent Cooperation in International Coalitions," *IEEE Intelligent Systems*, May/June, 2002, pp. 26–35.

[23] Wark, S., et al., "Dynamic Agent Systems in the CoAX Binni 2002 Experiment," *Special Session on Fusion by Distributed Cooperative Agents, Fusion 2003*, Cairns, Australia, July 7–10, 2003.

[24] DARPA, Cougaar Web site, 2003, http://www.cougaar.org.

[25] DARPA, Ultra*Log Web site, 2003, http://www.ultralog.net.

[26] BBN Technologies, "Cougaar Architecture Document," http://www.cougaar.org, version 10.0, 2003.

[27] OMG, "Common Object Request Broker Architecture: Core Specification," http://www.omg.org, 2002.

[28] OMG Web site, 2003, http://www.omg.org.

[29] Vinoski, S., "CORBA: Integrating Diverse Applications within Distributed Heterogeneous Environments," *IEEE Communications*, Vol. 35, No. 2, 1997.

[30] Yang, Z., and K. Duddy, "CORBA: A Platform for Distributed Object Computing (a State-of-the-Art Report on OMG/CORBA)," *Operating Systems Review*, Vol. 30, No. 2, 1996, pp. 4–31.

[31] Microsystems, S., Java Agent Services Web site, 2003, http://www.java-agent.org.

[32] Suri, N., et al., "NOMADS: Toward an Environment for Strong and Safe Agent Mobility," *Proceedings of Autonomous Agents 2000*, Barcelona, Spain, 2000, http://www.ecs.soton.ac.uk/~pg03r/mypapers/nomads2000.pdf.

[33] Suri, N., et al., "DAML-Based Policy Enforcement for Semantic Data Transformation and Filtering in Multi-Agent Systems," *Autonomous Agents and Multi-Agent Systems Conference (AAMAS'03)*, Melbourne, Australia, 2003.

[34] Bradshaw, J. M., et al., "Adjustable Autonomy and Human-Agent Teamwork in Practice: An Interim Report on Space Applications," in H. Hexmoor, R. Falcone, and C. Castelfranchi, (eds.), *Agent Autonomy*, Norwell, MA: Kluwer, 2002.

[35] DARPA, DAML Web site, 2003, http://www.daml.org.

[36] Berners-Lee, T., J. Handler, and O. Lassila, "The Semantic Web," *Scientific American*, Vol. 284, No. 5, 2001, pp. 34–43.

[37] Semantic Web site, 2003, http://SematicWeb.org.

[38] Berners-Lee, T., and E. Miller, "The Semantic Web Lifts Off," *ERCIM News*, Vol. 51, October 2, 2002.

[39] Berners-Lee, T., Semantic Web Road Map, 2003, http://www.w3.org/DesignIssues/Semantic.html.

[40] Hendler, J., "Agents and the Semantic Web," *EE Intelligent Systems and their Applications*, Vol. 16, No. 2, 2001, pp. 30–37.

[41] Stanford University, "The Protégé Project," http://protege.stanford.edu, 2000.

[42] Manchester University, OilEd Web site, 2003, http://oiled.man.ac.uk.

[43] W3 Consortium, RDF Web site, 2003, http://www.w3.org/TR/PR-rdf-syntax.

[44] OIL homepage, http://www.ontoknowledge.org/oil, 2003.

[45] Fensel, D., et al., "OIL in a Nutshell," in R. Dieng, et al., (eds.), *Knowledge Acquisition, Modeling, and Management, Proceedings of the European Knowledge Acquisition Conference* (EKAW-2000), Lecture Notes in Artificial Intelligence, LNAI, Berlin: Springer-Verlag, 2000.

[46] Hendler, J., and D. L. McGuinness, "Darpa Agent Markup Language," *IEEE Intelligent Systems*, Vol. 15, No. 6, 2001, pp. 72–73.

[47] Connolly, D., et al., "DAML + OIL Reference Description," http://www.w3.org/TR/daml + oil-reference-20011218, 2001.

[48] Dean, M., et al., "OWL Web Ontology Language 1.0 Reference," http://www.w3.org/TR/2002/WD-owl-ref-20020729, 2002.

[49] McIlraith, S., T. C. Son, and H. Zeng, "Semantic Web Services," *IEEE Intelligent Systems*, Vol. 16, No. 2, 2001, pp. 46–53.

[50] Christensen, E., et al., "Web Services Description Language (WSDL) 1.1," http://www.w3.org/TR/2001/NOTE-wsdl-20010315, 2002.

[51] Ankolekar, A., et al., "DAML-S: Web Service Description for the Semantic Web," *Proceedings of the ISWC 2002*, York, U.K., April 8–11, 2002.

[52] Ankolekar, A., F. Huch, and K. Sycara, "Concurrent Semantics for the Web Services Specification Language DAML-S," *Proceedings of the ISVR 2002*, 2002.

[53] Mitra, N., "Simple Object Access Protocol (SOAP) 1.1," http://www.w3.org/TR/SOAP, York, U.K., April 8–11, 2002.

[54] Sullivan, D., "The Search Report," http://searchenginewatch.com, December 18, 2001.

[55] "Buddy," Patent Pending RL 10,042.

[56] Parr, J. T., T. Stephenson, and H. Kindle, III, "A Scalable Dynamic Object Database for DDB," TASC Corporation, Reading, MA, 2001.

[57] Hackett, J., et al., "Multi-INT Registration Fusion (U)," Harris Corporation, Information Systems Division, Melbourne, FL, 2003.

[58] Omodei, M. M., et al., "More Is Better? Problems of Cognitive Self-Regulation in Dynamic Decision Making," *International Conference for Cognitive Research with Microworlds*, Granada, Spain, November 12–14, 2001.

[59] IST-013/RTG-002, "Visualisation of Massive Military Datasets: Human Factors, Applications, and Technologies," RTO-TR-030 AC/323(IST-013)TP/9, NATO Research and Technology Organisation, Quebec, Canada, 2001.

[60] Broughton, M., "Measuring the Accuracy of Commercial Automated Speech Recognition Systems During Conversational Speech," *Virtual Conversational Characters (VCC) Workshop, Human Factors Conference 2002 (HF2002)*, Melbourne, Australia, 2002.

[61] Doornbusch, P., and J. Watson, "Sound and Reality," *SimTecT 2002*, Melbourne, Australia, 2002.

[62] Ware, C., and G. Frank, "Evaluating Stereo and Motion Cues for Visualising Information Nets in Three Dimensions," *ACM Transactions on Graphics*, Vol. 15, No. 2, 1996, pp. 121–140.

[63] Authot, J., "Vision-Based Gesture Recognition as an Interface to Virtual Environments," Ph.D. Thesis, Department of Computer Science, Australian National University, Canberra, Australia, 2002.

[64] Taplin, P., et al., "Situation Awareness Using a Virtual Adviser," *Talking Head Workshop, OZCHI 2001*, Fremantle, Australia, 2001.

[65] Broughton, M., et al., "Conversing with Franco, FOCAL's Virtual Adviser," *Virtual Conversational Characters (VCC) Workshop, Human Factors Conference (HF2002)*, Melbourne, Australia, 2002.

[66] Lambert, D. A., "Advisers with Attitude for Situation Awareness," *Proceedings of the 1999 Workshop on Defense Applications of Signal Processing*, La Salle, IL, 1999.

[67] Marriot, A., et al., "The Face of the Future," *J. Research and Practice in Information Technology*, Vol. 32, No. 3/4, 2000, pp. 231–245.

[68] Gratch, J., and S. Marsella, "Tears and Fear: Modeling Emotions and Emotional Behaviours in Synthetic Agents," *AGENTS'01*, Montreal, Canada, 2001.

[69] Lambert, D. A., "Situations for Situation Awareness," *Fusion 2001*, Montreal, Canada, 2001.

CHAPTER 11

Knowledge-Based and Artificial Intelligence Systems

Steve Wark and Dale Lambert

As discussed in Chapter 10, IF systems that deal with high-level data fusion (i.e., situation and impact assessment) in the JDL data-fusion model generally rely on symbolic (or nonnumeric) reasoning systems. This is because symbolic reasoning facilitates reasoning with relations between objects and the effects of those relationships. In this chapter, we briefly discuss knowledge-based and alternative artificial intelligence system computational techniques that can be applied to the IF domain.

11.1 Reason

The term *automated reasoning* most commonly refers to the rational symbolic manipulation of stored representations. Symbols are constructed to represent aspects of interest in the world. Automated reasoning involves the computational manipulation of symbols to model reasoned conclusions about those aspects of interest.

The *classical planning problem* typifies the classical reasoning approach. It involves the description of:

- An *initial condition*;
- A *goal condition*;
- A *set of operators* or, when parameterized, a *set of operator schema*.

The initial and goal conditions describe possible states of the environment, with each state description being an instantaneous, abstract, and partial characterization of what the environment might be like, expressed in the spirit of the situation calculus of McCarthy [1]. Each state is described by an axiom expressed in some formal language L associated with a possibly implicit formal logic $<L, H>$, with inference relation $H \subseteq ((L \times L) \to L)$. The operator schemata are used to define operators representing actions that transform one state into another. Each operator schemata can be described in terms of:

- A *precondition*, being a state description of what the environment must be like in order for an operator instance of that schemata to be applied;

- A *descriptor*, used to identify that operator schemata;
- A *postcondition*, being a state description of what the environment will be like after an operator instance of that schemata has been applied.

For operator x, the precondition of x is denoted by pre(x), and the postcondition of x by post(x). On occasion, all three attributes of operator schemata will be parameterized [2].

A *plan* for any given set of operator schema is a partial ordering of operators instantiated from the operator schema. A plan is said to be executed whenever a linear ordering of operators from the plan is applied to generate a sequence of actions. The initial state description is intended to describe the environment when plan execution commences. The goal state description is intended to describe what that environment should be like when plan execution has completed. Formally, the classical planning problem is the problem of producing a plan such that:

- If i is the initial condition, then there is some operator a in the plan for which i H pre(a).
- If operator a immediately precedes operator b in the plan, then post(a) H pre(b).
- If g is the goal condition, then plan execution will halt after operator a for which post(a) H g.

Expressed intuitively, the classical planning problem is the task of generating a plan, that, when executed, performs a sequence of actions that transform the initial condition of the environment into the goal condition of the environment. The activity of generating such a plan is called planning, and programs that undertake this activity are called classical planners. Classical planners are further distinguished according to dependency, hierarchy, linearity, and conditionality.

11.2 Reaction

There is a dispute over whether externally induced reaction or internally conceived reason is the catalyst for intelligence. The *subsumption architecture* of R. A. Brooks [3–5] exemplifies the extreme reaction alternative. It consists of a vertical layering of hard-wired finite-state machines, as shown in Figure 11.1.

Coordination within the architecture proceeds by suppression and inhibition,[1] while the architecture functions as a direct sensor-to-effector mechanism. There are no concepts, no stored representations modeling the world, no represented plans, and no temporal reasoning, though activities are coordinated by an internal

1. *Suppression* occurs on the input side of a finite-state machine when the upper-level routine suppresses the lower-level machine by seizing, for some designated amount of time, all messages sent to the latter. In this way, higher-level machines usurp control of lower-level machine's inputs. *Inhibition* occurs on the output side of the finite-state machine when messages from the lower-level machine are inhibited for a specific amount of time by a message from a higher-level machine. In this way, higher-level machines prevent lower-level machines from interfering with their actions.

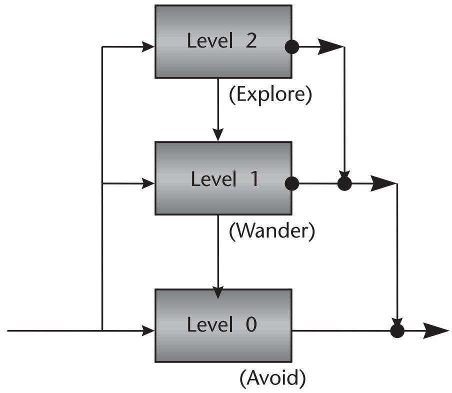


Figure 11.1 Subsumption architecture.

clock. The architecture is engineered in an ad hoc, incremental fashion by guessing which tasks should subsume others and by performing trial-and-error modifications until the machine behaves properly. Brooks promotes this engineering strategy as modeling evolution. The central tenets of Brooks's moboticist position are elegantly summarized by Kirsh [6]:

> (1) Behavior can be partitioned into task-oriented activities or skills, such as walking, running, navigating, collecting cans, vacuuming, chopping vegetables, each of which has its own sensing and control requirements which can be run in parallel with others.
>
> (2) There is a partial ordering of the complexity of activities such that an entire creature, even one of substantial complexity, can be built incrementally by first building reliable lower-level behavioral skills and then adding more complex skills on top in a gradual manner.
>
> (3) There is more information available in the world for regulating task-oriented activities than previously appreciated; hence, virtually no behavioral skill requires maintaining a world model. If you treat the world as external memory, you can retrieve the information you require through perception.
>
> (4) Only a fraction of the world must be sampled to detect this task-relevant information. Smart perception can index into the world cleverly, extracting exactly what is needed for the task control without solving the general vision problem.
>
> (5) The hardest problems of intelligent action are related to the control issues involved in coordinating the various behavioral abilities so that the world itself and a predetermined dominance or preference ordering will be sufficient to decide which activity layer has its moment in the sun.

The extreme reactionist view supplants rationalism with naturalism and replaces the manipulation of stored representations with external interaction.

11.2.1 Neural Networks

Neural networks provide an alternative nonsymbolic reactive approach. The field of neural networks became popular in the late 1980s early 1990s. A neural network is a function-approximation technique that maps n-dimensional Euclidean space to m-dimensional Euclidean space, where m is less than or equal to n [7]. Given a set of inputs and a desired set of outputs, a neural network can be trained to approximate any function well. The basis for artificial neural network theory is A. Kolmogorov's theory of mapping neural network existence, as given in [7]. Kolmogorov's theorem states that any continuous function f can be exactly implemented by a three-layer, feed-forward neural network having n fan-out processing elements in the first layer, $(2n + 1)$ processing elements in the hidden layer, and m processing elements in the top layer [7]. The theorem does not, however, state what connections are needed from the first layer to the hidden layer or from the hidden layer to the top layer. The theorem is mainly an assurance that a continuous function can be mapped exactly by a three-layer feed-forward neural network. The Stone Weierstrauss theorem states that, given the class of squashing functions, we can uniformly approximate any continuous function [8]. This is purely an existence theorem that allows for the use of artificial neural networks. It does not state how to construct a neural network or how to train its weights. Neural networks contain a set of weights that must be determined to approximate functions. To train neural networks, techniques such as back propagation [7, 9], evolutionary programming [10], and the extended Kalman filter [11] have been used.

A standard paradigm for a neural network is a feed-forward connection known as the multilayer perceptron. A multilayer perceptron equation is shown in (11.1).

$$NN_m = \sum_{i=1}^{N} w_{im} * \left(f_i \left(\sum_{k=1}^{J} I_k * w_{ki} \right) \right) \tag{11.1}$$

where NN_m is the mth output of the neural network, w_{im} is the mth output weight connected to the ith hidden node, and

$$f_i = \frac{2}{1 + \exp(-x_i)} - 1 \tag{11.2}$$

is the output of the ith hidden node; x_i is the dot product sum of the previous input layer's output with the connecting weights of the hidden layer; w_{ki} is the kth input weight connected to the ith hidden node; and I_k is the kth input feeding the neural network.

Figure 11.2 is an example of a three-layer perceptron neural network. The diagram shows three inputs into the first layer. The circles in the diagram represent nodes which can have many inputs but only one output. The output of each node in the first layer is sent to every node of the hidden layer. In the hidden layer, a dot product sum is performed for all inputs and their corresponding weights. This sum is then sent through the sigmoidal function to transform the linear dot product sum into a nonlinear function that becomes the output of each node in the hidden layer as shown in Figure 11.3.

The output of each hidden layer node is then sent to every node of the top layer, and another linear weighted sum is performed with the corresponding weights between the two layers. There are weights between each node that need to be determined for a neural network to perform a correct function mapping. In order

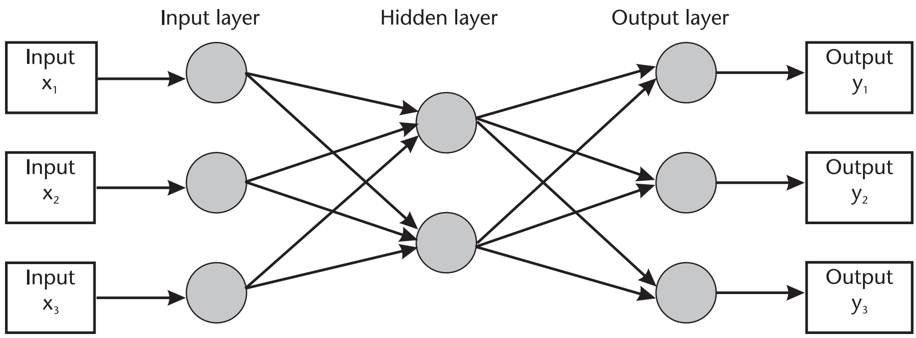


Figure 11.2 A neural network feed-forward architecture.

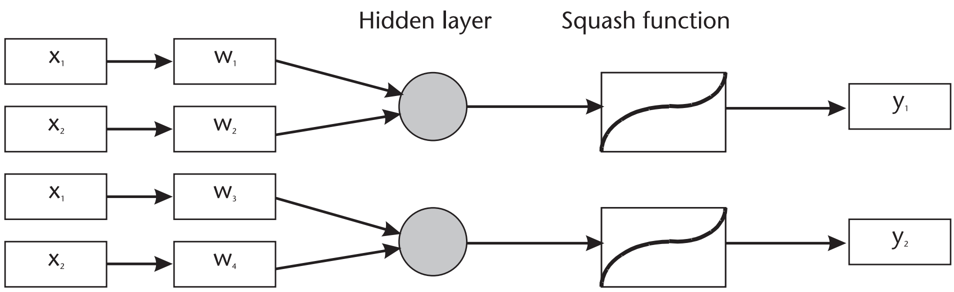


Figure 11.3 A view of a hidden layer, including the sigmoid squashing function.

to find the correct set of weights, a neural network has to be trained. Many different types of training methods exist, such as back propagation, least mean squares (LMS), and Hebb learning laws that determine a neural network's set of weights [7].

An example of a function-mapping problem is demonstrated with the exclusive-or function. The exclusive-or function, $y = f(x_1, x_2)$, is a binary function that contains four input-output pairs: $f(0, 0) = 0$, $f(1, 0) = 1$, $f(0, 1) = 1$, and $f(1, 1) = 0$. Figure 11.4 represents the exclusive-or function in a 2-D grid. The axes represent x_1, the first input, x_2, the second input, and the output, y, coming out of the page. A hyperplane is drawn from (–1, –0.5) to (1, 2) representing a linear sum $0 = w_0 + w_1 a_1 + w_2 a_2$, where w_0, w_1, and w_3 represent a set of weights [7]. A hyperplane can be a line in 2-D space or an ordinary plane in 3-D space. The hyperplane in Figure 11.4 is unable to put grid points (0, 1) and (1, 0) on one side and (0, 0) and (1, 1) on the other. Therefore, a hyperplane alone cannot perform the correct function mapping. Figure 11.5 represents a possible mapping that a neural network can perform. This figure demonstrates that a neural network can form a closed curve to encompass the desired function mapping. In Figure 11.6, the mean squared error of the neural network as it is training is plotted versus the training iteration number. This figure showns that, by iteration 20, the neural network's mean squared error is at zero. This implies that the neural network has learned and mapped the exclusive-or function exactly. Figure 11.7 shows the neural network weights as they train over the 20 iterations. Note that by iteration

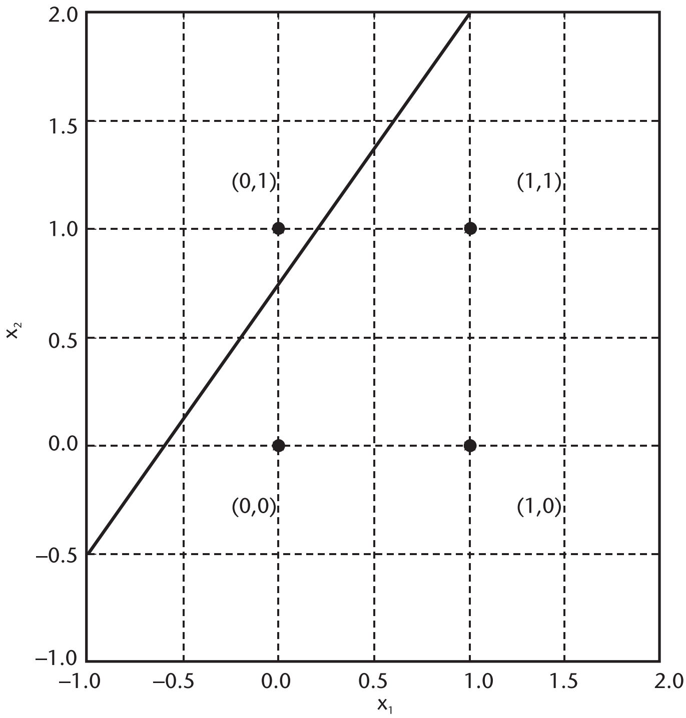


Figure 11.4 A 2-D plot of the exclusive-or function.

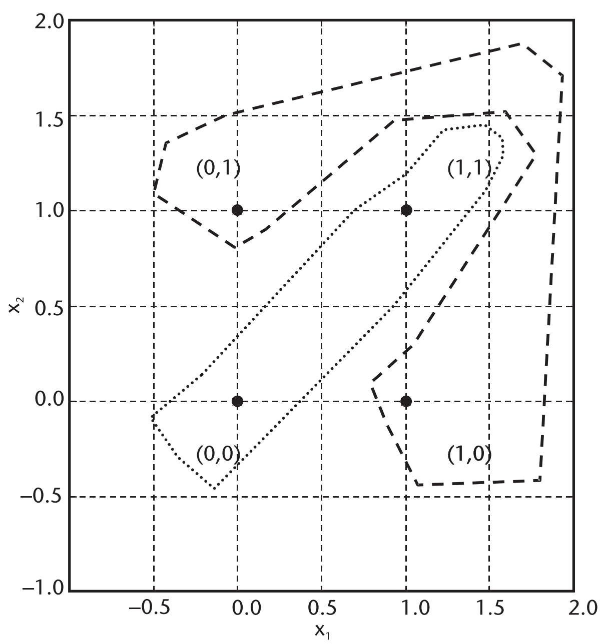


Figure 11.5 A possible neural network function mapping.

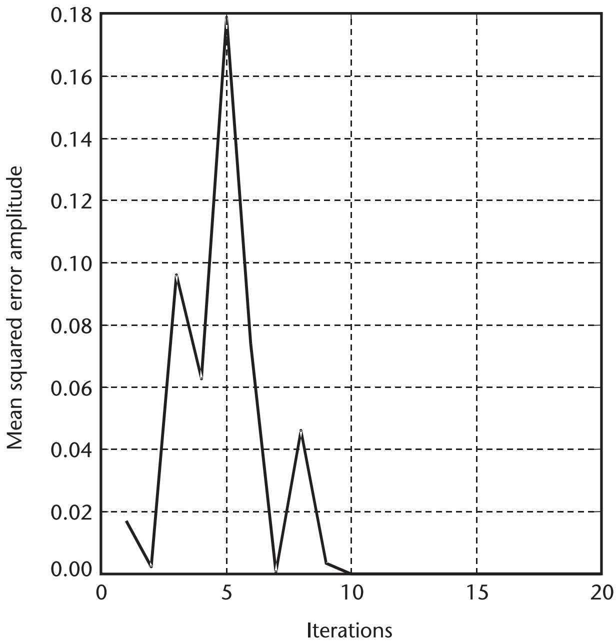


Figure 11.6 Neural network training error versus iteration count.

8, the neural network weights are in steady state, and little training occurs during the next 12 iterations.

In terms of the computational aspects of neural networks when applied to data-fusion process, there will not be an impact. Neural networks are incredibly fast and can approximate any desired function needed by the data-fusion system. Once a neural network has been trained to approximate a function, it can be inserted into the system to execute whenever data is available. The speed of the neural network is due to its dot product functionality. For a three-layer neural network architecture, there will be an input vector applied to a dot product of an input weight matrix. The dot product sums are then evaluated using a squashing function for the hidden layer, and a final dot product of the squashing function outputs with an output weight matrix is evaluated to form the outputs of the neural network.

As with all function-approximation techniques, enough data needs to be available to approximate the desired function properly. Unavailability of data to train artificial neural networks for a data-fusion system could make neural network technology unavailable. Computationwise, there is little affect, though, for a data-fusion system using artificial neural networks.

11.2.2 Reactive Planners

The standoff between the reactionist's sophisticated world and the reasoner's sophisticated mind is complicated when hybrid strategies between the two are acknowledged. As the first paper to seriously advance the possibility of engineering

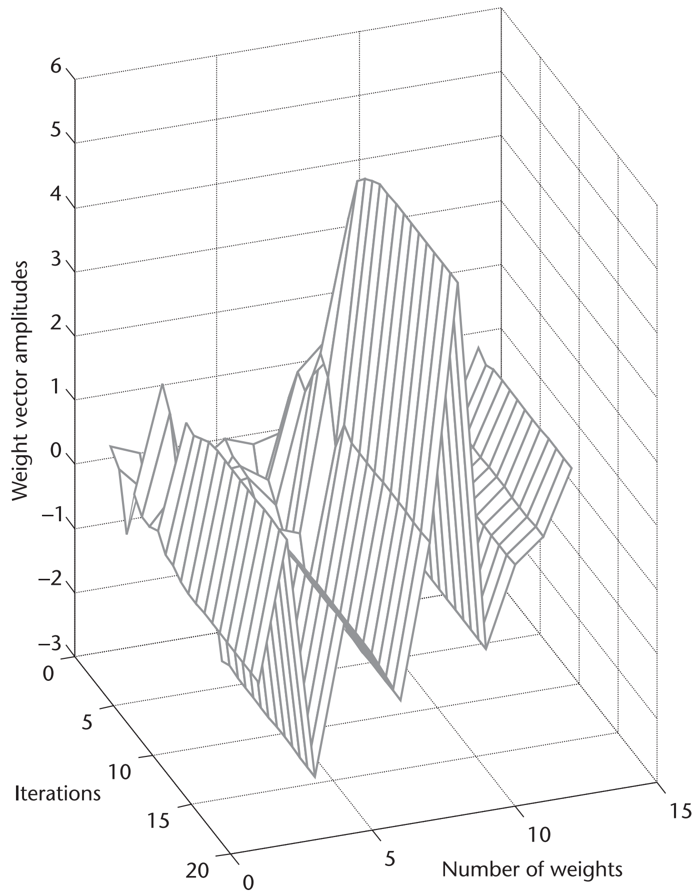


Figure 11.7 Input and output neural network weights versus training iteration.

intelligent machines,[2] the second of Turing's trilogy [12–14] offers some insight. The prospect of engineering intelligent machines through the manipulation of stored representations rested with [13], exposing the relationship between the universal Turing machines[3] of [12] and the emerging electronic digital computers of Turing's

2. This is not to suggest the notion of intelligent machinery was conceived originally by Turing, merely that he was the first to market the idea seriously. For example, it is widely known that John von Neumann toyed with the notion, but ultimately embraced a pessimistic attitude toward it. It is also worth noting that between the first two volumes of the Turing trilogy [12, 13], a number of events transpired. Of course, one of them was the arrival of the electronic digital computer, with Turing himself at the forefront of its development. In addition, Warren McCulloch and Walter Pitts had published "A Logical Calculus of the Ideas Immanent in Nervous Activity" in 1943, a paper in which they showed that neural configurations could be modeled by propositional logic and therefore that the central nervous system could be modeled as a machine. That same year also marked the publication of "Behaviour, Purpose and Teleology" by Arturo Rosenblath, Norbert Wiener, and Julian Bigelow. In it they exposed some of the nervous system's most common characteristics as "circular processes," thereby reducing ostensibly purposeful activity to feedback mechanisms. Taken together, these events suggested that if the minds were a product of entirely physical components, then a mechanical portrayal of the mind was perhaps a legitimate one.
3. In [13] they are called "universal logical computing machines."

day.[4] But Turing's vision for engineering intelligent machines favored a more naturalist appraisal. Interaction with the world was a motivating theme, though it was curbed to an extent in the interest of public safety [13, p. 13]:

> One way of setting about the task of building a "thinking machine" would be to take a man as a whole and to try to replace all the parts of him by machinery. He would include television cameras, microphones, loudspeakers, wheels and "handling servo-mechanisms," as well as some sort of "electronic brain." This would be a tremendous undertaking of course. . . . In order that the machine should have a chance of finding things out for itself, it should be allowed to roam the countryside, and the danger to the ordinary citizen would be serious. . . . Instead, we propose to try and see what can be done with a "brain" which is more or less without a body, providing, at most, organs of sight, speech, and hearing.

Engineering an intellect would begin with the supervised learning of *reactions* [13, p. 14]:

> If we are trying to produce an intelligent machine, and are following the human model as closely as we can, we should begin with a machine with very little capacity to carry out elaborate operations or to react in a disciplined manner to orders (taking the form of interference). Then, by applying appropriate interference, mimicking education, we should hope to modify the machine until it could be relied on to produce definite reactions to certain commands. This would be the beginning of the process.

This would culminate in sophisticated *search strategies* [13, pp. 22–23]:

> A very typical sort of problem requiring some sort of initiative consists of those of the form "Find a number *n* such that . . ." This form covers a great many problems. . . . We might arrange, however, to take all possible arrangement of choices in order, and go on until the machine proved a theorem, which, by its form, could be verified to give a solution of the problem. This may be seen to be a conversion of the original problem into another of the same form. Instead of searching through values of the original variable *n* one searches through values of something else. In practice, when solving problems of the above kind, one will probably apply some very complex "transformation" of the original problem, involving searching through various variables, some more analogous to the original one, some more like a "search through all proofs." Further research into intelligence of machinery will probably be greatly concerned with "searches" of this kind. We may perhaps call such searches "intellectual searches."

Not all reactive architectures fall within the extreme reactionist scheme. In the mid-1980s, the planning subdiscipline of classical artificial intelligence underwent an internal struggle. To the protagonists, the classical planning problem no longer fully expressed their ambition. The impetus for change was the inadequacy of

4. Specifically, Turing argued that the ACE computer he was working on at the time would, in effect, constitute a universal machine if its memory capacity were infinitely extended.

classical planners in the real world. The shortcomings are noteworthy, given the classical planning commitment to search and reasoning through the manipulation of stored representations.

- Classical planning is a very time-consuming exercise and is, broadly speaking, provably both computationally intractable and undecidable.[5] Planning undecidability requires that we should either not plan or else plan conservatively.
- The classical planner traditionally prepares its plan, then executes it. At least three drawbacks accompany this approach:
  a. The real world will often change between the inception of plan formation and plan execution, thereby making the plan to be executed redundant (see [15]).
  b. The classical planner operates with a superficial world model, and this often engenders problems of robustness during execution. A more environmentally interactive planner could potentially obviate some of the difficulties. (See [16] for a demonstration of this point with the problem of the DARPA autonomous land vehicle believing it has wandered into a radio terrain mask.)
  c. The requirement that all planning must precede execution necessitates that the programmer be able to determine the effects of the robot's actions a priori; outside contrived blocks worlds, this is often not possible.
- Classical planners traditionally formulate their plans based on the primitive actions the robot can perform. This is often not the best policy since many plans are not constructed from first principles (see [15]).
- Classical planners traditionally construct their plans afresh on each occasion. This is not only a waste of resources, but also significantly affects reactivity. Humans seem to rely on predefined plans to at least some extent. Plan libraries have become the norm.
- Classical planners traditionally formulate their plans blindly, without sensory access to the external world. Consequently, the plan-formulation process is unable to compensate for a changing environment during plan execution.
- Classical planners traditionally possess no concept of urgency and perform no reasoning under uncertainty. They are suited only to problems in which the world of interest is a clearly discernible product of their own actions.

Figure 11.8 identifies some of the better-known embedded architectures emanating from the period. They are ordered in the figure by an increasing design

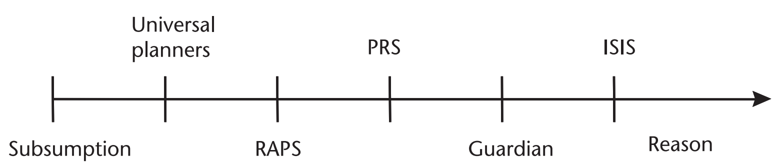


Figure 11.8 Embedded architecture continuum.

5. This, of course, depends on the representational capacity of the planner [2].

emphasis toward reason, independently of the extent to which those additional design facilities are actually used in any particular application.

Following the generalization of [17], it is convenient to call the collection of related, low-level fixed architectures to the immediate right of the subsumption architecture universal planners, after [18].[6] Rather than devise plans as the need arises, universal planners come with a ready made assortment of stored reactions to possible situations, which are combined as the situation dictates [24, p. 52]:

> A universal plan is executed by repeatedly starting at the top of the decision tree, testing the conditions encountered on the way down, branching right or left as appropriate, then executing the action found at the bottom. Thus, universal plans amount to deeply nested if-then-else's inside a do-forever.

Universal planners resemble the rapid, cyclic, perception-action character of the subsumption architecture, but their architectures are invariably arrived at in a more reasoned fashion through specially deployed programming language constructs, which may or may not be explicit in the final product. Universal planners can accommodate very weak world models in the form of internally manipulated state descriptions. They can also accommodate some longer-term computations spanning several perception-action cycles.

Beyond universal planners, the architectural emphasis extends further toward reason, with a proportional increase in both a dependency on internally manipulated representations (world model) and the sophistication of the representation language.[7] Woodridge cites the following objections to purely reactive architectures [37, p. 97]:

- Reactive planners without models of their environment must have sufficient information in their local environment to determine acceptable actions.
- Reactive planners tend only to have a short-term view because they rely on current state information only.
- Reactive planners tend not to learn from experience.
- Reactive planners often exhibit behavior that emerges from component systems, which makes it difficult to engineer them through a principled methodology.
- Reactive planners are difficult to engineer when they are complex systems involving many layers.

In general, the more the architectural emphasis advances reason, the more classical its representation becomes. Debate has raged over the extent to which reason should dominate reaction. In proposing the procedural reasoning system

6. Though important differences are involved, we include within this general class the situated automata of [19–23], the universal planners of [18, 24, 25], and the routine architectures of [26, 27].
7. In increasing reasoning architectural complexity, we have listed Charles Firby's reactive action packages [28, 29], Michael Georgeff's Procedural Reasoning System [15, 30, 31]; Barbara Hayes-Roth's Guardian [32–34] with the underlying BB1 blackboard architecture developed in [35] and the ISIS schedular of [36].

(PRS) architecture, [15] clearly distances itself from the lesser reason-based embedded architectures by remarking [15, p. 677][8]:

> The ability to act appropriately in dynamic environments is critical for the survival of all living creatures. For lower life forms, it seems that sufficient capability is provided by stimulus-response and feedback mechanisms. Higher life forms, however, must be able to anticipate future events and situations, and form plans of action to achieve their goals. The design of reasoning and planning systems that are embedded in the world and must operate effectively under real-time constraints can thus be seen as fundamental to the development of intelligent autonomous machines.

As an advocate of the subsumption architecture, Brooks [5, p. 571] is critical of more reason-oriented systems.

> The idea is that the reactive system handles the real-time issues of being embedded in the world, while the deliberative system does the "hard" stuff traditionally imagined to be handled by an Artificial Intelligence system. I think that these approaches are suffering from the well-known "horizon effect"—they have bought a little better performance in their overall system with their reactive component, but they have simply pushed the limitations of the reasoning system a bit further into the future.

Diverging opinions of just this sort prompted Kirsh [6, p. 161] to remark:

> It is an open question just where to draw the line between situationally determined activity—activity that can be initiated and regulated by smart perception action systems—and activity that requires thought, language-like conceptualization, and internal search.

More recently, systems like InteRRaP [38] have been proposed, which resemble a subsumption architecture of reactive planners. InteRRaP has three layers: a reactive behavior-based lower layer, a middle local-planning layer, and an upper cooperative-planning layer; each layer has a world representation at a level of abstraction.

11.3 Logical Reasoning

In more recent times, declarativist representations have been dominated by the logicist school of thought with artificial intelligence [39, p. viii]:

> We claim that AI deals mainly with the problem of representing and using declarative knowledge (as opposed to procedural) knowledge. Declarative knowledge is

8. This dualist approach reflects our rational account of ourselves. Reasoning is intuitively commendable when deliberating over the purchase of a house but seems less valuable when one has accidentally placed a hand on a damaging hot plate. In such circumstances, one does not consciously question the logical consequences of a cooked hand—one instinctively reacts!

> the kind that is expressed as sentences, and AI needs a language in which to state these sentences. Because the languages in which this knowledge is originally captured (natural languages such as English) are not suitable for computer representations, some other language with the appropriate properties must be used. It turns out, we think, that the appropriate properties include at least those that have been uppermost in the minds of logicians in their development of logical languages such as the predicate calculus.

The logicist view has become prominent within artificial intelligence [40, p. 132]:

> One can look at AAAI Proceedings and IJCAI Proceedings over the past several years to see the trend: more and more articles are not even readable unless one is familiar with concepts like logical formulas, inference, non-monotonic reasoning, models, quantification, unification, metalevel reasoning and the like. Knowledge about these subjects is part of the technical apparatus that all AI researchers need to have.

In the early days of artificial intelligence, classical logics were automated through *theorem provers*. Data structures syntactically isomorphic to the formal language expressions of the logic were employed to express formal theories, while a collection of computational procedures were employed to automatically manipulate the data structures in accordance with the syntactic manipulations specified by the logic's deductive-inference relation. The consequence was a machine capable of conducting automated deductive inference. Automated deduction was dominated by the resolution method of Robinson [41], which was founded upon just two rules of inference, *factoring* and *resolution*. Coding the deductive relation required coding two important procedures over and above the actual factoring and resolution-inference rules:

1. A *clausification* procedure reduced any given first-order formula to its Skolemized conjunctive normal form; expressed the atomic and negated atomic elements as primitives, called *literals*; expressed the disjunctions of the atomic and negated atomic elements as sets of literals, called *clauses*; and expressed the conjunction of these disjunctions, being the complete formula, as a set of clauses.
2. A *unification* procedure computes the *most general unifier* of two literals, with a *unifier* being a substitutional environment which renders the two literals identical.

The two rules of inference operate by finding the most general unifiers for literals in clauses.

Mechanisms for allowing knowledge to guide the search process subsequently led to rule based systems like PROLOG, with the representation language designed to exploit the underlying shared library definition (SLD)-resolution control mechanism. Unconditional knowledge is coded as facts, while conditional knowledge is coded as rules. The nature of logicist systems has diversified beyond simple rule-based systems. Sophisticated general theorem provers currently include first-order logic theorem provers like Otter, ordered semantic hyperlinking (OSHL), and Gandalf, and higher-order logic theorem provers like Isabelle and high order logic (HOL).

11.3.1 Decidability and Complexity

The logicist paradigm views computational systems as devices that compute logical inference. A principal advantage of this approach over simple rule-based systems is that it allows the system engineer to assess a system by assessing the capabilities of the logic it defines. The engineer can consider whether the system is sound, complete, decidable, and, if decidable, its computational complexity.

To address the issues of soundness and completeness, one has to consider both syntactic and semantic ways of finding conclusions, given the facts. A *proof* is a syntactic notion: Given the language, the symbols employed, and the set of axioms, what can be mechanically derived from the facts? An interpretation, or *model*, is a semantic notion: it associates nonlogical symbols of the language with terms of the domain, and we say that an interpretation satisfies the sentence if it makes it true. An interpretation that makes a sentence true is called its model, and a sentence is *valid* if it is true under all interpretations (every interpretation is its model). A logic is *sound* if every sentence that can be proven is true. A logic is *complete* if every sentence that is true can be proven (a proof of this sentence can be found). Given a logical system, one usually prefers that the system be both sound and complete.

A set of sentences is *decidable* if there is a procedure that, given a sentence, will decide whether that sentence is in the set of sentences. A theory (logical system) is decidable if the set of its true sentences is decidable. A theory is *finitely axiomatizable* if its true sentences are consequences of a finite set of sentences. It is an important fact that if a theory is finitely axiomatizable and complete, then it is decidable [42].

Given a logic, the logic is often required to be not only decidable also *tractable*—not only the logic, in principle, determine whether any given sentence is true but these determinations can be obtained reasonably quickly, using a mechanical procedure (implemented on a computer). This procedure, or algorithm, is expected to solve the following problem: Given the logic and a sentence, is the sentence true? It is crucial not only that an answer can be produced for any sentence (instance of the problem) but also that the answer is produced reasonably quickly. If an algorithm has a polynomial time complexity (the time required to produce an answer is a polynomial function of the length of input), then it is considered tractable; if an algorithm has an exponential time complexity, then it is not considered good; if no polynomial time complexity algorithm has been found for a problem, then the problem is considered intractable. The theory of *NP-completeness* [43] explores the tractability of problems. It classifies many problems as NP-complete, and such problems are generally believed to be intractable, but whether NP-complete problems are indeed intractable (whether it is certain that no polynomial time complexity algorithms can be found for them) remains an open question. However, NP-completeness of a given problem indicates that it is as difficult to solve efficiently (to find a polynomial time algorithm for it) as all of the problems in the NP-complete class (all of these problems are difficult and are believed to be intractable).

11.3.2 Knowledge Representation

The main activity in the area of knowledge representation and knowledge reasoning (KR) is to represent a domain of interest formally and to reason automatically about the domain using the employed representation. To do this, one needs to build a formal theory of the domain. From the logicist's standpoint, building formal theories (of domains) is what KR is about.

Depending on the domain, one might need a formal theory of time, space, and space-time; a formal theory of aircraft, weapon systems, radars, and military assets; a formal theory of agents; and so forth. Clearly, one formal theory might be built on top of another one; for instance, a formal theory of aircraft would need to be built on top of a formal theory of space-time (or formal theories of space and time).

Such formal theories could be seen as "ontologies," in a deep sense, of the domains. A formal theory of aircraft can be seen as an ontology of aircraft, a theory that explains what aircraft are, what they do, what its relevant properties and relations are, and how to represent and reason about them. Such ontologies, as formal theories, might differ wildly between domains in the sense that one ontology might employ a set of conceptual (ontological) relations very different from those employed by another ontology. For instance, a formal theory of space-time might employ the following conceptual relations: connects, north-of, far-away, between, before, during, and so forth (note that such relations would not be employed in formal theories in which space and time is of no concern). The main difficulty in building such ontologies is to decide what primitives (primitive relations) are needed, what other relations can be defined using the primitives, and what the logical and computational properties (soundness, completeness, decidability, complexity) of the ontologies or the formal theories are.

As many formal theories of domains need to be built on top of formal theories of space-time, or processes, some formal theories of space-time will be described.

An important class of ontologies, formal theories in a shallow sense, can be singled out: taxonomies, or formal theories not much more involved than taxonomies—let's call them "ontologies as taxonomies." Such ontologies seldom employ other conceptual relations than subsumption (kind-of, subclass-superclass) and, possibly, mereonomic relations (part-of, part-whole). In practice, building such ontologies often amounts to building a hierarchy of classes (of objects of the domain). This is the sense of the word "ontology" as employed when one talks about ontologies built in XML, resource description framework (RDF), DAML + OIL, and so forth, What is important about such ontologies is whether they are built using "only" XML, using both XML and RDF, or using a logic-based extension of RDF such as OIL—this is the question of how syntactic or semantic the approach taken is. It seems that taking a purely syntactic approach (XML only) is highly inappropriate. RDF provides some semantics, but is not expressive enough. Therefore, one should really focus on semantic approaches, approaches that extend RDF by providing logical formalisms equipped with model-theoretic semantics—after all, ontologies are supposed to capture (restrict) the meaning of terms, and any purely syntactic approach fails to do this.

One good example of ontology frameworks equipped with formal semantics is OIL, a framework built on description logics. The OIL (DAML + OIL, OWL) approach and some alternatives will be described.

11.3.3 Ontologies as Taxonomies

Ontology languages or frameworks based on logical, formal theories are to be preferred as they provide formal semantics, and questions of logical and computational properties of such frameworks can be properly addressed.

Three example frameworks are KIF/Ontolingua, F-logic/OntoEdit, and description logics/OIL.

11.3.3.1 Knowledge Interchange Format and First-Order Logic

The Ontolingua ontology environment is based on the Knowledge Interchange Format (KIF) as its knowledge representation formalism. The main problem with KIF is that it is based on first-order logic and is therefore undecidable (KIF even extends beyond first-order logic; for instance, it adds a form of reification mechanism, which allows the treatment of statements of the language as objects in their own right, thereby making it possible to express statements over these statements).

To make KIF usable, one needs to restrict KIF and consider sublanguages of KIF (so called small KIFs) that would have better computational properties than KIF (the "big KIF"). In the KIF community, this is called a "levels of conformance" problem, though to date no consensus on this has been reached. A recent proposal [44] suggests the following:

1. There will be two specs, one for the "full" KIF (with sorts, namespaces, and anything else people want), and one for Meta-KIF.
2. Each system defines syntactic conformance by providing in Meta-KIF the (subset of the) KIF syntax it supports.
3. Each system defines semantic conformance in terms of soundness, completeness, and decidability.

11.3.3.2 OntoEdit and F-Logic

OntoEdit, an ontology engineering environment, is based on the logical formalism of F-logic, or frame logic. The current version of OntoEdit supports F-logic, resource description framework semantics (RDFS), and OIL; it has an interface to the Karlsruher F-Logic Inference Engine, and soon an access to FaCT (a description-logic-based reasoner employed in OIL/OntoEdit) will be provided.

OntoEdit provides the following facilities [45]:

- The tool allows the user to edit a hierarchy of concepts or classes. These concepts may be abstract or concrete, which indicates whether or not making direct instances of the concept is allowed. A concept may have several names, which is essentially a way to define synonyms for that concept.

- Concepts may participate in binary-type relations. Attributes of concepts are also considered to be relations. For this purpose, built-in types, such as STRING, INTEGER and BOOLEAN, are introduced. Relations can also be composed based on other relations.
- Relations can be ordered in a hierarchy, which allows for inheritance or refinement of characteristics of relations. For example, the relation "hasRoom (x, y)" between a hotel and a room may be refined into the relation "hasDoubleRoom (x, y)" between a hotel and a double room, where "DoubleRoom" is a concept inheriting from the "Room" concept. Relations are refined by imposing restrictions on values, such as in the specified example, or on cardinality, for example, by narrowing a one-to-many relation down to a one-to-one relation.
- Each concept and relation can be documented explicitly within the ontology. This is especially important when exchanging ontologies. Metadata on the ontology, such as the creator and the date of last modification, can also be stored within the ontology. This fixed set of attributes consists of the dublic core attributes, as well as some ontology-specific attributes.
- Transformation modules can be linked into the system, which allow translation of the ontology from its own general, XML-based storage format to a more specific format. Currently, an F-Logic transformation module is available, and work on an RDF module is underway.
- An analysis of the ontology language employed in OntoEdit would amount to a discussion of the F-Logic formalism—this is beyond the intended scope of this book. Comparing F-logic to the description logics employed in OIL, F-logic is more expressive but harder to reason with.

11.3.3.3 OIL and Description Logics SHF, SHIQ, SHOQ(Dn)

OIL (the Ontology Inference Layer or Ontology Interchange Language) and its implementation in OilEd, a simple OIL editor, is a description-logic-based framework. Although a framelike language is provided as a front end, the ontologies one builds are really static Hayard free (SHF) or SHIQ knowledge bases. Therefore, it is appropriate to see OIL languages as description-logic languages SHF and SHIQ. Another emerging language is SHOQ(Dn). A short discussion on SHF, SHIQ, and SHOQ(Dn) is provided below.

For detailed discussions on OIL and its connection with DAML, see [46, 47]. The first version of the combined DAML and OIL language was called DAML-OIL, then renamed to DAML + OIL (but a description-logic language has not yet been worked out for DAML + OIL). Then, OWL, the Ontology Web Language, based on DAML + OIL, was proposed. The OWL language can be used to formalize a domain by defining classes and properties of those classes, to define individuals and assert properties about them, and to reason about these classes and individuals as specified by the semantics of the language [48].

The logic implemented in FaCT, the reasoner for OIL, is based on ALC_R+, an extension of ALC to include transitive roles [49]. For conciseness, this logic has been called S (due to its relationship with the proposition multimodal logic S4(m) [50]). SHF extends S with a hierarchy of roles and functional roles (attributes), while SHIQ adds inverse roles and fully qualified number restrictions. The SHIQ reasoner is of particular interest, both from a theoretical and a practical viewpoint. Adding inverse roles to SHF (resulting in SHIF) leads to the loss of the finite model property, and this has necessitated the development of a more sophisticated double dynamic blocking strategy that allows the algorithm to find finite representations of infinite models, while still guaranteeing termination [51]. Moreover, when SHIF is generalized to SHIQ, it is necessary to restrict the use of transitive roles in number restrictions in order to maintain decidability [52]. SHIQ is also of great practical interest as it is powerful enough to encode the logic DLR and can thus be used for reasoning about a wide range of conceptual data models (see, for example, [53] and extended entity-relationship (EER) schemas [54].

SHOQ(D) is a description logic derived from SHIQ, by giving up inverse roles (the "I" in SHIQ) but adding "individuals" (the "O") and concrete domains (the letter "D"). A tableaux algorithm for SHOQ(D) has been provided, but it has not yet been implemented. The ability to SHOQ(D) to build ontologies seems still to be in the relatively distant future. It seems appropriate to use SHF as it is easier to compute SHF ontologies than SHIQ ontologies and only to employ SHIQ when the additional expressibility (inverse roles, "I;" qualified number restrictions, "Q") is required.

Constructors for SHIQ, a logic employed by DAML + OIL, include:

- intersectionOf;
- unionOf;
- complementOf;
- toClass (\forall P.C);
- hasClass (\exists P.C);
- maxCardQ (<n P.C);
- minCardQ (>n P.C).

The set of constructors determines the expressibility of the logic, as well as the computational costs of reasoning with it.

Regarding complexity, both SHF and SHIQ are decidable but not (theoretically) tractable: SHF is in ExpTime (consisting of all languages with time complexity bounded above $2^{p(n)}$ for some polynomial p of the input length n), but the implementation is highly optimized and behaves well in practice. The implementation of SHIQ is an extension of the implementation of SHF, but the employed optimization techniques are not guaranteed to work for SHIQ. The logic SHOQ(D) has not been implemented yet.

11.3.4 Formal Theories of Space-Time (Ontologies as Formal Theories)

At the ontological or metaphysical level, the world consists of matter distributed in space and changing over time. It seems appropriate to have identities as "histories," or processes with "temporal parts," as fragments of space-time. To build a formal theory of, say, ships, one would build it on top of a formal theory of space-time; a theory of space-time would be used to describe (and reason about) processes, which are spatiotemporal fragments of (the matter of) the universe [55]. Then, ships would be properly individuated processes: A ship is a process (a spatiotemporal chunk of matter) that can stay afloat, move, communicate, shoot, approach, attack, and so forth. It is thus clear that a theory of processes would be strongly connected to a theory of space-time—it seems appropriate to consider a theory of processes that are spatiotemporal processes. Many formal theories of interest (e.g., theories of fighting ships, aircraft, military assets or platforms, and military operations) would be built on top of a formal theory of spatiotemporal processes.

It seems clear that one needs to consider formal theories of time and space before building a formal theory of space-time. This is not so because a theory of space-time should be a result of "combining" a theory of space and a theory of time—quite the opposite: a theory of space-time should be the most primitive, and a theory of space and a theory of time should be derivable from the theory of space-time. However, when building a theory of space-time, one needs to decide upon an appropriate set of "primitives," to come up with a reasonable set of primitives that one might consider primitives employed in theories of space and theories of time. For instance, there can be a theory of space that employs "overlap" as its primitive, and there can be a theory of time that employs "overlap" as its primitive, indicating that maybe "overlap" can be used as a primitive in a theory of space-time. But that does not mean that one will combine "spatial overlap" with "temporal overlap" to build "spatiotemporal overlap"—quite the opposite: "spatiotemporal overlap" is the most "basic" primitive, in the sense that "spatial overlap" and "temporal overlap" should be derivable from it.

We therefore consider some theories of time and some theories of space, comment on proposals for theories of space-time, suggest a theory of space-time, and also address theories of processes, indicating how a (nonspatiotemporal) process theory should be extended to incorporate space-time. A comment (not much more than that) is then made about a theory of ships—a theory of ships would be one of a bunch of formal theories built on top of theories of spatiotemporal processes. Indeed, Lambert [56] proposes a hierarchy of classes of formal theories encompassing metaphysical, physical, functional, intentional, and social levels.

11.3.4.1 Formal Theories of Time

We limit ourselves to interval structures and consider the following three theories:

- van Benthem theory (I, <, subset), where I is a set of intervals, "<" is a precedence relation (the "before" relation), and "subset" is the inclusion relation over I;
- Allen and Hayes theory (J, ||), where J is a set of temporal intervals and "||" is a binary relation of meeting of intervals (note that the 13 Allen's relations on intervals [before, starts, during] can be defined in terms of the meet relation);

- Tsang theory $(G, <, \text{overlap})$, where G is a set of intervals, “$<$” is a precedence relation (the “before” relation), and “overlap” is an overlapping relation on G.

For a discussion of how these theories are related to each other, see [57].

From an IF perspective, Tsang theory seems interesting as it takes as primitives the precedence relation “$<$” (a temporal “orientation” relation) and the overlap relation (a temporal “connection” relation). From a short discussion on theories of space, it will be evident that “connection” and “orientation” are primitives of interest—this means that a theory of space-time might be based on the primitives of “connection” (or “overlap”) and “orientation.”

11.3.4.2 Formal Theories of Space

A most successful proposal for qualitative spatial reasoning is a formal theory of space known as region connection calculus (RCC) [58]. The RCC theory is based on a single primitive of “connection” between spatial regions. Other relations (such as the eight relations of RCC-8) can be defined in terms of the connection relation. Certainly, it seems appropriate to consider RCC when building a “connection”-based theory of spatiotemporal regions.

Qualitative approaches to spatial “orientation” need to be taken into account if a theory of space-time is not only to “generalize” a theory of spatial connection but also a temporal theory such as the Tsang theory. This indicates that a simple theory of orientation needs to be considered in order to propose an orientation (or orientation plus connection or overlap) theory of space-time.

11.3.4.3 Formal Theories of Space-Time

There are very few proposals for a theory of space-time. An example is the work of Muller [59]. His theory is a theory of motion, taking space-time histories of objects as primitive entities.

It seems that a simple theory of space-time should be based on such primitives as “orientation” and “overlap” (or “connection”). Such a theory would nicely correspond to the temporal Tsang theory and a spatial theory of RCC extended with “orientation.” However, there are two important notes. First, orientation by itself is hardly enough: One employs orientation to locate entities, but to (even qualitatively) locate them, one needs both orientation and distance; therefore, one should add “distance” to “orientation” and “overlap.” Second, there are many discouraging results regarding the decidability and complexity of spatial, temporal, and spatiotemporal theories (see [60, 61]). It is clearly a very difficult task to come up with a reasonable theory of space-time that is tractable, or even just decidable. Consider the following quote [61]:

> Multi-dimensional modal logics are not easy to deal with, and we need to be careful in constructing effective and expressive spatio-temporal formalisms. For example, the straightforward attack on the problem by means of using the Cartesian products of frames for S4 and the flow of time $(N, <)$ (or any other infinite linear order) has not brought any result yet: whether the logic of such 2-dimensional frames is decidable remains one of the challenging open problems in the field.

11.3.4.4 Formal Theories of Processes

Lambert [55] proposes a theory of processes based on a structure $(P; =, <, +, ., -, 0, 1)$, where

P is a set of processes (fragments of the universe);

$=$ is an equality relation ($p = q$ means that p and q are the same processes);

$<$ is a fragmentation (part-of) relation;

$+$ is a "joint" (union) operation on processes;

. is a "meet" (intersection) on processes;

$-$ is a complementation operation;

0 and 1 are the empty process—the bottom element of the ordered structure $(P, <)$—and the universe (top) process.

The structure $(P; =, <, +, ., -, 0, 1)$ extends a purely mereological structure $(P; <, 0, 1)$; however, "$=$," "$+$," "$.$," "$-$," 0, and 1 can all be defined in terms of "$<$." Furthermore, an axiomatization can be given by providing the axioms of identity, fragmentation, universe, unity, and separation.

A mereological theory of processes can then be extended to incorporate the concepts of time and space by accommodating "times" and "spaces" as processes. For instance, the time "year 2001" is the fragment of the whole universe that started at the beginning of the year 2001 and ended at the end of the year 2001. Seeing purely temporal and spatial entities as processes conceptually unifies space, time, and processes. The next step is to add such primitives (taken from a formal theory of space-time) as "overlap," "orientation," and "distance."

11.4 Rule-Based Reasoning

Most practitioners accept that symbolic representations are useful. For instance, [62] argues,

> The situationalists are attacking the very idea of knowledge representation—the notion that cognitive agents think about their environments, in large part, by manipulating internal representations of the worlds they inhabit. Let us be frank: we think the representational hypothesis is a great idea. The reasons for being so positive are well documented, but we have two main justifications for our enthusiasm. First, it accounts for much that is otherwise completely puzzling about how cognition could happen in the physical world; second, it allows experiments and makes empirical predictions, which have so far largely been confirmed. But one needs to understand the key word "representation" in a sufficiently broad fashion.
>
> Internal representations might not be consciously available to introspection, might utilize ontological frameworks that are determined by social or other contexts, might be involved with (and have their use involved in) social practices or any other kind of human activity, and might be involved in perceptual or motor skills at any cognitive level. None of these are in any way at variance with the representationalist hypothesis.

Acknowledging the utility of symbolic representations exposes another previous debate within artificial intelligence. In the mid to late 1960s, the emphasis in AI shifted from search to representation [63, p. 9][9]:

> The most central idea of the pre-1962 period was that of finding heuristic devices to control the breadth of a trial-and-error search. A close second preoccupation was with finding effective techniques for learning. In the post-1962 era, the concern became less with "learning" and more with the problem of representation of knowledge (however acquired) and with the related problem of breaking through the formality and narrowness of the older systems. The problem of heuristic search efficiency remains as an underlying constraint, but it is no longer the problem one thinks about, for we are now immersed in more sophisticated subproblems, e.g. the representation and modification of plans.

This forged a number of representation schemes, including *production system, logical, procedural, semantic network*, and *frame-based representation schemes*. Whether representation ought to take the declarativist's data form of "knowing that" or the proceduralist's procedure form of "knowing how" became a matter of considerable debate in the early 1970s. The first part of [65] outlines the virtues of each camp's case. The threads of this debate extend back to the contrasting viewpoints espoused by Minsky and McCarthy at the Dartmouth Conference.

Declarative rule-based systems, based primarily on production-system and logical-representation schemes, became the dominant approach to encoding human knowledge during the period. Knowledge-based systems, also called expert systems, have been defined in terms of function and structure by well-known authors. Many early definitions assume rule-based reasoning. Edward Feigenbaum, the leading advocate of knowledge-based systems, a professor at Stanford and chief scientist of the U.S. Air Force from 1994 to 1997, gave the following definition [66]:

> Expert system is an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution. The knowledge of an expert system consists of facts and heuristics. The "facts" constitute a body of information that is widely shared, publicly available, and generally agreed upon by experts in the field.

As mentioned earlier, most early expert systems were rule-based systems. In these systems, the experts' knowledge was encoded in the form of associational "rules of thumb" (also referred to as heuristics) that mapped from observable features of the problem to conclusions. They had a simple control structure and a uniform representation of knowledge. It was recognized that these first generation expert systems were limited in their knowledge-representation capabilities and implicitly combined knowledge of different natures ("what," "how," and "why"). This led to several problems related to knowledge acquisition, explanation, brittleness, and maintainability.

9. The year 1962 is supposedly singled out because it dates the material of Feigenbaum and Feldman [64], which summarized the search phase at that time. There is perhaps also a tinge of Carnegie–MIT rivalry involved in choosing what is prima facie a premature date, at least from the discipline perspective.

The main components of a rule-based expert system are the knowledge base, inference engine, working memory, and user interface (see Figure 11.9). More elaborate expert systems also include a knowledge-acquisition facility and some explanation facilities.

11.4.1 The Knowledge Base

The knowledge base contains the expert system's knowledge represented as a set of rules and facts. Rules are used to represent heuristics, or "rules of thumb," that specify a set of actions to be performed for a given situation. Various formalisms can be used to express rules. A production rule is a method of knowledge representation characterized by an IF "*condition* THEN *action*" format. The condition part of a rule is a series of patterns that specify the facts (or data) that cause the rule to be applicable. The condition may be a compound of the Boolean connectives *and, or*, and *not*. The action part of a rule is the set of actions to be executed when the rule is applicable. An action can affect the value of working-memory variables, take some real-world action, or potentially do other things, such as stop the production system.

Certainty factors are sometimes used to represent the confidence one has that a fact is true or a rule is valid.

11.4.2 The Working Memory

The working memory contains all the information about the problem that is either supplied by the user or inferred by the system through the activation, or "firing," of rules.

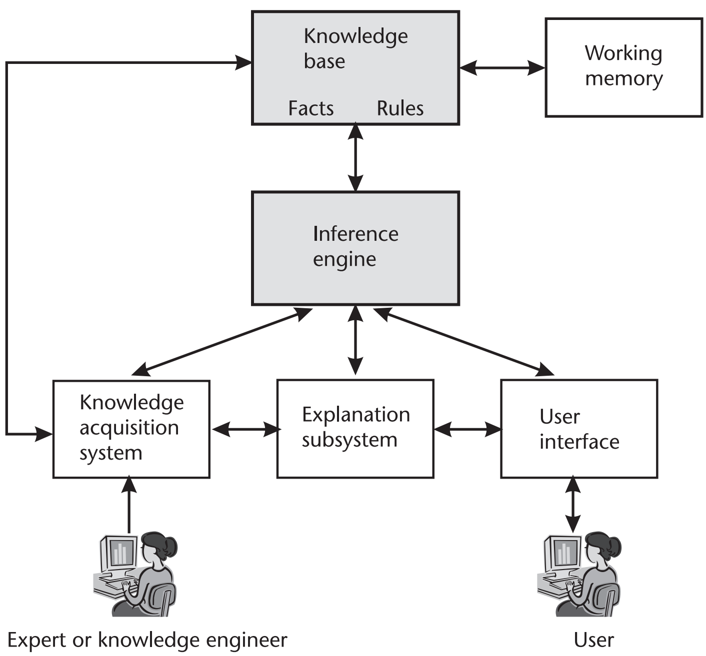


Figure 11.9 Components of a rule-based expert system.

11.4.3 The Inference Engine

The inference engine is the component of an expert system responsible for drawing new conclusions from both the current facts in the working memory and the rules contained in the knowledge base. It automatically matches facts against patterns and determines which rules are applicable. The process of matching facts to patterns is called pattern matching. The inference engine selects a rule and the actions of the selected rule are executed (which may affect the list of applicable rules by adding or removing facts). The inference engine then selects another rule and executes its actions. This process continues until no applicable rules remain. The inference engine also contains methods related to various matching and conflict-resolution strategies. Conflict resolution is a strategy for determining which rule is "fired" when the conditions of several rules are satisfied. This choice will influence the movement in the problem space and the amount of search to be made (i.e., the effectiveness of the system as a problem solver).

The basic execution cycle of rule-based systems consists of:

1. Determining which rules match facts in the working memory;
2. Choosing a rule to apply (conflict resolution);
3. Applying the rule, which will change the working memory;
4. Going to 1.

11.4.4 Reasoning Modes

Forward-chaining (or data-driven chaining) and backward-chaining (or goal-directed chaining) are the two main methods of running a production system.

- *Forward-chaining reasoning* progresses "forward" from the initial data toward the final conclusion. It is used to find the solution by starting with an assumption and working toward a final goal. According to the production system cycle, the conditional parts of the rules are checked against the content of the working memory, then the action parts of the suitable rules are fired.
- *Backward-chaining reasoning* starts with a conclusion and determines the data needed to reach such a conclusion. It is used when there is a goal to prove, and there is an attempt to establish the premise of that goal. It uses a process to find the solution by searching backwards from the solution toward the initial conditions, thus verifying the specified goal.

11.4.5 RETE Algorithm

The RETE algorithm is a well-known pattern-matching algorithm developed by Charles L. Forgy at Carnegie-Mellon University in the 1970s. It is very good at handling large numbers of rules efficiently. Forgy published an article about the algorithm in 1982 [67].

This algorithm is intended to improve the speed of forward-chained rule systems by limiting the effort required to recompute the conflict set after a rule is fired. It comes from the observations that: (1) the firing of a rule usually changes only a few facts, each of which affects only a few rules, and (2) the same pattern often appears on the left-hand side (LHS) of more than one rule.

The obvious implementation for an inference engine is to keep a list of the rules and to continuously cycle through the list, checking each rule's LHS against the knowledge base and executing the right-hand side (RHS) of any rules that apply. This is inefficient because most of the tests made on each cycle will have the same results as during the previous iteration.

RETE is a compilation algorithm that transforms a rule set into a tree of interrelated nodes. The internal nodes represent tests appearing on the LHS (conditions) of the rules, and the leaves of the tree represent the RHS (actions) of the rules. RETE allows for sharing tests among rules and stores information about the object in memory that partially satisfies one or more rule conditions. This network of nodes processes facts that are being added to, or removed from, the knowledge base. RETE is used to implement event-driven programming more than backward-chaining and is especially quick to react when new information is added to the network. Only new facts are tested against any rule LHSs. Additionally, new facts are tested against only the rule LHSs to which they are most likely to be relevant. As a result, the computational complexity becomes linear in the size of the fact base.

In short, the RETE constructs the tree after parsing the system rules. Data elements enter the tree at an input node (a given condition) and follow certain branches (between nodes) to attain termination nodes, where the production can be launched. The data consequently encounters only rules for which it has to be tested.

The RETE is less efficient when it is applied to large amounts of data or to very rapidly changing data [68]. Forgy created a substantially revised algorithm named RETE II in the 1980s. The new algorithm is as good as the original RETE at handling large numbers of rules, but it is dramatically faster than the original algorithm when dealing with large amounts of data or rapidly changing data. Unfortunately, the RETE II algorithm is not available in the public domain. RETE II is exclusively available from Production Systems Technologies.

11.4.6 Examples of Expert Systems

DENDRAL was one of the first expert systems developed by Feigenbaum at Stanford University (1965). It establishes the structure of a molecule given its atomic formula and its spectrogram mass.

MYCIN is the best-known expert system, developed in the mid-1970s at Stanford by Bruce Buchanan and Edward Shortliffe, to diagnose infectious blood diseases and prescribe antibiotic treatment. It then reached human-level precision. The MYCIN program was important to the development of artificial intelligence because it provided clear indication that the techniques being developed would be of practical importance. MYCIN is an example of a backward-chaining approach (i.e., it works backwards from goals to given data).

The following is a MYCIN rule example:

IF the infection is bacteremia
& the site of the culture is one of the sterile sites
& the suspected portal of entry is the GI tract
THEN there is suggestive evidence (prob = 70%)
that the identity of the organism is bacteroides.

11.4.7 Expert-System Shells

Expert-system development environments, also called expert-system shells, have been designed to facilitate the building of complex expert systems. The development process is reduced to model the essential problem-solving knowledge and to write it down with the knowledge-representation formalism provided by the shell.

The C Language Integrated Production System (CLIPS) is a well-known productive-development and -delivery expert-system tool that was developed at NASA's Johnson Space Center in the mid-1980s to facilitate putting expert systems to work. The Java Expert System Shell (JESS) is a Java version of CLIPS. An interesting aspect of JESS is that it facilitates the integration of an expert system with a Java application through the use of Java Beans. Java Beans objects that are modified in an application can automatically trigger rules that match these objects.

11.5 Case-Based Reasoning

The rule-based approach assumes that systems of interest operate in accordance with principles that can be expressed by rules. Case-based reasoning challenges this approach to software engineering. Rules are formulated by people attempting to explain systems of interest, and the people understand those systems through experience, not rules [69, p. 15]:

> Human experts are not systems of rules, they are libraries of experiences. Further, these libraries are adaptable. When a new experience takes place, it isn't simply added to a data base of prior experiences. Most experiences are like others that have come before. A new experience relates to, modifies, replaces, amplifies, or otherwise perturbs many of the extant prior experiences.

This leads to an experiential problem-solving paradigm [69, p. 25]:

> The basic idea of case-based reasoning is simple:
>
> A case-based reasoner solves new problems by adapting solutions that were used to solve old problems.
>
> This differs from rule-based reasoning, which solves problems by chaining rules of inference together. A case-based reasoner
>
> - finds those cases in memory that solved problems similar to the current problem, and
> - adapts the previous solution or solutions to fit the current problem, taking into account any difference between the current and previous situations.

Aamodt and Plaza [70] describe the four basic steps of the case-based cycle, shown in Figure 11.10.

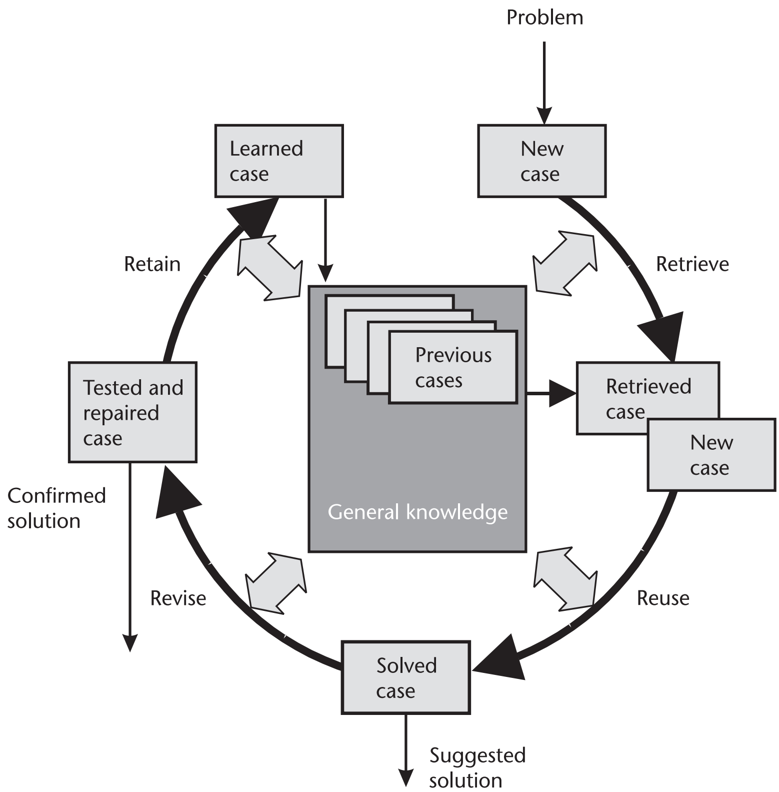


Figure 11.10 The CBR cycle. (*After:* [70].)

1. Retrieve the most similar case or cases.
2. Reuse the information and knowledge in that case to solve the problem.
3. Revise the proposed solution, based on the extent of the solution's success.
4. Retain the parts of this experience likely to be useful for future problem solving.

Figure 11.11 illustrates the interaction of various components used to create a CBR system in CHEF [69, p. 181].

CBR attempts to model human experience of systems rather than capture the logical principles or rules by which those systems operate. Proponents of CBR contend that there is a trade-off between these approaches [69, p. 26]:

> These trade-offs hold in general between rule-based and case-based reasoning. A rule-based system will be flexible and produce nearly optimal answers, but it will be slow and prone to error. A case-based system will be restricted to variations on known situations and produce approximate answers, but it will be quick and its answers will be grounded in actual experience.

CBR has been applied as an approach to level 4 adaptation of a level 2 fusion process. This involved the selection and adaptation of a collection of ATTITUDE [71] routines for assessing aircraft movement [72].

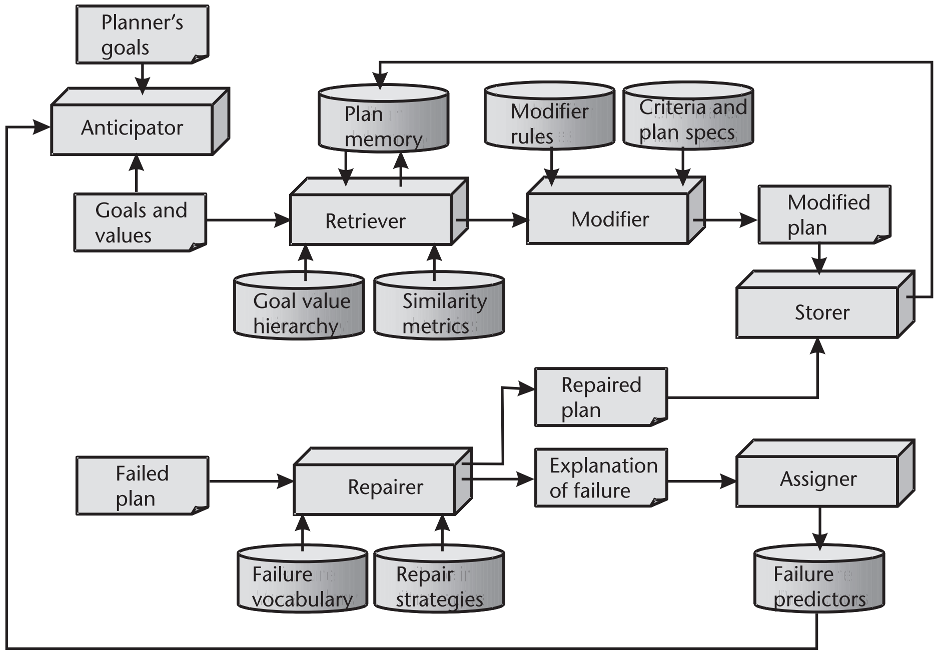


Figure 11.11 CHEF CBR system. (*After:* [69, p. 181].)

References

[1] McCarthy, J., "Programs with Commonsense," in R. J. Brachman and H. J. Levesque, (eds.), *Readings in Knowledge Representation*, Los Altos, CA: Morgan Kaufmann Publishers, 1968, pp. 299–307.

[2] Chapman, D. H., "Planning for Conjunctive Goals," *Artificial Intelligence*, Vol. 32, 1987, pp. 333–377.

[3] Brooks, R. A., "A Robust Layered Control System for a Mobile Robot," *IEEE Journal of Robotics and Automation*, Vol. RA-2, 1986, pp. 14–23.

[4] Brooks, R. A., "Intelligence Without Representation," *Artificial Intelligence*, Vol. 47, 1991, pp. 139–160.

[5] Brooks, R. A., "Intelligence Without Reason," *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, Sydney, Australia, August 1991, pp. 569–595.

[6] Kirsch, D., "Today the Earwig, Tomorrow Man?" *Artificial Intelligence*, Vol. 47, 1991, pp. 161–184.

[7] Hecht-Nielsen, R., *Neurocomputing*, Reading, MA: Addison-Wesley, 1990.

[8] Stubberud, A. R., and H. Wabgaonkar, "A Neural Network–Based Non-Linear Control Technique," in C. T. Leondes, (ed.), *Advances in Control*, New York: Academic Press, 1992.

[9] White, D., and D. Sofge, *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*, New York: Van Nostrand Reinhold, 1992.

[10] Fogel, D. B., *Evolutionary Computation Toward a New Philosophy of Machine Intelligence*, New York: IEEE Press, 1995.

[11] Singhal, S., and L. Wu, "Training Multilayer Perceptrons with the Extended Kalman Algorithm," in D. S. Touretzky, (ed.), *Advances in Neural Information Processing Systems*, San Mateo, CA: Morgan Kaufmann, 1989, pp. 133–140.

[12] Turing, A. M., "On Computable Numbers with an Application to the Entscheidungsproblem," in M. Davis, (ed.), *The Undecidable*, New York: Raven Press Books, 1937.

[13] Turing, A. M., "Intelligent Machinery," *Machine Intelligence*, Vol. 5, 1947, pp. 3–23.

[14] Turing, A. M., "Computing Machinery and Intelligence," in E. Feigenbaum and J. Feldman, (eds.), *Computers and Thought*, New York: McGraw-Hill, 1950, pp. 11–35.

[15] Georgeff, M. P., and Lansky, A. L., "Reactive Reasoning and Planning," *Proceedings of AAAI-87*, Seattle, WA, 1987, pp. 677–682.

[16] Payton, D., "Exploiting Map Plans as Resources for Action," source unknown, 1990, pp. 311–320.

[17] Ginsberg, M. L., "Universal Planning: An (Almost) Universally Bad Idea," *AI Magazine*, Vol. 10, No. 4, 1989, pp. 40–44.

[18] Schoppers, M. L., "Universal Plans for Reactive Robots," *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, San Francisco, CA, 1987, pp. 1039–1046.

[19] Rosenschein, S. J., and L. P. Kaelbling, "The Synthesis of Digital Machines with Provable Epistemic Properties," *Proceeding of the 1986 Conference on Theoretical Aspects of Reasoning about Knowledge*, San Mateo, CA, 1986, pp. 83–98.

[20] Rosenschein, S. J., and L. P. Kaelbling, "Integrating Planning and Reactive Control," source unknown, 1990, pp. 359–366.

[21] Rosenschein, S. J., and L. P. Kaelbling, "A Situation View of Representation and Control," *Artificial Intelligence*, Vol. 73, 1995, pp. 149–173.

[22] Kaelbling, L. P., "An Architecture for Intelligent Reactive Systems," in M. P. Georgeff and A. L. Lansky, (eds.), *Reasoning about Actions and Plans*, San Mateo, CA: Morgan Kaufmann Publishers, 1987, pp. 395–410.

[23] Kaelbling, L. P., "Goals as Parallel Program Specifications," *Proceedings of the Seventh National Conference on Artificial Intelligence*, Vol. 9, 1988, pp. 60–65.

[24] Schoppers, M. J., "In Defense of Reaction Plans as Caches," *AI Magazine*, Vol. 10, No. 4, 1989, pp. 51–60.

[25] Schoppers, M. J., "The Use of Dynamics in an Intelligent Controller for a Space Faring Rescue Robot," *Artificial Intelligence*, Vol. 73, 1995, pp. 175–230.

[26] Agre, P., and D. Chapman, "Pengi: An Implementation of a Theory of Activity," *Proceedings of the 6th National Conference on Artificial Intelligence*, Seattle, WA, 1987.

[27] Chapman, D., and P. E. Agre, "Abstract Reasoning as Emergent from Concrete Activity," in M. P. Georgeff and A. L. Lansky, (eds.), *Reasoning about Actions and Plans*, San Mateo, CA: Morgan Kaufmann Publishers, 1987, pp. 411–424.

[28] Firby, R. J., "An Investigation in Reactive Planning in Complex Domains," *Proceedings of AAAI-87*, Seattle, WA, 1987.

[29] Firby, R. J., "Task Directed Sensing," *SPIE: Sensor Fusion II: Human and Machine Strategies*, Vol. 198, 1989, pp. 480–489.

[30] Georgeff, M. P., and F. F. Ingrand, "Decision-Making in an Embedded Reasoning System," *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, Detroit, MI, 1989, pp. 972–978.

[31] Georgeff, M. P., and F. F. Ingrand, "Real-Time Reasoning: The Monitoring and Control of Spacecraft Systems," *Proceedings of the IEEE Applications of Artificial Intelligence*, Santa Barbara, CA, 1990.

[32] Hayes-Roth, B., "An Architecture for Adaptive Intelligent Systems," *Artificial Intelligence*, Vol. 72, 1995, pp. 329–365.

[33] Hayes-Roth, B., et al., "Intelligence Monitoring and Control," *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, San Mateo, CA, 1989, pp. 243–249.

[34] Hayes-Roth, B., "Architecture Foundations for Real-Time Performance in Intelligent Agents," *Real-Time Systems*, Vol. 2, 1990, pp. 99–125.

[35] Hayes-Roth, B., "A Blackboard Architecture for Control," *Artificial Intelligence*, Vol. 26, 1985, pp. 251–321.

[36] Fox, M. S., and S. Smith, "ISIS: A Knowledge-Based System for Factory Scheduling," *Expert Systems*, Vol. 1, 1984, pp. 25–49.

[37] Wooldridge, M., *Introduction to Multiagent Systems*, New York: John Wiley & Sons, 2002.

[38] Muller, J. P., and M. Pischel, "Modeling Interacting Agents in Dynamic Environments," *Proceedings of the 11th European Conference on Artificial Intelligence*, Amsterdam, the Netherlands, 1994, pp. 709–713.

[39] Genesereth, M. R., and N. J. Nilsson, *Logical Foundations of Artificial Intelligence*, San Mateo, CA: Morgan Kaufmann Publishers, 1987.

[40] Nilsson, N., "On Logical Foundations of Artificial Intelligence," *Artificial Intelligence*, 1989, pp. 132–133.

[41] Robinson, J. A., "A Machine-Oriented Logic Based on the Resolution Principle," *Journal of the ACM*, Vol. 12, No. 1, 1965, pp. 23–41.

[42] Enderton, H. B., *A Mathematical Introduction to Logic*, New York: Academic Press, 1972.

[43] Garey, M. R., and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Murray Hill, NJ: Bell Telephone Laboratories, 1979.

[44] Welty, C. A., "(KIF) Levels of Conformance," 2002, http://philebus.tamu.edu/pipermail/kif/2002-January/000972.html.

[45] OntoEdit Web site, http://www.ontoknowledge.org/tools/ontoedit.shtml.

[46] Ontoknowledge Organisation, OIL homepage, 2003, http://www.ontoknowledge.org/oil.

[47] DARPA, DAML Web site, 2003, http://www.daml.org.

[48] W3 Consortium, OWL Guide, 2002, http://www.w3.org/TR/2002/WD-owl-guide-20021104.

[49] Sattler, U., "A Concept Language Extended with Different Kinds of Transitive Roles," in G. Görz and S. Hölldobler, (eds.), *Lecture Notes in Artificial Intelligence*, Springer-Verlag, 1996, pp. 333–345.

[50] Schild, K., "A Correspondence Theory for Terminological Logics: Preliminary Report," *Proceedings of IJCAI-91*, Sydney, Australia, 1991, pp. 466–471.

[51] Horrocks, I., and U. Sattler, "A Description Logic with Transitive and Inverse Roles and Role Hierarchies." *Journal of Logic and Computation*, Vol. 9, No. 3, 1999, pp. 385–410.

[52] Horrocks, I., U. Sattler, and S. Tobies, "Practical Reasoning for Expressive Description Logics," in H. Ganzinger, D. McAllester and A. Voronkov, (eds.), *Lecture Notes in Artificial Intelligence*, Berlin: Springer-Verlag, 1999.

[53] "OIL Primitives," http://www.ontoknowledge.org/oil/TR/primitives.html, accessed November 11, 2006.

[54] Calvanese, D., et al., "Source Integration in Data Warehousing," *Proceedings of DEXA-98*, Vienna, Austria, August 24–29, 1998.

[55] Lambert, D. A., "Situations for Situation Awareness," *Fusion 2001*, Montreal, Canada, August 7–10, 2001.

[56] Lambert, D. A., "Grand Challenges of Information Fusion," *Proceedings of the 6th International Conference on Information Fusion* (*forthcoming*), Cairns, Australia, July 8–11, 2003.

[57] Hajnicz, E., *Time Structures: Formal Description and Algorithmic Representation*, New York: Springer, 1996.

[58] Randell, D. A., Z. Cui, and A.G. Cohn, "A Spatial Logic Based on Regions and Connection." *Proceedings of KR'92*, San Mateo, CA, 1992, pp. 165–176.

[59] Muller, P., "Topological Spatio-Temporal Reasoning and Representation," *Computational Intelligence*, Vol. 18, No. 3, 2002.

[60] Wolter, F., and M. Zakharyaschev, "Qualitative Spatio-Temporal Representation and Reasoning: A Computational Perspective," *Proceedings of the 7th Conference on Principles*

of Knowledge Representation and Reasoning, KR-2000, Breckenridge, CO, 2000, pp. 3–14.

[61] Bennett, B., et al., "Multi-Dimensional Model Logic as a Framework for Spatio-Temporal Reasoning," *Applied Intelligence*, Vol. 17, 2002, pp. 239–251.

[62] Hayes, P. J., K. M. Ford, and N. Agnew, "On Babies and Bathwater: A Cautionary Tale," *AI Magazine,* 1994, pp. 15–26.

[63] Minsky, M., *Semantic Information Processing,* Cambridge, MA: MIT Press, 1968.

[64] Feigenbaum, E., and J. Feldman, *Computer and Thought,* New York: McGraw-Hill, 1963.

[65] Winograd, T., "Frame Representations and the Declarative/Procedural Controversy," in R. J. Brachman and H. J. Levesque, (eds.), *Readings in Knowledge Representation*, pp. 357–370, Los Altos, CA: Morgan Kaufmann Publishers, 1975.

[66] Harmon, P., and D. King, *Expert Systems*, New York: John Wiley & Sons, 1985.

[67] Forgy, C. L., "RETE: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem," *AI Magazine*, Vol. 19, 1982, pp. 17–37.

[68] Production Systems Technologies Web site, 2001, http://www.pst.com/rete.htm.

[69] Riesbeck, C. K., and R. C. Schank, *Inside Case-Based Reasoning*, Hillsdale, NJ: Lawrence Erlbaum, 1989.

[70] Aamodt, A., and E. Plaza, "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches," *AI Communications*, Vol. 7, No. 1, 1994, pp. 39–59.

[71] Lambert, D. A., "Advisers with Attitude for Situation Awareness," *Proceedings of the 1999 Workshop on Defense Applications of Signal Processing*, La Salle, IL, 1999, pp. 113–118.

[72] Rutten, M., and D. A. Lambert, "Automated Situation Assessment Refinement," in L. Reznik and V. Kreinovich, (eds.), *Soft Computing in Measurement and Information Acquisition*, New York: Springer, 2003.

CHAPTER 12

Software Architectures

Steve Wark and Jean Roy

The preceding chapters have discussed the characteristics of the environments where information fusion can be applied and the requirements these impose on IF systems. We have also discussed aspects of the computational infrastructure relevant to the implementation of IF systems and introduced a number of middleware systems that can be used to provide heterogeneous, flexible, evolvable, distributed environments suitable for coalition C2 and IF systems. In addition, we introduced computational issues related to the information sources and user interfaces that are needed for these systems.

In Chapter 11, we introduced some of the computational issues associated with the knowledge-based and reasoning systems that are required to address higher level (levels 2 and 3) data fusion in the JDL model. In essence, this addressed the issues of how computational systems can be used to reason about a situation and its consequences.

This chapter discusses a number of software architectures that are being used in IF systems. This chapter does not address how IF components are connected together, or even the particular computational algorithms used for individual IF components; instead, it addresses the communication and interaction paradigms used in these systems to share knowledge, build workflows, and provide (perhaps emergent) IF capability.

The first two sections describe demonstration systems for information fusion and the architectures they employed. The following sections describe models and general architectures that can be used to develop IF systems.

12.1 Visual Data Fusion Computational Model

As discussed previously, the VDF model is a step to reformulating the JDL data-fusion model [1], using an enhanced framework for a "humancentric" process. It consists of interfaces, multimedia information sources, fusion processes, pattern display, and past learning.

The VDF model has four human-to-machine interfaces: linguistic, contextual, conceptual, and visual. Human input is primarily accomplished through spoken or textual language [2], providing computer-assisted fusion as a computation with words using fuzzy variables [3]. Linguistic input requests form a dynamic human-preference model [4], representing one's current problem-solving intentions; this intent is limited by problem context [5] and concept reference.

Pattern sets represent abstract image objects [6] and are primary agents of "solution" information transfer from machine to human in VDF. Patterns enhance understanding by assisting concentration on essentials, while suppressing the less relevant. They represent information abstraction or amalgamation [7], viewpoints [8], or general impressions [6]. Pattern-set members can be made relevant to certain phenomena within conceptual contexts by selective processing. As visual features [9] based on complex perceptual relations, the "specific meaning" of some phenomena to an individual is more easily perceived. For example, a human can match patterns in one frame to another, relating concepts or viewpoints through time. Patterns additionally provide paths to use hidden human knowledge, experience, and intuition, all pathways to the creative process.

The extended model still lacks flexibility for problem formulation with its closed structure. Systems implementing fusion levels, relevance, and linguistic interpretation can be collectively described within a human-system cooperation paradigm [10]. Figure 12.1 shows this evolution.

The VDF model can be decomposed into a series of interconnected submodels, forming an approximate computational IF process. The submodels are the human or linguistic, context, concept, and fusion-assistance processes. From a top-level information flow viewpoint, the model can be interpreted as mappings between image information. These mappings can be represented as data relations between input concept, visual- or data-fusion and linguistic processes, and associated problem frames of reference (FRs):

$$[S \xrightarrow{F} G \xrightarrow{DF} W' \xrightarrow{VF} W]_{FR} \tag{12.1}$$

where S represents information source(s); G, W', W are information and fuzzy images; F is an information search filter process; FR is the problem frame of reference; and DF/VF represents data- or visual-fusion operations and processes. See Figure 12.2 for a data description of the computational process.

12.1.1 Human or Linguistic Model

A human "model" consists of one's predefined concepts, context, and intent. Uni- and multimedia information is interpreted as related to predefined concepts, all

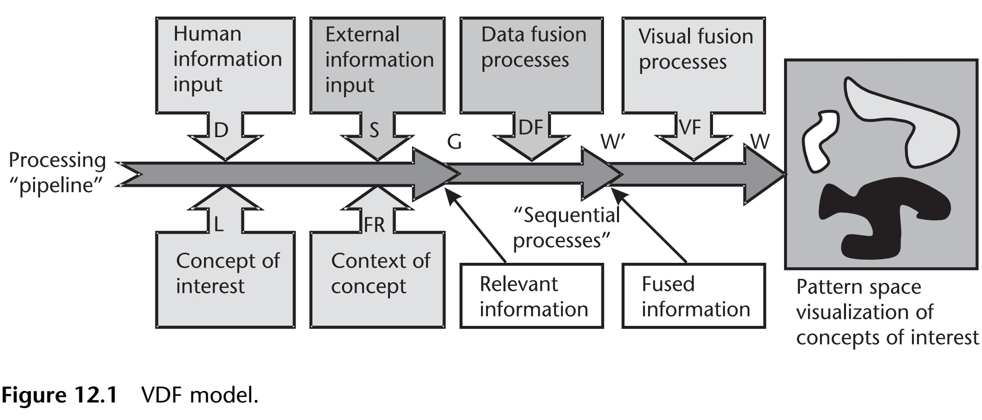


Figure 12.1 VDF model.

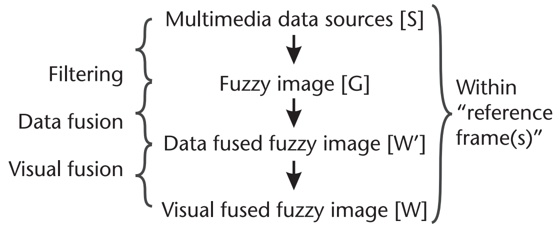


Figure 12.2 Computational VDF model.

within a specific context. Both concept and context are closely related since information is usually perceived within specific environments. Together they form a baseline for the human VDF process model; that is, when perceiving something, we first ascertain its environment before understanding can commence. Physical concepts associated with "normal" environments give a basis for interpretation; for example, the concept "book" and the context "bookshelf" are perceived as something to read (e.g., for information, pleasure). Note a different perception results if context changes from "bookshelf" to "under a table leg," and perceived as a mechanical structure.

12.1.2 Context

Model context frame of reference is represented in geometric terms, as general or specific localizations directed by machine while human perceptual processes are not modeled.

12.1.3 Concepts

Concepts and their specific environments can be represented by directed, hierarchical networks, with root concepts relating subconcepts.

12.1.4 Intent

Human intent is mathematically represented as the union of all requested linguistic constraints [9], weighted by fuzzy qualifiers. The constraints are the specific intentions of a human, interpreted within a predefined conceptual network, modulated by fuzzy qualifiers, such as "all," "some," and "very." Interaction between human intent and computer is without learning, with the most recent constraint taking precedence over prior ones. Individual sentence intent is a union of interpreted words or groups as fuzzy terms. Overall, intent I for simple information requests is the union of terms in one or more sentences.

12.1.5 Fusion-Assistance Processes

Fusion as a cognitive process is represented as operators on information images. They are: (1) data, concerning the preparation of data "chunks" for perception or cognition, and (2) visual, concerning processes that accent and display information for maximum utilization of human perceptual abilities through vision.

12.1.6 Data Fusion

Data fusion (DF) combines input information from G, modulated by the operator "*" (multiplication). A fusion accrual operator (FUS) transforms each pixel (see [11]). Current data input to FUS is at each corresponding pixel (see Figure 12.3).

DF fuses the current information image G with the prior fuzzy image. The resulting DF equation is

$$\mu^{DF}(x_G, y_G) = \left[\mu^{U}(x_G, y_G) \cap \bigcup_i \mu^{UA_i}(x_G, y_G)\right] \tag{12.2}$$

$$* \; FUS\left(\left\{\bigcup_j G(x_G, y_G, j) \xrightarrow{\mu_{TZ}} W'(x_G, y_G)\right\}, \left\{\mu^{W}_{old}(x_W, y_W)\right\}\right)$$

Equation (12.2) states that input G is transformed into fuzzy image W', fused with W_{old}, and the result is modulated by universal and area contextual frames.

12.1.7 Visual Fusion

Visual fusion (VF) occurs primarily within a human, with interaction between human visioperceptual systems and mapped DF patterns. A VF operator implements assistance of perception through relevance thresholding, image processing, and pattern color mapping.

12.1.8 Image Processing

A pattern display is the raw information for human perception. Various image-processing functions are useful in assisting perceptual processes. For an introduction

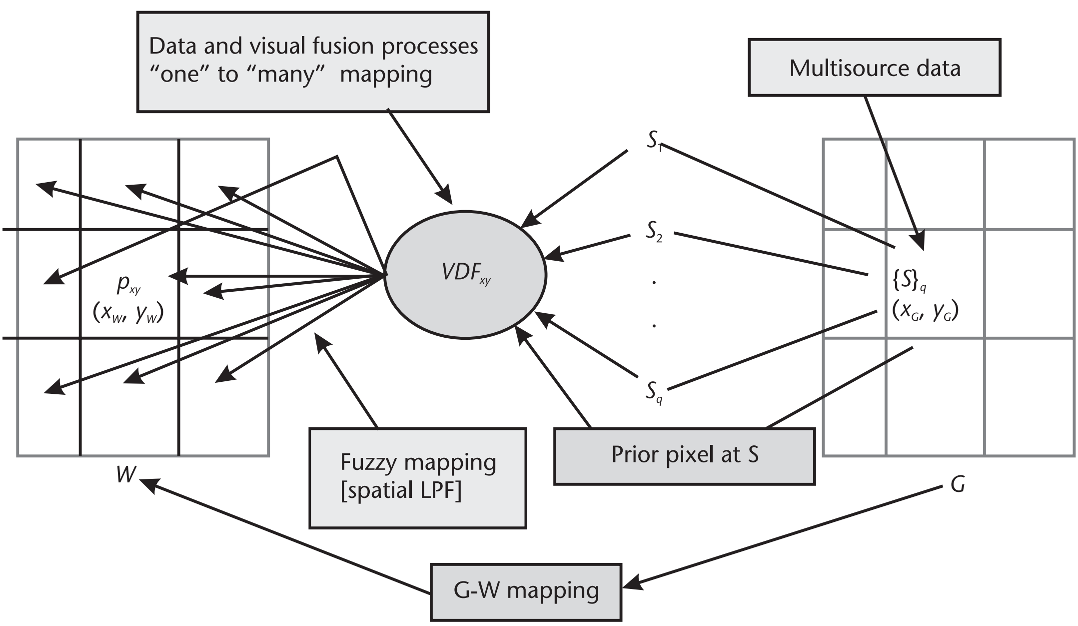


Figure 12.3 Pixel data-fusion operator.

to image processing, see [12, 13]. For example, from image to image, the pattern creation and extinction can be implemented by a pattern-aging image-processing function. Another example is the use of motion and edge detection.

12.2 Blackboard Systems

In the early 1970s, a new problem-solving paradigm, called the blackboard model, emerged with the HEARSAY speech understanding systems to deal with complex problems requiring more flexibility and modularity to represent domain knowledge and reasoning. In particular, it was developed to deal with the difficult characteristics of the speech-understanding problem:

- Large solution space;
- Noisy, unreliable, or uncertain data;
- Variety of input data;
- Need to integrate diverse information;
- Need for many independent or semidependent pieces of knowledge to cooperate in forming a solution;
- Need to use multiple reasoning methods;
- Need for multiple lines of reasoning;
- Need for an evolutionary solution.

The main characteristics of the blackboard model [14–16] are high-level organization of information or knowledge, dynamic control, and an incremental and opportunistic problem-solving process. The domain knowledge is segmented into cooperating modules with their own reasoning modes that interact or communicate via a global data structure. A control mechanism dynamically determines which module of knowledge to apply according to the solution state, enabling opportunistic reasoning.

12.2.1 Terminology and Definitions

The blackboard model corresponds to the most abstract level. The *blackboard framework* is more a detailed view of the blackboard model. *Blackboard shells* serve as a template for the creation of blackboard systems. A *blackboard application* is a blackboard system designed for a particular task. The blackboard framework comprises three components, as illustrated in Figure 12.4 [16]:

- *A set of knowledge sources (KSs):* Independent entities that contain the knowledge needed to solve a problem;
- *The blackboard data structure:* a global data store that corresponds to the working memory of rule-based expert systems;
- *The control mechanism:* A mechanism that reacts to any change on the blackboard (event) to determine what actions to take next (focus of attention).

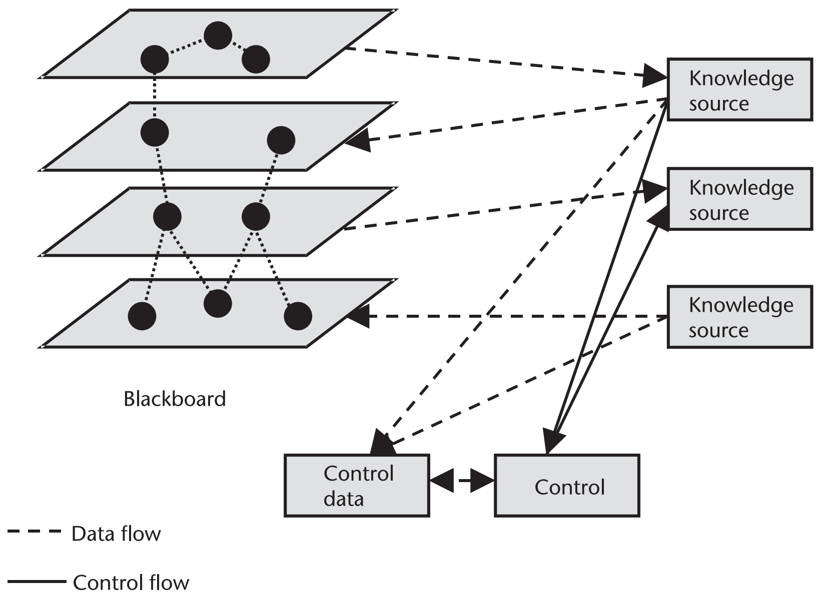


Figure 12.4 The blackboard framework.

Blackboard systems and rule-based (production) systems share some similarities. The blackboard corresponds to the production systems' working memory, the KSs correspond to condition-action rules, and the control strategy corresponds to the conflict-resolution strategy. However, the biggest difference is one of granularity. Production systems' rules are all isomorphic in form and much simpler than blackboard KSs. In fact, a KS could itself be implemented as a production system.

12.2.2 The Blackboard

The blackboard is a global data structure that serves as a medium for communication among the KSs. It is organized into levels of abstraction (conceptual classes) and contains input data, partial solutions (called hypotheses), and, eventually, solutions. Levels form a loose hierarchical structure in which each level can be described as an abstraction of elements of the next lower level. Often, the levels correspond to a particular compositional hierarchy, or part-of hierarchy. Other relationships between levels can be supports or explains links. Objects on the blackboard are also called nodes since they are interlinked into a network structure. Each node can have a number of attributes, each with attached values. Often, hypotheses are assigned credibility ratings as they are created and modified. Furthermore, the solution space can be divided into multiple dimension spaces or blackboard panels.

12.2.3 The Knowledge Sources

Knowledge sources are computational modules embodying the problem-solving knowledge. They are often represented as procedures or sets of rules. The idea is to split the knowledge needed by the system into separate, independent modules, each contributing to the solution of the problem. Thus, KSs usually use information from one level of the blackboard and make some change(s) to another level.

KSs are self-enabling. They are responsible for knowing the conditions under which they can contribute to the solution. So, they are enabled based on the state of the blackboard. KSs cannot communicate with each other directly, but do so rather via changes made to the blackboard.

A KS has two parts: (1) the KS precondition that determines when the KS can be executed by testing the current state of the blackboard, and (2) the KS body that encodes the computation to be performed by the KS when executed. Often, preconditions pass information to the body to be used during execution (e.g., variable bindings). The body of the KS can be either a procedure or a set of production-style rules.

The reasoning style is opportunistic in the sense that the KSs trigger when the opportunity arises. There is no predefined reasoning scheme. Instead, the order of execution and type of reasoning is determined at run-time based on the current state of the system. When the changes are made to the level just above the level being examined, the KS performs bottom-up reasoning, and when the changes are made to the level just below, the KS performs top-down reasoning.

12.2.4 The Control Mechanism

The problem-solving behavior of a blackboard system is determined by the strategy encoded in the control module. Because the execution of enabled KSs must be sequentialized on computers having a single processor, and because blackboard systems typically deal with combinatorially explosive problems, a control module has to focus the problem-solving process. The control mechanism monitors the changes on the blackboard and decides what actions to take next (focus of attention). It uses a control information structure that records changes made on the blackboard (e.g., an event list). Thus, the control is event driven in blackboard architectures.

The most basic control cycle corresponding to a method for either choosing a single KS or ordering all the enabled KSs for execution in a series consists of the following steps:

1. Determine which KSs are enabled.
2. Choose which of the enabled KSs are to be executed, based on some rating function. Typically, only one will execute per cycle.
3. Execute the KS(s). This will cause changes in the state of the blackboard, which will enable other KSs.
4. Go to 1.

The focus of attention can be either the KSs to activate next, the blackboard objects to consider, or a combination of both. The solution is built incrementally, and any type of reasoning step (e.g., data driven, goal driven, model driven) can be applied at each stage of the solution formation.

The control component has to select the action with the maximum expected value. The value of an action is determined by how much it contributes toward progress in problem solving relative to the computational costs of the action, where progress is judged by how much the action reduces the remaining effort required to meet system goals [17].

12.2.5 Hearsay-II, the First and Best-Known Blackboard System

The objective of the Hearsay-II project [18] was to design a speech-based interface to a database of computer-science abstracts to interpret spoken commands and queries. Hearsay-II's input consisted of a spoken database query and the system had to interpret the speech signal and execute the appropriate query.

The blackboard stored interpretations on eight levels, from low-level acoustical signal parameters to high-level sentences, forming a part-of hierarchy.

The solution was built incrementally. Partial interpretations were called hypotheses. KSs used previously introduced hypotheses about various parts of the signal, along with knowledge of speech constraints to introduce new hypotheses or change the strength or weight of existing hypotheses. Nodes formed a graph made of both AND and OR links, with the OR links used to tie together mutually inconsistent alternative hypotheses.

KSs consisted of preconditions and action parts. Preconditions were divided into two parts: trigger and test. The trigger provided simple conditions under which the KS might be useful. The test performed more extensive computations to determine the appropriateness of the action component. It then recorded the set of hypotheses identified as appropriate in a structure called the stimulus frame and the description of the changes that would be made by the action module in a structure called the response frame.

Hearsay-II used an agenda-based control mechanism: All possible actions were placed onto the agenda, the actions were rated on each cycle, and the most highly rated action was chosen for execution (see Figure 12.5). The control was performed

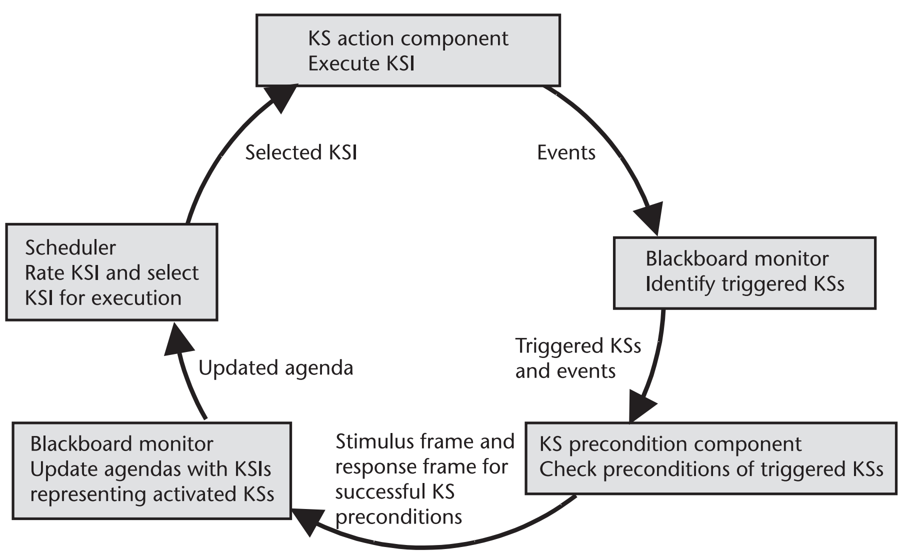


Figure 12.5 Hearsay II control cycle.

by a scheduler and by a blackboard monitor. The monitor observed each change to the blackboard, examined each KS's precondition for activation, and passed the KS's identity to the scheduler. (To optimize this process, changes on the blackboard are described as blackboard event types, and each KS provides a list of event types in which it is interested.) The scheduler then had to select the next activity to perform (either a test precondition or an action) by rating all actions on the agenda based on the information provided by the stimulus and response frames. The control mechanism in Hearsay-II was implemented as a hard-coded procedure.

12.2.6 Types of Control Strategies

Blackboard systems differ greatly in their individual control strategies [17]. The main types of control strategies are:

- *Agenda-based control* (e.g., Hearsay-II, as described earlier).
- *Hierarchical control* using meta–knowledge sources (e.g., CRYSALIS, ATOME). CRYSALIS uses a two-level hierarchy of control KSs (strategy and tasks) to select the domain KSs to be executed. The strategy selects a sequence of tasks to be executed, while task-level KSs select a sequence of domain KSs to be executed. The strategy KS provides the coarse focus of the system based on key blackboard hypotheses, and the task KSs provide the fine focus of the system. In CRYSALIS, all KSs are implemented as sets of rules, without preconditions. The CRYSALIS architecture limits opportunism and context switching. This is partly due to the fact that CRYSALIS pursues a single line of reasoning at a time. In addition, the CRYSALIS application does not require real-time performance that would make the ability to switch focus rapidly more critical.
- *Goal-directed control* (e.g., Distributed Vehicle Monitoring Test Bed). This type of architecture uses goals to integrate data-directed and goal-directed reasoning. It extends the Hearsay-II architecture by adding a goal blackboard and a goal processor. The goal processor instantiates goals on the goal blackboard whose structure mirrors that of the domain blackboard. It is driven by three mapping functions: hypothesis to goal, goal to subgoal, and goal to KS. New goals cause KSs that might achieve the goals to have their preconditions checked. Goals are rated based on the ratings of the hypotheses that stimulated their creation, the ratings of supergoals, and the blackboard level of the goal. The use of explicit goals provides information about the global context of an action. But goals in this context only provide an understanding of the immediate consequences of actions. Goal-directed blackboard goals cannot represent complex, long-term goals.
- *Blackboard-based control* (e.g., BB1). The BB1 blackboard framework introduces the notion of the flexible, run-time control. It extends the Hearsay-II control through the addition of a control-planning mechanism. The control problem is treated as a problem-solving task in itself. The control strategy is stored on a control blackboard that has predefined levels (problem, strategy, focus, policy, to-do set, chosen action), and control KSs are capable of incrementally building and modifying control plans. Control KSs are treated

like domain KSs and selected by the same scheduler. The BB1's scheduler uses a set of active heuristics (rating functions) to rate potential actions. These heuristics can be changed dynamically.

The evolution of blackboard control architectures goes toward sophisticated goal-directed control strategies, based on a detailed representation of goals and the relationship between goals and long-term and global effects of actions, as well as their immediate and local effects.

12.2.7 Summary of Blackboard Architectures

The main advantages of blackboard architectures are:

- They are general and flexible.
- Many various KSs can participate in forming the emerging solution.
- They use multiple reasoning strategies.
- There is no a priori commitment to the order of inferencing steps, such as forward- or backward-chaining.
- They allow flexible switching between bottom-up and top-down reasoning.
- They allow automatic enabling of KSs. Each KS can contribute opportunistically since each has continual access to the state of the solution. Thus, the right knowledge can be applied at the right time.
- The solution is built incrementally, piece by piece, as KSs are activated.
- They are modular.
- Knowledge is partitioned into separate KSs.
- Incorporation of different KSs to do the same task is easy, and the control can select the best-suited KS.
- The blackboard model has proven itself useful in the context of real-time control [19].

However, this type of architecture also has some drawbacks:

- Due to the shared aspect of data structures, representation changes may require modifications to a number of KSs.
- During the 1990s, research was conducted on aspects that needed improvement, such as performance, real-time, and parallelism.

Note that an effective control is critical in blackboard applications that involve significant uncertainty in the data and problem-solving knowledge [17]:

> How the problem is partitioned into subtasks makes a great difference to the clarity of the approach, the speed with which solutions are found, the resources required, and even the ability to solve the problem at all.

12.3 Multiagent Systems

12.3.1 Agent Model

A more recent phenomenon has been the emergence of multiagent systems. Shoham offers the following assessment of use of the term *agent* [20, p. 52][1]:

> Most often, when people in AI use the term "agent," they refer to an entity that functions continuously and autonomously in an environment in which other processes take place and other agents exist.

Shoham's remark identifies three key characteristics of agents:

- *Embeddedness:* Agents exist in an environment in which other processes take place and interact continuously with that environment.
- *Autonomy:* Agents operate autonomously within their environment. They act independently based upon their own volitions.
- *Community:* Agents operate within a community of other agents, which may be human or artificial.

Collectively, these characteristics distinguish agent systems from earlier conceptualizations within computer science. Object-oriented systems aim to model a conception of a world of objects [21, p. 167]:

> One powerful design strategy, which is particularly appropriate to the construction of programs for modelling physical systems, is to base the structure of our programs on the structure of the system being modelled. For each object in the system, we construct a corresponding computational object. For each system action, we define a symbolic operation in our computational model. . . . To a large extent, then, the way we organise a large program is dictated by our perception of the system to be modeled.

Wegner [22] identifies three essential characteristics for object-oriented design:

1. *Objects:* These are the basic computational entities, composed of a local state (often hidden) and a collection of operations permitted for that state. The computational objects are intended to model objects (or individual things) existing out in the world. The computational operations are intended to model the operational properties that can occur with those objects out in the world.
2. *Classes:* These define sets of possible objects. The possible objects composing classes usually support common operations. The classes are intended to model kinds of objects sharing common principles in the world.
3. *Inheritance:* This defines a class hierarchy in which some classes are defined in terms of others. This allows for the inheritance of objects and their operations into another class. This is intended to model an inheritance ordering over the kinds or classes of objects genuinely resident in the world.

1. Shoham goes on to forward a more restrictive interpretation of "agent."

Object-oriented systems differ from agent systems in that objects come with no presupposition of embeddedness, and objects are typically not autonomous [23, p. 26]:

> Objects do it for free; agents do it because they want to.

Blackboard systems often provide the elements of embeddedness and community, but without autonomy [23, p. 309]:

> Blackboard systems were highly influential in the early days of multiagent systems, but are no longer a major research activity . . . [They] were not autonomous agents—they were more closely related to the knowledge sources in the blackboard model—but the metaphors of cooperation and distribution are clearly evident.

In motivating the ATTITUDE multiagent system, Lambert exposes a strong motivation for agent systems generally [24]:

> The current computer science paradigm began in the 1940s with a communicative gulf, with a human user flush with conceptualisations at one extremity and the computer as a complex electronic switching device at the other. The current computer science paradigm has sought to bridge this gulf by dragging the computer closer to the user by embedding human conceptualisation within the machine and then interfacing those conceptualisations to the user as if primitive thereafter. Thus, we have seen the familiar progression of machine languages, assembly languages, floating point arithmetic, higher-level languages, graphical user interfaces, and speech-processing systems. If we continue to pursue this paradigm, then at the automation limit, we would interact with the computer as if it were another user, and we would predict and explain its behaviour in a similar manner to how we predict and explain human behaviour.

From the software-engineering standpoint, multiagent systems aspire to the automation limit of the current computer-science paradigm.

12.3.2 Organizational Models

The emergent behaviors of multiagent systems are not only determined by the capabilities of each of the agents but also, and perhaps primarily, by the organization model used for agent interaction. A number of common organization models are shown in Figure 12.6.

In this figure, the boxes represent agents, and the circles represent particular tasks (or functions, physical devices, or databases) in the system, which are performed (or controlled) by an agent. The arrows show the direction of control between entities in the system, and dashed lines represent cooperation between agents. The system is defined by its three tasks. The way the agents are structured and operate controls how these tasks are performed, hence how the system performs to achieve its (system or global) goals. Note that, except for the centralized organizational model, all of these models are distributed multiagent systems. The system is defined by more than one agent that cooperates or interacts with other agents to achieve the system goals.

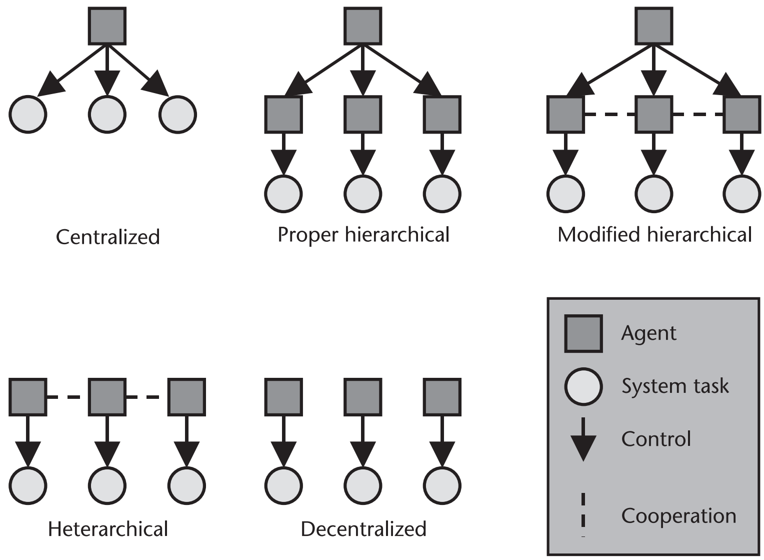


Figure 12.6 Organizational models. (*After:* [25, 26].)

12.3.2.1 Centralized

The centralized organizational model consists of a single agent that controls all tasks within the multiagent system. This model has the usual characteristics of centralized systems, as discussed previously.

12.3.2.2 Decentralized

The decentralized organizational model consists of separate agents that control each task required by the complete system. The system's tasks are divided into small, manageable tasks, each within the capability of a single agent. Therefore, control of the system's tasks is distributed among the collection of agents. A decentralized system allows local optimization, performing its own task in the best way possible, and since the agent is performing a smaller task, it can be more responsive (i.e., there is no bottleneck).

The decentralized model has many of the advantages of distributed systems discussed previously. However, in a truly decentralized system, no agent has a global view of the system beyond itself, nor is there any mechanism for coordinating the actions between the agents. Hence, when the agents' tasks depend on each other, decentralized systems will not operate optimally and may instead produce anarchy.

12.3.2.3 Hierarchical

In a hierarchical organizational model, or the proper hierarchical organizational model in Figure 12.6, the control of the tasks is distributed among separate (slave) agents, and a single (master) agent controls the actions of the slaves. The master provides the centralized component in the system, having a global view of the system. Global decisions can be made using the global information in order to coordinate the activities of the slaves (hence, the tasks), potentially allowing for a high degree of global optimization. The modular hierarchical structure makes it easier to develop systems. Hierarchical models are suited to domains that also have a hierarchical organizational structure (which are a large proportion of domains, e.g., real-world systems); thus, mapping from the domain to the agent system is easier.

Although the hierarchical structure is more distributed, allowing the master agent to delegate most of the decisions to slave agents, the master agent is still a centralized component within the system; thus, the hierarchical structure may suffer from some of the disadvantages of the centralized organizational model. Also, the system depends on communication for operation; hence, communication problems between agents may cause the system to collapse.

The modified hierarchical organizational model has the same structure as the proper hierarchical model, but the slaves can cooperate with each other. This cooperation allows the slaves to share local information about each other, providing them with a partial global view of the system. Using this information, slaves can make some global decisions in order to coordinate their activities with other slaves, providing some scope for global optimization. Since slave agents can make some global decisions, the master agent need only supply loose control (as compared to the master agent in a proper hierarchical system) and provide the rest of the global decisions in the system. This system gives the slaves greater autonomy, allowing them to react to local situations more quickly (more responsively) as they do not have to ask the master for permission to act. Disadvantages include greater communication requirements (hence, greater dependency on reliable communication links) and loss of control by the master (which may not always be a disadvantage but may reduce optimality).

12.3.2.4 Heterarchical

The heterarchical organization model is a decentralized organization model where each agent can cooperate with the others [25–28]. As with the modified hierarchical organizational model, cooperation allows the agents to share local information, giving them a partial view of the global system. Agents can then make global decisions based on the (partial) global information they have in order to coordinate their activities, giving this type of model a higher degree of global optimality and coherence than the decentralized model. Large heterarchical systems can be difficult to develop and manage, with many agents at the same level, lacking organizational structure, trying to cooperate and produce a coordinated solution. The overhead for each agent to communicate with other agents in large systems can be extensive if dependency between tasks is high or a large amount of information is required to coordinate their activities (hence, demanding greater dependency on reliable and fast communication links).

12.3.3 Negotiation

When we have a society of agents that are self-interested (have their own selfish goals), they may need to cooperate with other agents in order to achieve tasks that they cannot perform themselves or that would be more efficiently achieved by cooperating with other agents. Negotiation between agents allows agreements to be reached on the exchange and allocation of resources or tasks so as to increase some utility metric [23, 29].

12.3.3.1 Game-Theoretic Approaches

Agent negotiation methods may be based on game-theory research. They specify a protocol that the agents use to negotiate with other agents. There are two types of negotiation domains: task-oriented domains and worth-oriented domains. In task-oriented domains, a set of agents has a set of tasks that the agents must achieve. They need to negotiate with others to find a way for these tasks to be distributed among the set of agents in order to achieve their own tasks in a more efficient manner than if they performed their own tasks themselves. In worth-oriented domains, each agent wants to achieve some state in the environment, and each state that it wants to achieve (or can be in) has some worth. The agents also have *joint plans*, which define the actions they can perform as a group. The agent must negotiate over joint plans to achieve the state of the environment with the greatest worth.

In negotiation, agents have a *negotiation set*, which consists of all the possible proposals that the agents can make. The *protocol* defines which proposals are legal in the negotiation. Agents use *strategies* to determine which proposals they will make, and agents generally do not have access to (a view of) other agents' strategies. There is also a *termination rule* that determines when an agreement has been found. Negotiation involves a series of rounds where agents make a proposal at each round, based on negotiation set, strategy, and protocol, until eventually (one would hope) an agreement is reached in a particular round, which is determined by the termination rule.

There are two popular strategies in the task-oriented domain:

1. *Monotonic Concession Protocol:* Each agent proposes a deal from its negotiation set. An agreement is reached if an agent proposes a deal such that the utility of its proposal for every other agent is greater than the utility of those agents' own proposals. If an agreement is not reached, then the agents must go another round and propose again, but they cannot propose a deal to any agent that is worse than the previous proposal. If an agreement cannot be found, then the negotiation terminates, and agents perform their own tasks (the conflict deal).
2. *Zeuthen Strategy:* The agent's first proposal is its preferred deal. Whether a particular agent should concede to a worse deal for itself is determined by a *willingness-to-risk-conflict measure*, which is the difference between the utility of the current proposal and that of the conflict deal (perform its own tasks). Therefore, if the willingness-to-risk-conflict measure is low, the agent does not have much to lose if a conflict deal results (over the current proposal); hence, it will be less willing to concede.

12.3.3.2 Argumentation

There are two disadvantages with game-theoretic approaches to negotiation [23]. First, agents cannot justify their position in the negotiation. For example, in buying a car, a dealer may justify the high price by referring to the many features of the car, or the buyer may justify paying less because the features are not required for his or her purposes. Also, if an agent purchases a car for some user, then the user may want to know why the agent paid as much as he or she did. Second, an agent's preferences may change during the negotiation process. If the buyer intends to buy a particular brand of car and, during the negotiation process, discovers that the brand of car tends to break down often, then the buyer may decide to buy a more reliable brand of car.

In order to overcome these problems, argumentation in negotiation can be used. An agent attempts to convince another of the truth or falsity of one or more propositions, with justification, in order to support its case in the negotiation. An agent can use *modus ponens* in its argument ($A \rightarrow B$); that is, if you accept A, and A implies B, then you must accept B. Arguments can be defeated by *rebutting* or *undercutting*. Rebutting is where an agent tries to falsify the conclusion of a given argument; for instance, with the argument $A \rightarrow B$, an agent may rebut this argument with $C \rightarrow \neg B$ (if you accept C, and C implies $\neg B$, then you must accept $\neg B$). Undercutting is where an agent tries to falsify the grounds of a conclusion of a given argument; for instance, with the argument $A \rightarrow B$, an agent may undercut this argument with $C \rightarrow \neg A$ (if you accept C, and C implies $\neg A$, then you must accept $\neg A$).

Different types of arguments have various degrees of acceptability. For example, the acceptability of «$\varnothing \rightarrow p \vee \neg p$» (a tautological argument) is stronger than «$(p, p \rightarrow q) \rightarrow q$» because it is not possible to construct an argument that will defeat «$\varnothing \rightarrow p \vee \neg p$». Five classes of arguments in *increasing* order of acceptability are as follows (Δ is a set of logical formulae that may be inconsistent) [23]:

1. The class of all arguments that may be made from Δ;
2. The class of all nontrivial arguments that may be made from Δ;
3. The class of all arguments that may be made from Δ for which there are no rebutting arguments;
4. The class of all arguments that may be made from Δ for which there are no undercutting arguments;
5. The class of all tautological arguments that may be made from Δ.

The stronger the argument the agent presents in a negotiation, the better. Rebutting an argument is a stronger defeat than undercutting an argument because undercutting allows the agent to find another argument to support the same conclusion [30].

12.3.3.3 Auctions (Competition)

Auctions are a useful and simple (to implement) technique for an auctioneer (agent) to allocate a task or resource to one of a set of bidders (agents). The auctioneer advertises the task or resource that it wants to allocate and receives bids from the bidders. Via some metric to determine the value of the bids (e.g., financial cost), the auctioneer allocates the task or resource to one of the bidders.

The Contract Net Protocol (CNP) is a popular mechanism for agents to allocate tasks and resources to a collection of agents. It uses the auction technique known as the *first-price sealed-bid auction*, which consists of a single round of bidding where bidders submit their bids for the task or resource to the auctioneer. The task or resource is awarded to the agent that made the best bid (based on some criteria). CNP will be discussed further in following sections.

There are other auction techniques, including

- *English auctions:* The auctioneer starts the auction by advertising the task or resource and suggesting a reserve price. Bidders must bid more than the current price, where all bids can be seen by all agents participating in the auction. The task or resource is allocated to the bidder that provided the bid with the highest price, which no other bidder is willing to better.
- *Dutch auctions:* The auctioneer begins the auction by advertising the task or resource at some high price (above the expected price), and lowers the price continually until a bidder makes a bid for the task or resource equal to the current price, at which point the task or resource is allocated to this bidder.
- *Vickrey auctions:* Vickrey auctions have only a single round. The auctioneer advertises the task or resource to the bidders, and the bidders submit their bids, which no other agent can view. The task or resource is awarded to the bidder that provided the highest price, but the bidder only pays the price of the second highest bid. The motivation behind Vickrey auctions is that it makes truth-telling a dominant strategy (i.e., bidders are better off bidding their true valuation of the task or resource). If the bidder bids more than the task or resources true valuation, he or she runs the risk of paying more than its true valuation. If the bidder bids less, he or she will have a smaller chance of winning the task or resource, without changing the price he or she will need to pay (since the bidder pays the price of the second to highest bid).

12.3.4 Cooperative Distributed Problem Solving

Cooperative distributed problem solving (CDPS) is a very large topic that has been studied for well over a decade. Summarizing the work of Wooldridge [23], a definition of CDPS is as follows [31]:

> CDPS studies how a loosely-coupled network of problem solvers[2] can work together to solve problems that are beyond their individual capabilities. Each problem-solving node in the network is capable of sophisticated problem-solving and can work independently, but the problems faced by the nodes cannot be completed without cooperation. Cooperation is necessary because no single node has sufficient

2. This can include benevolent or self-interested agents.

expertise, resources, and information to solve a problem, and different nodes might have expertise for solving different parts of the problem.

There are two issues to be considered in CDPS [32]:

1. *Coherence:* A measure(s) of how well the multiagent system performs as a single entity, where the measure may be quality of solution, efficiency of resource usage, and so forth;
2. *Coordination:* The extent to which agents can prevent extraneous activity (i.e., avoid conflicting or interfering with the activities of other agents).

A CDPS process has at most three stages [33]:

1. *Problem decomposition and allocation:* The complete problem is decomposed into smaller subproblems, and those are decomposed further into smaller subproblems, until the subproblems can be solved by individual agents. These subproblems are allocated to suitable agents to solve.
2. *Subproblem solution:* The allocated subproblem is solved locally by the agents. Agents may share information to assist each other in solving their subproblems.
3. *Solution synthesis:* The subproblem solutions are assembled to provide a complete solution to the complete problem.

12.3.4.1 Task Sharing and Result Sharing

Task sharing is the process of decomposing a problem and allocating the decomposed parts of the problem to various (suitable) agents (i.e., CDPS process 1, above). In a system comprising heterogeneous and self-interested agents, techniques for task sharing include those described previously.

CNP [34] facilitates task sharing. A *manager agent* requiring that some task be performed advertises the task to other *bidding agents* within the network. The manager agent may broadcast the task to all agents or may limit the broadcast if it has knowledge of which agents may be suitable candidates. The bidding agents that receive the task announcement may submit a bid to perform the task, if they believe they are eligible to do so, before some specified deadline. The manager agent analyses the bids and selects the most appropriate bidding agent to perform the task, granting that agent the task. The bidding agent awarded the task may report back to the manager agent on the completion of the task.

Result Sharing

Result sharing is the process whereby agents share information or results to assist others in solving their problems (i.e., CDPS process 2, above). This can increase group performance by providing other agents a global view of the situation, allowing agents to cross-check results, and reducing the time to produce a result by not duplicating work [35].

Combining Task and Result Sharing

FELINE [36] is a cooperating expert system comprising cooperating experts (agents) that have expertise in distinct areas. Agents cooperate both to *share knowledge* and to *distribute subtasks*. Each agent has an *environment model*, which contains beliefs about itself and its environment (other agents). Each entry in the environment model contains two attributes: *skill* and *interests*. Skill identifies the particular agent's domain expertise, while interests are a set of hypotheses for which the agents require a truth value. FELINE utilizes three speech acts: *request, respond*, and *inform*. An agent sends a request (to an agent with the expertise to derive a solution) when it requires a truth value for a hypothesis. An agent responds when it derives the truth value for hypothesis that was requested. An agent will inform another of the truth value of a hypothesis if that agent is *interested* in the hypothesis.

FELINE allows both goal-driven (backward-chaining) and data-driven (forward-chaining) inferencing. In goal-driven inferencing, when the agent wants to establish the truth value of some hypothesis that is not known, it will search its environment model to discover which agent has the skill to do so. The agent will be sent a request and return a response with the truth value of the hypothesis. With data-driven inferencing, if an agent receives or generates[3] a new hypothesis, it searches its environment model for agents that have an interest in the hypothesis. An inform message is sent to the relevant agents, containing the hypothesis and the truth value.

12.3.4.2 Handling Inconsistency

Agents may have inconsistencies in their beliefs or goals. Durfee, Lesser, and Corkill propose three approaches to deal with inconsistencies [37]:

1. Do not allow it to occur or ignore it (e.g., in the CNP, the manager agent has the only view of the problem).
2. Resolve inconsistencies through negotiation, which is undesirable due to the computation costs.
3. Build systems that degrade gracefully in the presence of inconsistencies. This is the most desirable approach. Systems that behave robustly in the presence of inconsistency are referred to as functionally accurate or cooperative (FA/C) [38]. In FA/C systems, agents exchange high-level (partial) results rather than raw data. Inconsistencies in the raw data may not prevent the agent from progressing with the problem solving and may not have any (or, at least, not a large) effect on the consistency of the high-level results at all. Also, inconsistency (and uncertainty) is implicitly resolved during the problem-solving process when partial results are exchanged and compared with other partial solutions. Finally, there may be many ways to arrive at a solution; hence, if one method fails (due to localized failures or bottlenecks in problem solving), an alternative approach can be used.

3. It does this by inferencing over its own facts and rules.

12.3.4.3 Coordination

Wooldridge [23] addresses the coordination problem this way:

> The coordination problem is that of managing inter-dependencies between the activities of agents: some coordination mechanism is essential if the activities that agents can engage in can interact in any way.

There are two types of coordination relationships, *positive* and *negative* [39]. Positive relationships occur when agents' actions (plans) complement (or overlap) each other; hence, they can be combined to reduce the workload of the agents in the system. This can be explicit (e.g., an agent requests help from another agent) or implicit (e.g., an action, or a side effect of the action, that an agent performs happens to be the same action that another agent needs to perform). Negative relationships occur when agents' actions conflict (interfere) with each other, resulting in agents' not being able to achieve their goals or making it harder (entailing more work than originally intended) for them to achieve their goals. Agents should be able to recognize and manage these relationships at run time in order to coordinate their activities dynamically. Approaches to achieving this include partial global planning, joint intentions, mutual modeling, and norms and social laws.

Partial Global Planning

Partial global planning (PGP) [40–42] was used in the Distributed Vehicle Monitoring Test Bed (DVMT), which was used to track vehicles. DVMT uses distributed sensors to monitor vehicle paths. An agent will have access to one sensor, or a small set of them, thus having a restricted (local) view of the system. Each agent forms a local plan based on its own local goals and sensor data. Agents then share their local plans to obtain a more global view of the system and alter their local plans to coordinate their activities. A metalevel structure is used to guide the cooperation process, deciding which information agents should share, with whom, and when.

Decker's generalized partial global planning (GPGP) [43] is an extension of PGP, used in his task analysis and environment modeling system (TAEMS) test bed. It uses five techniques for coordinating agent activities:

1. *Updating nonlocal viewpoints:* Agents can share no information, all information, or some intermediate-level information.
2. *Communicating results:* Agents may communicate all of their results, communicate only those results that are essential to satisfy obligations, or send results to those who have a registered interest in them.
3. *Handling simple redundancy:* If two or more agents are working on the same problem, then one is selected at random to perform the task, and the result is sent to the other interested agents.
4. *Handling hard coordination relationships:* If agents encounter negative relationships (conflicting or interfering actions), then activities are rescheduled to resolve the problem.

5. *Handling soft coordination relationships:* If agents encounter positive relationships (overlapping or complementing actions), then rescheduling (negotiation) is performed to take advantage of it, and if no solution is possible, the system takes no action.

Joint Intentions

Based on human teamwork models, a collection (team) of agents have a *joint intention* [44, 45] when each agent has the same intention toward a common goal, and they all cooperate with each other to achieve it. Joint intention implies cooperation because agents with the same intention may not necessarily cooperate (or act as a team). For example [46],

> A group of people are sitting in a park. As a result of a sudden downpour all of them run to a tree in the middle of the park because it is the only available source of shelter. This may be coordinated behaviour, but it is not cooperative action, as each person has the intention of stopping themselves from becoming wet, and even if they are aware of what others are doing and what their goals are, it does not affect their intended action. This contrasts with the situation in which the people are dancers, and the choreography calls for them to converge on a common point (the tree). In this case, the individuals are performing exactly the same actions as before, but because they each have the aim of meeting at the central point as a consequence of the overall aim of executing the dance, this is cooperative action.

Being part of a team implies *responsibility* toward other agents in the team [44]. Therefore, for example, if one agent in the team determines that a goal (joint intention) is not achievable, then it is responsible for informing other agents in the team that are also working toward achieving the same goal.

When agents form a joint intention, they have a *joint commitment* to achieving the common goal. The joint commitment persists among the team until it becomes redundant. Reasons for dropping a joint commitment include: (1) the goal has been achieved, (2) the goal can no longer be achieved, (3) or the motivation to achieve the goal is no longer present (irrelevant) [47]. Convention [48] specifies the conditions under which a commitment should be abandoned and how agents should behave locally and toward each other when this occurs. It may specify that agents that drop joint intentions (because, say, the goal can no longer be achieved) must inform other agents in the team of this fact. If all agents adopt the convention, then every agent knows what is expected of itself and of others in the team.

A joint intention can be considered as the common intention of a collection of agents having a joint persistent goal (JPG) [44] (the achievement of a goal G), while believing that all others in the team also have the JPG to achieve G. A JPG is when all agents believe the goal G has not been achieved but is possible, and they have a mutual persistent goal to achieve G. Until the goal G is achieved, unachievable, or irrelevant, the goal to achieve G will persist. If an agent discovers that the JPG has been achieved, is unachievable, or has become irrelevant, it will have a goal of making this fact a mutual belief by informing the other agents in the team.

Practical applications of the theory of joint intentions include ARCHON [48–52] and Tambe's Steam framework [53]. ARCHON is an industrial control system where commitments and conventions are encoded as rules in a rule-based system, adding coordination structures in the agents' reasoning mechanism. The Steam framework facilitates teamwork among agents and is currently used in military simulations and robotic soccer. Programmed in the Soar rule-based architecture, it encodes about 250 domain-independent commitment and convention rules, which also allow for hierarchical team structures among agents.

Finally, Wooldridge and Jennings presented a four-stage teamwork-based model of CDPS [54, 55]:

1. *Recognition:* An agent should recognize the potential for cooperative action with respect to its goal(s). This occurs if an agent cannot achieve the goal itself or if it would be beneficial for the agent not to work alone, and the agent also believes that a group of agents can achieve the goal.
2. *Team formation:* The agent forms a team of suitable agents to achieve the goal, say, by requesting assistance. This results in a group of agents with a nominal commitment to achieving the mutual goal and also to believing that they can achieve the goal, without knowing how they can or will achieve it.
3. *Plan formation:* Agents must collectively come to some agreement on the course of action (distributed plan) required to achieve the mutual goal, say, by negotiation.
4. *Team action:* The distributed plan is executed with agents' using a convention, such as JPG, to define the social behavior of the team members throughout the execution.

Mutual Modeling

Coordination by mutual modeling, or "cooperation without communication" [56] requires that agents have a model of the beliefs, intentions, and so forth, of other agents in the society and that they coordinate their activities based on this model (i.e., if an agent knows what other agents are going to do, then it can plan its own actions to coordinate with their actions). Explicit communication is not required for this type of coordination.

Gasser's multiagent computing environment (MACE) system [57, 58] uses the mutual-modeling technique. Agents in MACE contain "acquaintance models," which maintain six types of information about other agents: the name of the agent, the class that the agent belongs to (to provide organizational structure), the role of the agent in the class, the skills or capabilities of the agent, the goals that the agent wants to achieve, and plans as to how the agent may achieve its goals. Agents in MACE use these acquaintance models to coordinate their activity with other agents.

Norms and Social Laws

Agents can coordinate their activities using norms and social laws [23].

> A norm is simply an established, expected pattern of behaviour; the term social law carries essentially the same meaning, but it is usually implied that social laws carry with them some authority.

An example of a norm is thanking someone who performs some duty for you; an example of social law is driving at or under the speed limit (it is a social law because it is enforced). Agents can use norms and social laws to regulate their behavior in social settings and thus act in a coordinated manner. Conventions are essentially norms and social laws, defining how an agent should behave in various situations [59].

Conventions within an agent society are defined either offline or as an emergent property of the system. Defining conventions offline is simpler to implement and provides greater control over the system's functionality. An example of offline design includes defining negotiation protocols or the CNP. Disadvantages include the following: (1) not all system characteristics may be known at design time, (2) agents' goals, hence conventions, may continually change, and (3) it may be difficult to define or predict suitable conventions for complex systems. In such cases, allowing conventions to be defined as an emergent property of the system may be more suitable.

In order for some norm or social law to emerge from a set of agents, the agents need to be able to "reach a global agreement on the use of social conventions by using only locally available information" [23]. Shoham and Tennenholtz [60] studied the problem of obtaining global agreement on one of two strategies used by agents based on local (partial) information (i.e., an agent has no global view to see the strategies used by all other agents; it can only see the strategy of one other agent). An agent must make a decision on which of the two strategies to use based on the current and previous local information about strategies it has seen other agents use. A number of different approaches for agents to update their strategy have been proposed (see references for an evaluation/discussion) [61–63]:

- *Simple majority:* Agents change to the strategy that they have observed most often.
- *Simple majority with agent types:* This is the same as the simple-majority approach, except agents can see the local information of other agents of the same type and base their decisions on the collective information.
- *Simple majority with communication on success:* When an agent reaches a certain level of success with a particular strategy, local information about its experiences with (only) this strategy is sent to agents of the same type.
- *Highest cumulative reward:* Each strategy has a particular associated payoff, and the agent uses the strategy that results in the highest cumulative payoff at the time.

12.3.4.4 Multiagent Planning and Synchronization

There are three categories of multiagent planning [35]:

1. *Centralized planning for distributed plans:* A centralized agent develops a plan for a group of agents, then distributes the plan to the agents for execution.
2. *Distributed planning:* A group of agents cooperatively form a centralized plan, where each agent is an expert in different aspects of the overall plan and contributes to its formation.

3. *Distributed planning for distributed plans:* A group of agents form individual plans and cooperate to coordinate their local plans with other agents' local plans.

The third category, which is the most difficult to implement and control, can be achieved using either negotiation, if self-interested agents are present, or *plan merging*, the process of collecting individual agents' plans and generating a conflict-free (synchronized) multiagent plan. Both Georgeff [64] and Stuart [65] have implemented such systems.

12.3.5 Agent Technologies

Some technologies applied to agent-based computing, multiagent systems, and agent infrastructure are discussed next.

12.3.5.1 Agent Communication Languages

Interagent communication is the lifeblood of multiagent systems. Agents need to communicate with users, clients, services, and each other in order for the power of multiagent systems to be realized. Common agent communication languages (ACLs) are required to allow diverse agent systems to communicate and exchange complex beliefs, plans, goals, and intentions.

The ACLs [66, 67] most widely used for agent-to-agent communication are

- *Knowledge Query Manipulation Language (KQML)* [68]: This was developed in the early 1990s as part of the U.S. ARPA Knowledge-Sharing Effort and is a language and protocol for exchanging information and knowledge.
- *Foundation for Intelligent Physical Agents (FIPA)–ACL* [69]: FIPA is a nonprofit organization aimed at producing standards for the interoperation of heterogeneous software agents. The FIPA-ACL incorporates many of the aspects of KQML.

Both of these ACLs are based on speech-act theory: messages (or performatives) are considered communicative acts by which agents can exchange beliefs with each other and invoke goals in other agents [66]. These systems specify a set of message types and associated handshaking protocols for making exchanges. Message content can be expressed in any language the agent developer deems suitable. They incorporate, advertise, publish, and subscribe mechanisms to allow agents to advertise their capabilities and publish information for access by registered subscribers. Messages in these languages are usually expressed as strings rather than language-specific data structures, which makes it possible for very different agent implementations to communicate using them.

FIPA also produces higher-level specifications that describe how messages should be exchanged for a given interaction type. The interaction types defined by FIPA in their Interaction Protocol Library include request, query, contract net, and auction interactions.

As well as agent-to-agent communication, agents need to communicate with user interfaces, other services, and clients. To achieve this, XML formats such as SOAP and Web service definition language (WSDL) allow expressive communication to occur, but at some cost in network bandwidth and user accessibility. User interfaces must interpret these messages before they become accessible to the operator. KQML and FIPA-ACL also allow agents to communicate messages in these formats by using XML as the message content language.

The CoAX experiments [70–72] demonstrated how agents and humans could cooperate in a coalition environment, using a combination of KQML, FIPA-ACL, and XML to provide a dynamic, adaptable, and flexible C2 system.

12.3.5.2 Matchmaker Agents

Matchmaker agents in multiagent systems are used to maintain updated repositories of information on agents currently engaged in the system, their capabilities, and the services that they can provide (via a publish/subscribe mechanism for example). This removes the burden on individual agents to identify all the agents in the system with which they may need to interact. Instead, agents need only contact the matchmaker agent with a description of the task required, and the matchmaker (perhaps adaptively) determines the agent most suitable for this task. Broker agents can extend this concept by accepting the task, then assigning it to another agent registered with the broker. The broker can prioritize and optimize task assignments, perhaps decomposing the task into partial tasks and assigning the partial tasks to other capable agents. This provides an effective means for mediating the interactions between agents in an open system. The RETSINA [73] and Sensible Agents [74] architectures are examples of agent platforms that offer such matchmaking and brokering services.

12.3.5.3 Mobile Agents

Mobile agents are agents that can move across a network to execute on another host platform or virtual machine. Mobile agents allow a truly distributed system, as agents with specialist capabilities can be deployed dynamically to network nodes that provide system resources, such as spare CPU time, fast network access, and access to information sources or sensor feeds. This allows the deployed agents to perform information processing that takes advantage of the available resources and to send the information products back across the network to the clients. In this way, mobile agents allow distributed systems to optimize system resource usage dynamically. To be effective in this way, however, mobile agents need to be small enough to be transmitted efficiently across the network to a remote node.

A key issue that must be addressed with mobile agents is the administratability of these systems to protect host nodes from malicious or defective agents. This involves consideration of issues such as authorization policies, resource usage, monitoring, and control. The KAoS [75–79] framework has been used with the NOMADS [80] mobile-agent system to apply policies successfully to control mobile agents in the CoAX experiments and protect a coalition C2 system from malicious attack [71, 81]. The CoAX experiments also demonstrated that policies can be used to allow mobile agents to move seamlessly from the NOMADS infrastructure to Dartmouth's D'Agents [82] mobile-agent infrastructure [72].

12.3.5.4 Agent Languages

Numerous agent languages and architectures are available. This section does not attempt to survey them in detail but instead discusses some of the systems that have been used in the areas of military and C2 systems, such as the DARPA Ultra*Log and CoAX programs. A key point to make here is that just as any coalition IF system will need to integrate distributed heterogeneous components, any multiagent system that operates in this environment will also need to integrate heterogeneous agent systems. The ability of these architectures to exploit common ACLs and XML (to interact with Web services and users) will be critical.

ATTITUDE

ATTITUDE [24] is an agent programming language developed at DSTO, based on an extended belief, desire, and intention (BDI) architecture that incorporates research into multiagent reasoning, contextual reasoning, and reasoning under uncertainty. In the BDI architecture, agents have beliefs about the world, and they must satisfy some primary goal or intention by forming desires that are subtasks or desirable states that the agent wishes to occur, moving the agent closer to achieving its intention. Intentions and desires are attained using plans, or what ATTITUDE calls routines, which are sets of instructions that tell the agent how to accomplish certain goals (or desires or intentions). The term *routine* is used because it applies to both computer science (i.e., computer-program routine) and behavior (e.g., a person has a routine to drive a car: open the door, get in, put the key in the ignition, and so forth). ATTITUDE has been designed specifically to support the programming of reactive systems and for information fusion.

ATTITUDE is so named because it utilizes *propositional-attitude* expressions as programming instructions to achieve its desires and intentions. Propositional-attitude instructions have the form

<subject> <attitude> <proposition>

where

<subject> denotes an individual (agent) or group of individuals whose mental state is being characterized (e.g., Fred, Harry);

<attitude> is the subject's dispositional attitude toward that claim about the world (e.g., believe, ask if believe, desire, also desire, expect, anticipate);

<proposition> is a propositional expression that describes some propositional claim about the world (e.g., it is raining, the sky is blue, today is Monday).

Examples of propositional-attitude instructions include

Fred believe (sky is blue);

Wilma expect (it will rain);

Barney desire (new hat).

Using these mental attitudes, we can determine how an individual will act and behave. By applying these mental attitudes within an ATTITUDE agent, its actions and behavior can be controlled. For example, when a software agent encounters the instruction "Fred believe (sky is blue)," it will issue a message to software agent Fred instructing him to believe that "(sky is blue)." Similarly, when encountering the instruction "I believe (sky is blue)," it will itself attempt to believe that "(sky is blue)."

An important characteristic of attitude programming is that each propositional-attitude instruction either succeeds or fails, possibly with side effects, depending on whether the recipient agent is able to satisfy the instructional request. Computational routines for a software agent arise by linking together instructions. The execution path selected through a network of instructions is determined by the successes and failures of the instructions attempted along the way. The flow of control is therefore governed by a semantics of success. ATTITUDE contains several control structures to manage this. For example:

- *Kleene star* ("*"): This will continuously execute a section of the routine until execution fails.
- *Concatenate* ("^"): This will concatenate a sequence of two or more propositional-attitude instructions to form a routine or part thereof. It succeeds when all instructions inside the concatenation operator succeed.
- *Exclusive union* ("|"): This is used when there are alternatives to achieve a goal. It uses two or more routines, or parts thereof, which the agent will attempt to execute in order, and succeeds when one of the routines executes successfully.
- *Guarded exclusive union* ("#"): This can be used to simulate an if-then-else statement. A "guarded" statement (instruction or routine) is executed, and if it succeeds, its corresponding routine is executed. If the guarded statement fails, then the next guarded statement in the guarded exclusive union operator, together with its corresponding routine, is attempted, and so on. If the guarded statement succeeds, but its corresponding routine fails, or if none of the guarded statements succeeds, then an error statement is executed.
- *Intrinsic urgency* ("!"): This is used to assign a particular execution priority to the specified section of a routine. This determines the order of execution of propositional-attitude instructions on the ATTITUDE task agenda.

ATTITUDE beliefs can include Horn clause rules, and an inference engine allows declarative reasoning with the beliefs in an agent's knowledge base. This has also been extended to include a Bayesian inference engine [83] by assigning a set of conditional probabilities to the beliefs stored in the knowledge base.

An agent's knowledge base in ATTITUDE can be partitioned into *events*, which represent a collection of beliefs about the world over a bounded region of time and space. For example:

John believe (sky is blue) in event ?Monday

John believe (sky is gray) in event ?Tuesday

John has two beliefs about the color of the sky, one that it is blue and the other that it is gray. It can be seen that the two beliefs correspond to two different events, one event being ?Monday and the other being ?Tuesday. Events give the user the ability to query the knowledge base in relation to certain events. Events can be "clipped together" using Boolean operators to represent a *scenario* in the world, and the inference engine can then be applied to this scenario. This allows ATTITUDE to apply contextual and what-if reasoning with its beliefs, as shown in Figure 12.7, where the assessment of the combined situation is radically different from the assessment of the initial situation alone.

ATTITUDE can dynamically form a group of agents and use the group as the *subject* for its proposition attitude instructions. This allows team behavior to be captured and also enables ATTITUDE to perform inferences across the beliefs of all members of a group so that, as an ensemble, they can reach conclusions that no agent could reach individually.

ATTITUDE is implemented in an interpreted environment, which can be run on Win32, Linux, Sun Solaris, and SGI IRIX platforms. An ATTITUDE plan file contains the definition, routines, variables, and so forth, of multiple individual agents. When ATTITUDE is executed, it reads and compiles these agents into a set of internal data structures and executes an initialization routine that starts up the appropriate agents. During execution, ATTITUDE can receive external messages (as propositional-attitude instructions or combinations thereof) from various input streams, which it will interpret and execute like any other instruction. Using this mechanism, ATTITUDE agents can learn new routines and functions that are sent

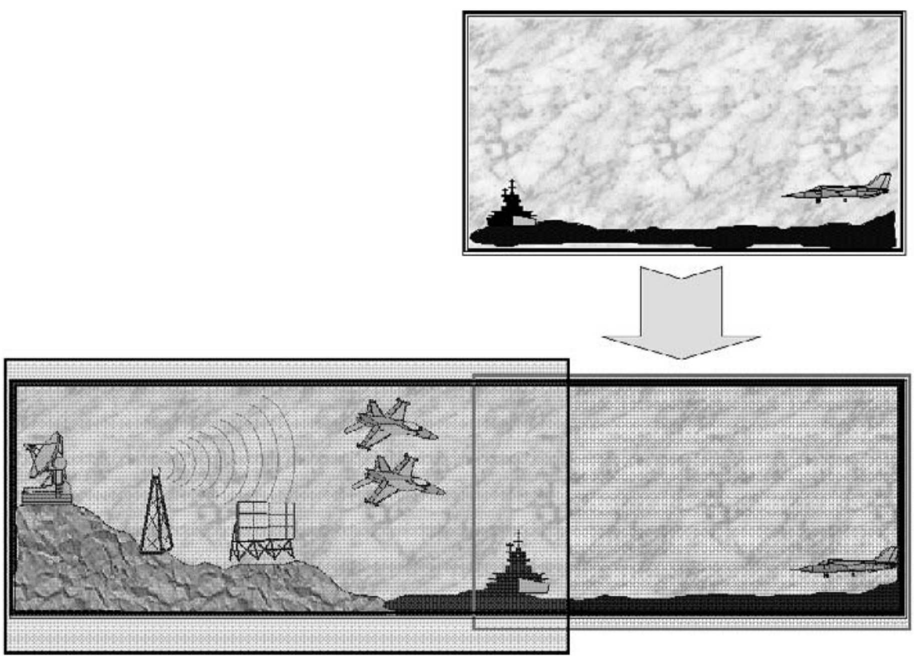


Figure 12.7 ATTITUDE events can be clipped together to form scenarios.

to them as messages. ATTITUDE can also send messages to various output streams, which may be connected to physical effectors or sent across the network. A Java wrapper has been developed for ATTITUDE to allow it connect to third-party software, such as the CoABS middleware [84]. All communication across the CoABS Grid is then done via the external messaging mechanism described above, as shown in Figure 12.8.

JACK

JACK Intelligent Agents [85] is a framework for building and running multiagent applications. JACK incorporates the BDI model and allows developers to create new reasoning models to suit their particular requirements. JACK includes facilities for the creation of intelligent team behavior through the advanced JACK Teams model.

JACK is implemented in Java, and the JACK Agent Language extends Java with constructs for agent characteristics, such as plans and events. JACK agents can be run on one CPU or distributed among multiple CPUs on a local-area network, wide-area network, or the Internet. It builds on the security model provided by the JAVA platform to secure communications and block unauthorized access to data. JACK agents are lightweight and take advantage of the efficient Java multi-threading environment.

JACK is a third generation agent system, emerging from the two previous generations of agent systems, PRS and dMARS. JACK has a small computational footprint (JACK can run on a personal digital assistant, or PDA) and an efficient, component-based design (to ease integration with other software). JACK provides a set of graphical user interfaces suitable for developers and less technical users, such as analysts and operational staff. It also includes graphical design and tracing facilities.

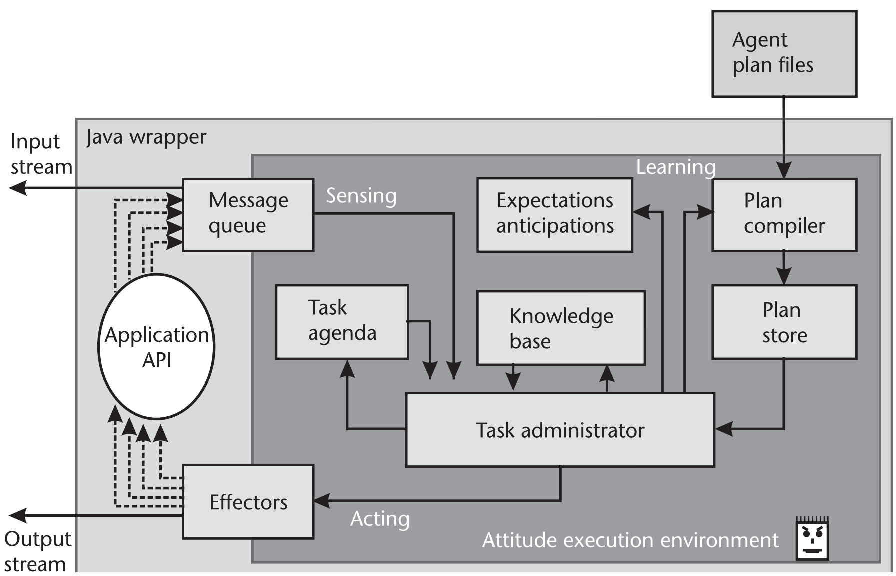


Figure 12.8 Attitude multiagent architecture, showing encapsulation in Java wrapper. Individual agents access partitions of knowledge base and plan store.

JACK is a commercial product with commercial-product design and support standards, with new releases about every six months.

Cougaar

As discussed in Chapter 10, Cougaar [86] is a software architecture, developed for the DARPA Advanced Logistics Project (ALP) and the DARPA Ultra*Log programs, that enables distributed agent-based applications [87].

Cougaar agents are based around a blackboard and plug-ins, as shown in Figure 12.9. Cougaar agents maintain knowledge of their internal state and the external world on the blackboard, which is handled by specialized plug-ins known as logic providers.

The Cougaar agent model is designed to emulate the human cognitive process, which iteratively and/or recursively invokes one of several strategies:

- *Decomposing*: Breaking a problem into smaller subproblems;
- *Delegating:* Giving a problem to another resource to solve;
- *Consolidating:* Taking a number of independent pieces and handling them as a single problem;
- *Monitoring:* Continually checking to make sure things are proceeding as planned and correcting or reacting accordingly;
- *Gathering:* Getting information from the outside world;
- *Reporting:* Reporting back to the outside world;
- *Acting:* Performing some action that impacts with real entities in real time.

Cougaar agents contain specialist plug-in templates that match to the elements of this model:

- *Task expander (decomposing):* Takes a task and decomposes it into subtasks;

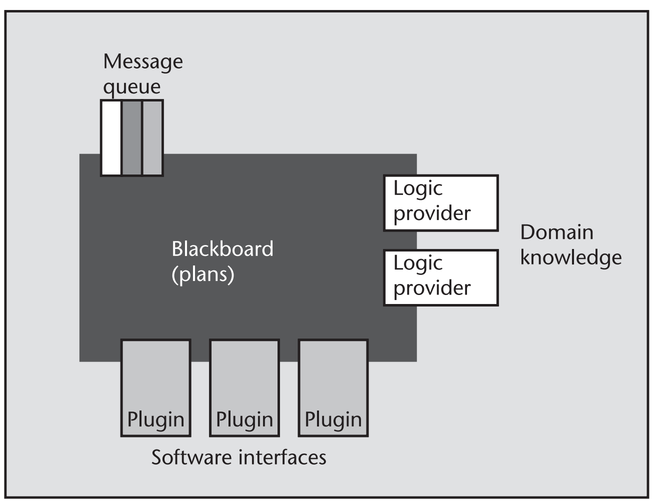


Figure 12.9 Cougaar agent model.

- *Task allocator (delegating):* Allocates tasks to appropriate resources for final handling or further disposition;
- *Task aggregator (consolidating):* Joins a sets of tasks into a supertasks;
- *Task assessor (monitoring):* Assesses the plan for consistency and forces replanning when necessary;
- *Logical data model plug-in (gathering):* Reads new or changed information from external data sources;
- *User interface plug-in (reporting):* Provides an external user interface;
- *Execution (acting):* Interacts with external entities, objects, and systems;

Cougaar plug-ins are designed to run relatively independently from the agent. A plug-in scheduler is used to monitor the activity of the plug-ins and blackboard, and only runs plug-ins when there is activity that is "interesting" to the plug-in or a scheduled execution window occurs. As discussed previously, Cougaar also supports connections and interfaces to contemporary, legacy, and partner systems through plug-ins. Plug-in interfaces include SQL, JDBC, XML, Java JINI, screen scraping, and dynamic link library (DLL) invocations [88]. A plug-in API allows access and interface to other services.

RETSINA

Reusable Environment for Task Structured Intelligent Network Agents (RETSINA) [73] is an open multiagent system developed at the Intelligent Software Agents Laboratory at Carnegie Mellon University that supports communities of heterogeneous agents. The RETSINA system (Figure 12.10) has been implemented on the premise that agents in a system should form a community of peers that engage in peer-to-peer interactions. Any coordination structure in the community of agents should emerge from the relations between agents, rather than as a result of the imposed constraints of the infrastructure itself. In accordance with this premise, RETSINA does not employ centralized control within the multiagent system; rather, it implements distributed services that facilitate the interactions between agents as opposed to managing them.

The RETSINA functional architecture consists of four basic agent types:

1. *Interface agents:* Interact with users, receive user input, and display results;
2. *Task agents:* Help users perform tasks, formulate problem-solving plans, and carry out these plans by coordinating and exchanging information with other software agents;
3. *Information agents:* Provide intelligent access to a heterogeneous collection of information sources;
4. *Middle agents:* Help match agents that request services with agents that provide services.

Each RETSINA agent has separate reusable modules for communicating, planning, scheduling, and monitoring the execution of tasks and requests from other agents, as shown in Figure 12.11.

- The communication and coordination module accepts and interprets messages and requests from other agents.

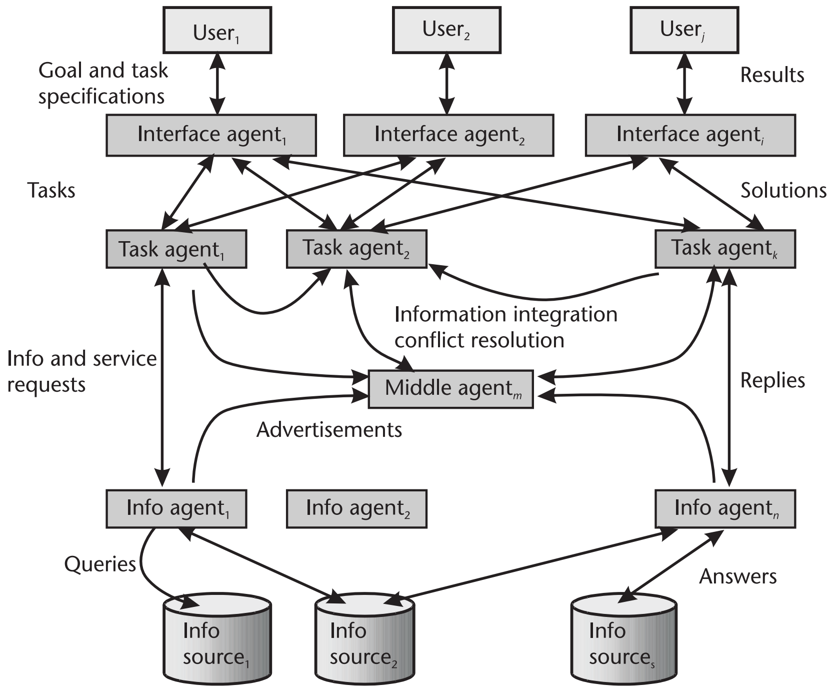


Figure 12.10 RESTINA multiagent architecture. (*After:* [73].)

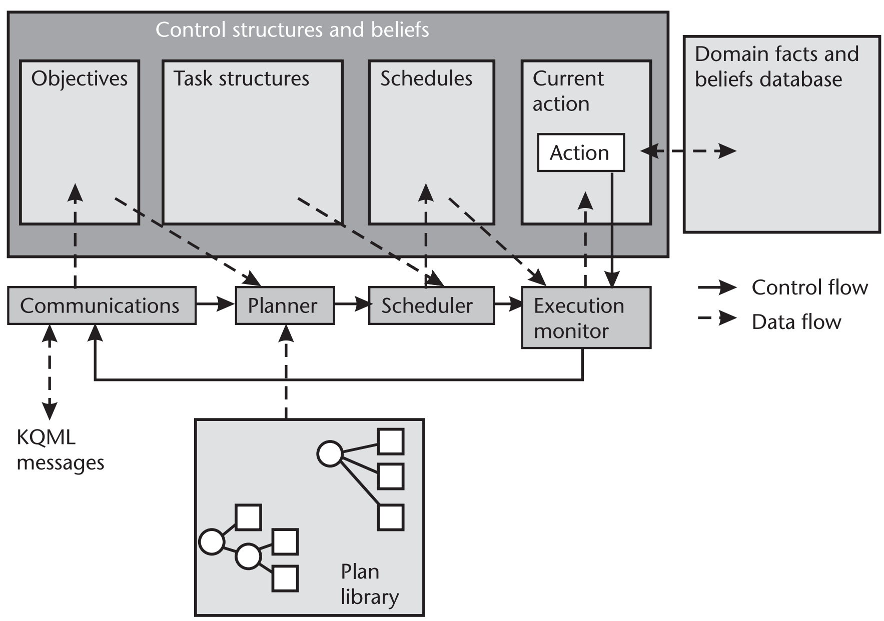


Figure 12.11 RETSINA agent architecture. (*After:* [73].)

- The planning module takes as input a set of goals and produces a plan that satisfies the goals.
- The scheduling module uses the task structure created by the planning module to order the tasks.
- The execution module monitors this process and ensures that actions are carried out in accordance with computational and other constraints.

RETSINA facilitates communication among agents of different types by using *middle* (or *matchmaker*) *agents* to serve as liaisons between agents that request services and agents that provide services. This is an important feature of the RETSINA architecture as interactions are governed by the services that can be provided by the agent in the system and the descriptions provided to the middle agents, rather than by any a priori knowledge maintained by the system developer. This makes the RETSINA model flexible, extensible, and suitable for coalition IF systems as demonstrated in the CoAX experiments [71, 72].

Sensible Agents

The Sensible Agents architecture [74], developed by the Laboratory for Intelligent Processes and Systems at the University of Texas, Austin, provides an environment for heterogeneous distributed agents to operate and communicate with each other and third-party external environments. The Sensible Agents system uses modules to integrate domain-specific actions with domain-independent representation and execution, as shown in Figure 12.12.

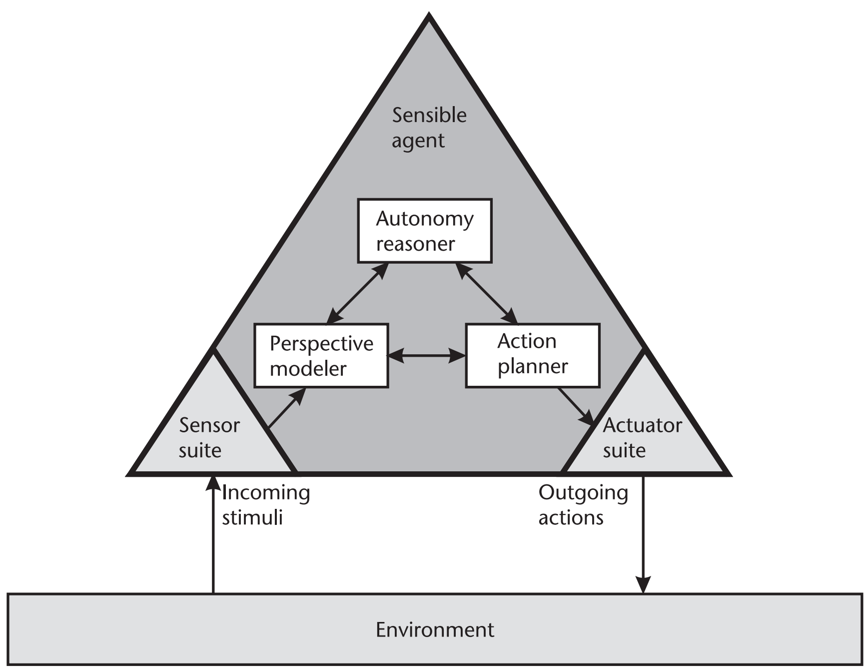


Figure 12.12 Sensible Agents architecture. (*After:* [74].)

The Sensible Agents components are:

1. *Autonomy Reasoner:* Negotiates with other agents to form situation-appropriate decision-making frameworks for collaborative problem solving. This includes adaptively forming agent organizations to achieve the goal at hand [89].
2. *Action Planner:* Produces plans to solve domain problems, executes actions according to it plans, identifies conflicts with the goals and plans of other agents, and suggests strategies to resolve them. Given that sensible agents can dynamically change their organization structure, the Action Planner gives agents the ability to change coordination techniques to match.
3. *Perspective Modeler:* Maintains the agent's local, subjective beliefs about itself and the world. The Perspective Modeler interprets data received from information sources (including the self-agent's sensors and other agents) and changes its models accordingly, using a belief revision process based on the source's reputation, the certainty a source places on the data, and the age of the data. The Perspective Modeler also maintains beliefs about states and events external to the self-agent and predicts the actions of other agents and the environment. This includes temporal issues, addressing the question of how confidence in a piece of information should be discredited or depreciated as time passes. This integration of "information-staleness" factors is important in weighting older, more certain data against more recent, yet more uncertain, information.

The Perspective Modeler maintains reputation values essential for belief revision for each information source. The reputation values maintained by the Perspective Modeler are assessed using two methods: (1) direct trust revision, in which a source's reputation is revised based on its past transaction history with the agent, using dissimilarity metrics to measure the quality of information received, and (2) recommended trust revision, in which a source's reputation is affected by trust information recommended by other agents. This allows the Sensible Agents system to evaluate the reliability of the information sources (including agent-mediated services) available to it and introduce appropriate strategies to deal with unreliable, uncertain, or fraudulent information. This is an important capability for IF systems dealing with information sources with variable reliability or trustworthiness, for instance, in coalition or public networks like the Internet.

ZEUS

ZEUS [90–92] is an open agent architecture developed by the Intelligent Systems Research Group at British Telecommunication Laboratory. The ZEUS toolkit consists of components written in the Java programming language that can be categorized into three libraries:

- Agent Component Library;
- Agent building tools;
- A suite of utility agents comprising name-server, facilitator, and visualizer agents.

The Agent Component Library is a collection of classes that form the building blocks of individual agents and together implement the application-independent functionality required for collaborative agents. The library addresses the issues of communication, representation, and coordination.

For communication, the Agent Component Library provides a performative-based agent communication language (KQML), an asynchronous, socket-based, message-passing system, an editor for describing domain-specific ontologies used to provide the concepts used in the ACL content language, and a frame-based knowledge-representation language for representing domain concepts.

For reasoning and multiagent coordination, the Agent Component Library provides a general-purpose planning and scheduling system suitable for typical task-oriented application domains and the cooperative problem solving inherent to these applications, as well as a coordination engine that controls the social behavior of an agent (i.e., when and how it interacts with other agents).

The application domain influences the functioning of the planner and coordination engines, so the Agent Component Library also provides a library of predefined, reusable coordination protocols (e.g., contract net and various auction protocols); a number of predefined organizational relationships (e.g., superior, subordinate, coworker, and peer); and knowledge representation mechanisms and databases for describing and storing the resources and competencies of an agent.

Together, the components of the Agent Component Library enable the construction of an application-independent (generic) ZEUS agent that can be customized for specific applications by imbuing it with domain-specific resources, competencies, information, organizational relationships, and coordination protocols.

As shown in Figure 12.13, the generic ZEUS agent includes the following components:

- *Mailbox:* This handles communications between the agent and other agents.
- *Message handler:* This processes incoming messages from the mailbox and dispatches them to the relevant agent component.
- *Coordination engine:* This makes decisions concerning the agent's goals (e.g., how they should be pursued and when to abandon them). It is also responsible for coordinating the agent's interactions with other agents using its known coordination protocols and strategies.
- *Acquaintance database:* This describes the agent's relationships with other agents in the society and its beliefs about the capabilities of those agents. The coordination engine uses information from this database when making collaborative arrangements with other agents.
- *Planner and scheduler:* This plans the agent's tasks based on decisions taken by the coordination engine and the resources and task specifications available to the agent.
- *Resource database:* This maintains a list of resources owned by and available to the agent. This database also supports a direct interface to external systems, which allows it to link dynamically to and utilize proprietary databases.
- *Ontology database:* This stores the logical definition of each fact type: its legal attributes, the range of legal values for each attribute, any constraints

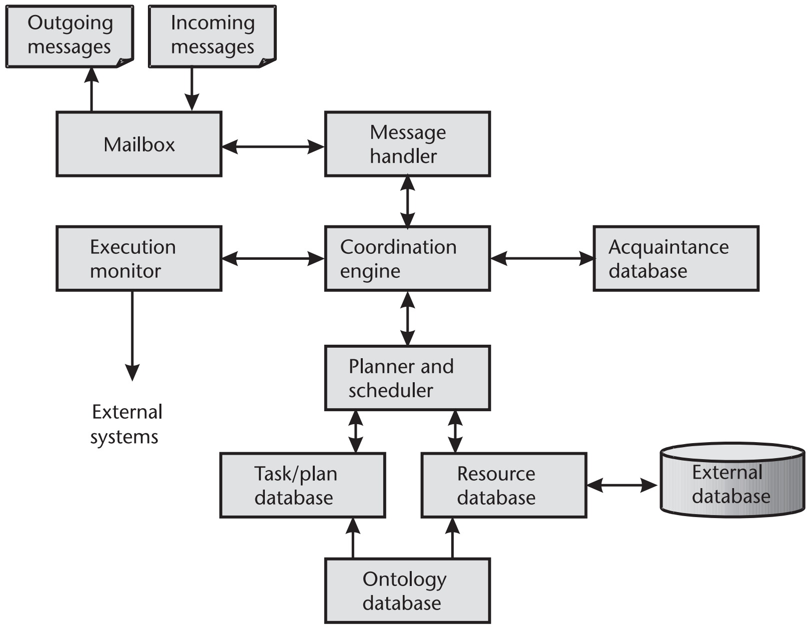


Figure 12.13 Architecture of a generic ZEUS agent. (*After:* [93].)

between attribute values, and any relationships between the attributes of the fact and other facts.

- *Task/plan database:* This provides logical descriptions of planning operators (tasks) known to the agent.
- *Execution monitor:* This maintains the agent's internal clock and starts, stops, and monitors tasks that have been scheduled for execution or termination by the planner/scheduler. It also informs the planner of successful or exceptional terminating conditions of the tasks it is monitoring. In order to manage tasks, the execution monitor also has a direct interface to external systems.

ZEUS provides an open architecture for agent development that can provide the extensibility important for IF systems.

Others

There are many other multiagent systems currently in use, information on which can be found at www.agentlink.org (see, for example, [94]).

References

[1] Antony, R., *Principles of Data Fusion Automation*, Norwood, MA: Artech House, 1995.

[2] Zue, V. W., "Navigating the Information Superhighway Using Spoken Language Interfaces," *IEEE Expert*, October 1995, pp. 39–43.

[3] Zadeh, L., "Fuzzy Logic = Computing with Words," *IEEE Transactions on Fuzzy Systems*, Vol. 4, No. 2, 1996.

[4] Linden, G., S. Hanks, and N. Lesh, "Interactive Assessment of User Preference Models: The Automated Travel Assistant," in A. Jameson, C. Paris and C. Tasso, (eds.), *Proceedings of the Sixth International Conference on User Modeling*, New York: Springer Wien, 1997.

[5] Saracevic, T., A. Spink, and M. Wu, "Users and Intermediaries in Information Retrieval: What Are They Talking About?" in A. Jameson, C. Paris and C. Tasso, (eds.), *Proceedings of the Sixth International Conference on User Modeling*, New York: Springer Wien, 1997.

[6] Vamos, T., "A Strategy of Knowledge Representation for Uncertain Problems: Modeling Domain Expert Knowledge with Patterns," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 25, No. 10, 1995.

[7] Maeder, J., "Human Understanding Limits in Visualization," *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 11, No. 2, 1997, pp. 229–237.

[8] Sugimoto, M., K. Hori, and S. Ohsuga, "A System for Visualizing the Viewpoints of Information and Its Application to Intelligent Activity Support," *IEEE Transactions on Systems, Man and Cybernetics—Part C*, Vol. 28, No. 1, 1998.

[9] Samtaney, R., et al., "Visualizing Features and Tracking Their Evolution," *Computer*, July 1994, pp. 20–27.

[10] Mitchel, M., and G. H. Sundstrom, "Human Interaction with Complex Systems: Issues and Research Approaches," *IEEE Transactions on Systems, Man and Cybernetics—Part A*, Vol. 27, No. 3, 1997.

[11] Murphy, R., "Biological and Cognitive Foundations of Intelligent Sensor Fusion," *IEEE Transactions on Systems, Man and Cybernetics—Part A*, Vol. 26, No. 1, 1996.

[12] Rosenfeld, A., and C. Kak, *Digital Picture Processing*, Vol. 2, New York: Academic Press, 1982.

[13] Gonzalez, R., and P. Wintz, *Digital Image Processing*, Reading, MA: Addison-Wesley, 1977.

[14] Pfleger, K., and B. Hayes-Roth, *An Introduction to Blackboard-Style Systems Organization,* Technical Report KSL-98-03, Knowledge Systems Laboratory, Stanford, CA, 1998.

[15] Nii, H. P., "Blackboard Systems: The Blackboard Model of Problem-Solving and the Evolution of Blackboard Architectures," *AI Magazine*, Vol. 7, No. 2, 1986, pp. 38–53.

[16] Engelmore, R., and A. Morgan, *Blackboard Systems*, Reading, MA: Addison-Wesley, 1988.

[17] Carver, N., and V. Lesser, "The Evolution of Blackboard Control Architectures," *Expert Systems with Applications*, Vol. 7, No. 1, 1994, p. 130.

[18] Erman, L. D., et al., "The Hearsay-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty," *ACM Computing Surveys*, Vol. 12, No. 2, 1980, pp. 213–253.

[19] Hayes-Roth, B., "Architectural Foundations of Real-Time Performance in Intelligent Agents," in J. M. David, J. P. Krivine, and R. Simmons, (eds.), *Second Generation Expert Systems*, New York: Springer-Verlag, 1993.

[21] Abelson, H., G. J. Sussman, and J. Sussman, *Structure and Implementation of Computer Programs*, Cambridge, MA: MIT Press, 1985.

[20] Shoham, Y., "Agent-Oriented Programming," *Artificial Intelligence*, Vol. 60, 1993, pp. 51–92.

[22] Wegner, P., "Concepts and Paradigms of Object-Oriented Programming," *OOPS Messenger*, Vol. 1, No. 1, 1990, pp. 1–87.

[23] Wooldridge, M., *Introduction to Multiagent Systems*, New York: John Wiley & Sons, 2002.

[24] Lambert, D. A., "Advisers with Attitude for Situation Awareness," *Proceedings of the 1999 Workshop on Defense Applications of Signal Processing*, La Salle, IL, 1999.

[25] Dilts, D. M., N. P. Boyd, and H. H. Whorms, "The Evolution of Control Architectures for Automated Manufacturing Systems," *J. Manufacturing Systems*, Vol. 10, No. 1, 1991, pp. 79–93.

[26] Duffie, N. A., and R. S. Piper, "Nonhierarchical Control of a Flexible Manufacturing Cell," *Robotics and Computer-Integrated Manufacturing*, Vol. 3, No. 2, 1987, pp. 175–179.

[27] Hatvany, J., "Intelligence and Cooperation in Heterarchic Manufacturing Systems," *Robotics and Computer-Integrated Manufacturing*, Vol. 2, No. 2, 1985, pp. 101–104.

[28] Vamos, T., "Cooperative Systems Based on Non-Cooperative People," *IEEE Control Systems Magazine*, Vol. 3, 1983, pp. 9–14.

[29] Rosenschein, J. S., and G. Zlotkin, *Rules of Encounter*, Cambridge, MA: MIT Press, 1994.

[30] Jennings, N. R., et al., "Automated Negotiation: Prospects, Methods and Challenges," *International Journal of Group Decision and Negotiation*, Vol. 10, No. 2, 2001, pp. 199–215.

[31] Durfee, E. H., V. R. Lesser, and D. D. Corkill, "Trends in Cooperative Distributed Problem Solving," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 1, No. 1, 1989, pp. 63–83.

[32] Bond, A. H., and L. Gasser, *Readings in Distributed Artificial Intelligence*, San Mateo, CA: Morgan Kaufmann, 1988.

[33] Smith, R. G., and R. Davis, "Frameworks for Cooperation in Distributed Problem Solving," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 11, No. 1, 1980.

[34] Smith, R. G., "The Contract Net Protocol: High Level Communication and Control in a Distributed Problem Solver," *IEEE Transactions on Computers*, Vol. C29, No. 12, 1980, pp. 1104–1113.

[35] Durfee, E. H., "Distributed Problem Solving and Planning," in G. Weiß, (ed.), *Multiagent Systems*, Cambridge, MA: MIT Press, 1999, pp. 121–164.

[36] Wooldridge, M., G. M. P. O'Hare, and R. Elks, "FELINE—A Case Study in the Design and Implementation of a Cooperating Expert System," *Proceedings of the 11th European Conference on Expert Systems and Their Applications*, Avignon, France, 1991.

[37] Durfee, E. H., V. R. Lesser, and D. D. Corkill, "Cooperative Distributed Problem Solving," in E. A. Feigenbaum, A. Barr and P. R. Cohen, (eds.), *Handbook of Artificial Intelligence*, Reading, MA: Addison-Wesley, 1989, pp. 83–147.

[38] Lesser, V. R., and D. D. Corkill, "Functionally Accurate, Cooperative Distributed Systems," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 11, No. 1, 1981, pp. 81–96.

[39] von Martial, F., "Interactions among Autonomous Planning Agents," *Proceedings of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-89),* Amsterdam: Elsevier, 1990.

[40] Durfee, E. H., and V. R. Lesser, "Using Partial Global Plans to Coordinate Distributed Problem Solvers," *Proceedings of the 10th International Joint Conference on Artificial Intelligence (IJCAI-97),* Milan, Italy, 1987.

[41] Durfee, E. H., *Coordination of Distributed Problem Solvers*, Boston, MA: Kluwer Academic, 1988.

[42] Durfee, E. H., "Planning in Distributed Artificial Intelligence," in G. M. P. O'Hare and N. R. Jennings, (eds.), *Foundations of Distributed Artificial Intelligence*, New York: John Wiley & Sons, 1996.

[43] Decker, K. S., "TAEMS: A Framework for Environment-Centered Analysis and Design of Coordination Algorithms," in G. M. P. O'Hare and N. R. Jennings, (eds.), *Foundations of Distributed Artificial Intelligence*, New York: John Wiley & Sons, 1996, pp. 429–447.

[44] Levesque, H. J., P. R. Cohen, and J. H. T. Nunes, "On Acting Together," *Proceedings of the 8th National Conference on Artificial Intelligence (AAAI-90),* Boston, MA, 1990.

[45] Cohen, P. R., and H. J. Levesque, "Teamwork," *Nous*, Vol. 25, No. 4, 1991, pp. 487–512.

[46] Searle, J. R., "Collective Intentions and Actions," in P. R. Cohen, J. Morgan, and M. E. Pollack, (eds.), *Intentions in Communications*, Cambridge, MA: MIT Press, 1990, pp. 401–416.

[47] Cohen, P. R., and H. J. Levesque, "Intention Is Choice with Commitment," *Artificial Intelligence*, Vol. 42, 1990, pp. 213–261.

[48] Jennings, N. R., "Commitments and Conventions: The Foundations of Coordination in Multiagent Systems," *The Knowledge Engineering Review*, Vol. 8, No. 3, 1993, pp. 223–250.

[49] Jennings, N. R., "Controlling Cooperative Problem Solving in Industrial Multi-Agent Systems Using Joint Intentions," *Artificial Intelligence*, Vol. 75, No. 2, 1995, pp. 195–240.

[50] Wittig, T., *ARCHON: An Architecture for Multi-Agent Systems*, Chichester, U.K.: Ellis Horwood, 1992.

[51] Jennings, N. R., et al., "Using ARCHON to Develop Real-World DAI Applications for Electricity Transportation Management and Particle Acceleration Control," *IEEE Expert*, Vol. 11, No. 6, 1996, pp. 60–88.

[52] Perriolat, F., et al., "Using ARCHON: Particle Accelerator Control," *IEEE Expert*, Vol. 11, No. 6, 1996, pp. 80–86.

[53] Tambe, M., "Towards Flexible Teamwork," *Journal of AI Research*, Vol. 7, 1997, pp. 83–124.

[54] Wooldridge, M., and N. R. Jennings, "Formalizing the Cooperative Problem Solving Process," *Proceedings of the 13th International Workshop on Distributed Artificial Intelligence (IWDAI-94),* Lake Quinalt, WA, 1994.

[55] Wooldridge, M., and N. R. Jennings, "The Cooperative Problem Solving Process," *Journal of Logic and Computation*, Vol. 9, No. 4, 1999, pp. 563–592.

[56] Genesereth, M. R., M. Ginsberg, and J. S. Rosenschein, "Cooperation without Communication," *Proceedings of the 5th National Conference on Artificial Intelligence (AAAI-86),* Philadelphia, PA, 1986.

[57] Gasser, L., C. Braganza, and N. Herman, "Implementing Distributed AI Systems Using MACE," *Proceedings of the 3rd IEEE Conference on AI Applications*, 1987.

[58] Gasser, L., C. Braganza, and N. Herman, "MACE: A Flexible Testbed for Distributed AI Research," in M. Huhns, (ed.), *Distributed Artificial Intelligence*, San Mateo, CA: Morgan Kaufmann, 1987, pp. 119–152.

[59] Lewis, D., *Convention: A Philosophical Study*, Cambridge, MA: Harvard University Press, 1969.

[60] Shoham, Y., and M. Tennenholtz, "Emergent Conventions in Multi-Agent Systems," *Proceedings of Knowledge Representation and Reasoning (KR&R-92),* 1992.

[61] Shoham, Y., and M. Tennenholtz, "On the Synthesis of Useful Social Laws for Artificial Agent Societies," *Proceedings of the 10th National Conference on Artificial Intelligence (AAAI-92),* San Diego, CA, 1992.

[62] Shoham, Y., and M. Tennenholtz, "On the Emergence of Social Conventions: Modelling, Analysis and Simulations," *Artificial Intelligence*, Vol. 94, No. 1–2, 1997, pp. 139–166.

[63] Walker, A., and M. Wooldridge, "Understanding the Emergence of Conventions in Multi-Agent Systems," *Proceedings of the 1st International Conference on Multi-Agent Systems (ICMAS-95),* San Francisco, CA, 1995.

[64] Georgeff, M. P., "Communication and Interaction in Multi-Agent Planning," *Proceedings of the 3rd National Conference on Artificial Intelligence (AAAI-83),* Washington, D.C., 1983.

[65] Stuart, C. J., "An Implementation of A Multi-Agent Plan Synchroniser Using Temporal Logic Theorem Prover," *Proceedings of the 9th International Joint Conference on Artificial Intelligence (IJCAI-85),* Los Angeles, CA, 1985.

[66] Labrou, Y., T. Finin, and Y. Peng, "Agent Communication Languages: The Current Landscape," *IEEE Intelligent Systems*, Vol. 14, No. 2, 1999, pp. 45–52.

[67] Luck, M., P. McBurney, and C. Preist, "Agent Technology: Enabling Next Generation Computing (a Roadmap for Agent-Based Computing)," 2002, http://www.ecs.soton.ac.uk/~mml/papers/al2roadmap.pdf.

[68] UMBC KQML Web site, 2003, http://www.cs.umbc.edu/kqml.

[69] Foundation for Intelligent Physical Agents (FIPA) Web site, 2002, http://www.fipa.org.

[70] Allsopp, D. N., et al., "Toward Semantic Interoperability in Agent-Based Coalition Command Systems," *Proceedings of the 1st International Semantic Web Working Symposium (SWWS'01),* July 30–August 1, 2001, http://www.semanticweb.org/SWWS.

[71] Allsopp, D. N., et al., "Coalition Agents Experiment: Multiagent Cooperation in International Coalitions," *IEEE Intelligent Systems*, May/June 2002, pp. 26–35.

[72] Wark, S., et al., "Dynamic Agent Systems in the CoAX Binni 2002 Experiment," *Special Session on Fusion by Distributed Cooperative Agents, Fusion 2003*, Cairns, Australia, July 7–10, 2003.

[73] CMU Intelligent Software Agents Laboratory, "Reusable Environment for Task Structured Intelligent Network Agents (RETSINA)," 2003, http://www.ri.cmu.edu/projects/project_76.html.

[74] UT-Austin LIPS, "UT-Austin LIPS Sensible Agents," 2003, http://www.lips.utexas.edu.

[75] Bradshaw, J., et al., "Representation and Reasoning for DAML-Based Policy and Domain Services in KAoS and Nomads," *Autonomous Agents and Multi-Agents Systems 2003 (AAMAS'03),* Melbourne, Australia, July 14–18, 2003.

[76] Bradshaw, J. M., "Terraforming Cyberspace: Toward a Policy-Based Grid Infrastructure for Secure, Scalable, and Robust Execution of Java-Based Multi-Agent Systems," *Computer*, Vol. 34, No. 7, 2001, pp. 49–56.

[77] Bradshaw, J. M., et al., "Making Agents Acceptable to People," in N. Zhong and J. Liu, (eds.), *Handbook of Intelligent Information Technology*, Amsterdam: IOS Press, 2003.

[78] Bradshaw, J. M., et al., "KAoS: Toward an Industrial-Strength Generic Agent Architecture," in J. M. Bradshaw, (ed.), *Software Agents*, Cambridge, MA: AAAI Press/MIT Press, 1997, pp. 375–418.

[79] Bradshaw, J. M., et al., "Adjustable Autonomy and Human-Agent Teamwork in Practice: An Interim Report on Space Applications," in H. Hexmoor, R. Falcone, and C. Castelfranchi, (eds.), *Agent Autonomy*, Norwell, MA: Kluwer, 2002.

[80] Suri, N., et al., "NOMADS: Toward an Environment for Strong and Safe Agent Mobility," *Proceedings of Autonomous Agents 2000*, Barcelona, Spain, 2000.

[81] Suri, N., et al., "DAML-Based Policy Enforcement for Semantic Data Transformation and Filtering in Multi-Agent Systems," *Autonomous Agents and Multi-Agent Systems Conference (AAMAS'03),* Melbourne, Australia, July 14–18, 2003.

[82] Dartmouth College, D'Agents Web site, 2003, http://agent.cs.dartmouth.edu.

[83] Fabian, I., and D. A. Lambert, "First-Order Bayesian Reasoning," *Proceedings of Artificial Intelligence (AI'98),* Brisbane, Australia, July 14–17, 1998.

[84] DARPA, CoABS Program Web page, 2003, http://www.darpa.mil/ipto/research/coabs.

[85] Agent Oriented Software, JACK Web site, 2003, http://www.agent_software.com/shared/home.

[86] DARPA, Cougaar Web site, 2003, http://www.cougaar.org.

[87] DARPA, Ultra*Log Web site, 2003, http://www.ultralog.net.

[88] BBN Technologies, "Cougaar Architecture Document," version 10.0, 2003, http://www.cougaar.org.

[89] Barber, K. S., *Adaptive Organisations*, Austin: University of Texas, 2002.

[90] Nwana, H., et al., "ZEUS: A Tool-Kit for Building Distributed Multi-Agent Systems," *Applied Artificial Intelligence Journal*, Vol. 13, No. 1, 1999, pp. 129–186.

[91] BT, ZEUS Web site, 2001, http://193.113.209.147/projects/agents.htm.

[92] Azarmi, N., and S. Thompson, "ZEUS: A Toolkit for Building Multi-Agent Systems," *Proceedings of Fifth Annual Embracing Complexity Conference*, Paris, France, April 2000.

[93] Collis, J., and D. Ndumu, "The Zeus Agent Building Toolkit (ZEUS Technical Manual)," British Telecommunications, 1999, pp. 360–361.

[94] Mangina, F., "Review of Software Products for Multi-Agent Systems," 2002, http://www.agentlink.org/admin/docs/2002/MAS.pdf.

CHAPTER 13

Conclusions

Data and information fusion clearly has a critical role to play in future command-and-control systems; it is a key enabler in achieving high-quality situation awareness for optimal decision-making. Data and information fusion is not something that happens in a vacuum, and it should not be decoupled from the decision-making process. This is the reason why we have reviewed many existing models of decision-making and tried to put into perspective information fusion with respect to the overall command-and-control process.

As a discipline, data and information fusion draws together concepts from a wide range of diverse fields: psychology, human factors, knowledge representation, artificial intelligence, mathematical logic, signal processing, and so forth. Most of these aspects are discussed in this book at various levels of depth. In fact, the contribution of this book can be organized into three main categories:

1. Concepts, definitions, and models (Chapters 2–5);
2. Mathematical and logical approaches (Chapters 6–9);
3. Computational aspects of information fusion (Chapters 10–12).

Chapters 2 through 5 provided the common foundation for the analysis and development of IF capabilities. It does so through a review of many concepts, definitions, and models regarding decision-making, situation analysis and awareness, and data and information fusion.

Knowledge, belief, and uncertainty are three key notions in the situation-analysis process (through data and information fusion). Belief and knowledge representation are crucial to transforming data into knowledge. A formalization is necessary to be able to deal with knowledge or uncertainty: a formal framework in which knowledge, information, and uncertainty can be represented, combined, managed, reduced, increased, and updated. Chapter 6 discussed the key notions of knowledge, belief, and uncertainty in relation to information fusion. The potential theoretical frameworks available to model the situation-analysis process can be divided into main categories: qualitative approaches (Chapter 7) and quantitative approaches (Chapter 8). Qualitative approaches seem better suited to reasoning on knowledge, while quantitative approaches are better candidates for uncertainty representation and management. Hence, a good solution for a global modelization of the situation could be a hybrid approach (Chapter 9) mixing quantified evaluations of uncertainty and high reasoning capabilities.

Chapters 10 through 12 reviewed the computational implementations of information fusion. This section of the book addressed the key characteristics of the IF domain and the performance requirements that they impose on IF systems. It reviewed the key elements of computational infrastructure relevant to the design and performance of IF systems, including system architecture, computer networks, software middleware, issues with information sources, and human-computer interfaces. We also considered key concepts in knowledge-based and artificial intelligence systems that have an impact on higher-level fusion processes, including expert systems, reasoning systems, neural networks, and computational complexity. Software architectures that can be used to implement IF systems were reviewed, as were issues associated with the blackboard and multiagent architectures as they can be applied to IF systems.

List of Acronyms

ACL	agent communication language
ACWA	applied cognitive work analysis
ADA	argument-driven action
AFRL	Air Force Research Laboratory
AI	artificial intelligence
ALP	Advanced Logistics Project
API	application programming interface
ATMS	assumption-based truth maitenance system
ATO	air tasking order
BDI	belief, desire, intention
C2	command and control
CAPI	client application programming interface
CBR	case-based reasoning
CC&D	concealment, cover, deception
CDPS	cooperative distributed problem solving
CE	conditional event
CEC	Cooperative Engagement Capability
CERDEC	Communications Electronics Research Development and Engineering Center
CNP	Contract Net Protocol
CoABS	Control of Agent-Based Systems
COA	course of action
CoAX	Coalition Agents Experiments
COMAST	Commander Australian Theatre
COP	common operating picture
CORBA	Common Object Request Broker Architecture
COTS	commercial off the shelf
CPA IUOT	closest point of approach in units of time
CPA	closest point of approach
CPT	conditional probability table
CPU	central processing unit
CSE	cognitive systems engineering
CTA	cognitive task analysis

CWA	cognitive work analysis
DAG	direct acyclic graph
DAML	DARPA Agent Markup Language
DBS	Direct Broadcast System
DDB	dynamic database
DDF	distributed data fusion
DF	data fusion
DFG	data-fusion group
DFS	Data-Fusion Subpanel
DIF	data and information fusion
DIFG	Data-and-Information-Fusion Group
DM	decision-making
DNO	director of northern operations
DoD	Department of Defense
DS	Dempster-Schafer
DSTO	Defence Science and Technology Organisation
DVMT	Distributed Vehicle Monitoring Test Bed
EBA	elimination by aspects
EEE	Expeditionary Sensor Grid Enabling Experiments
EER	extended entity-relationship
ESIOP	environment-specific inter-ORB protocols
EUT	expected utility theory
FASUR	fuse-act situational user refinement
FBE	Fleet Battle Experiment
FIPA	Foundation of for Intelligent Physical Agents
FISST	finite-set statistics
FOCAL	Future Operations Centre Analysis Laboratory
FR	frame of reference
FUS	fusion accrual operator
GBS	Global Broadcast System
GIOP	General Inter-ORB Protocol
GNW	Goodman-Nguyen-Walker
GPGP	generalized partial global planning
HPDS	High Performance Data Store
HTTP	Hypertext Transfer Protocol
HUMINT	human intelligence
I2WD	Intelligence and Information Warfare Directorate
IA	imagery analysts
ID	identification
IDL	Interface Definition Language
IE	information exploitation
IF	information fusion

I	interface
IIOP	Internet Inter-ORB Protocol
IMINT	image intelligence
INT	intelligence
IOR	Interoperable Object Reference
ISL	Intelligent Services Layer
IT	information technology
JAAS	Java Authentication and Authorization Service
JBI	Joint Battlespace Infosphere
JDL DFG	Joint Directors of Laboratories' Data Fusion Group
JDL	Joint Directors of Laboratories
JPG	joint persistent goal
JVM	Java virtual machine
KAoS	Knowledgeable Agent-Oriented System
KBS	knowledge-based system
KID	knowledge/information/data
KIF	Knowledge Interchange Format
KPO	KAoS Policy Ontologies
KQML	Knowledge Query Manipulation Language
KR	knowledge reasoning
KS	knowledge source
L/R	left-eye/right-eye
LAN	local-area network
LEX	lexicographic
LHS	left-hand side
LUS	lookup service
MFAR	multifunction phased array radar
MHYDA	Multiple-Hypothesis Data Association
MP	modus ponens
MSP	multiple-source processing
NDM	naturalistic decision-making
OIL	Ontology Interchange Language
OMA	Object Management Architecture
OMG	Object Management Group
ONT	Office of Naval Technology
OODA	observe, orient, decide, and act
ORB	object request broker
OWA	ordered weighted averaging
OWL	Ontology Web Language
PD	policy directory
PGP	partial global planning
PI	photo interpreters

POP	point of presence
PSCEA	product space conditional event algebra
RAAF	Royal Australian Air Force
RAP	recognized air picture
RCPA	range at closest point of approach
RDECOM	Research, Development, and Engineering Command
RDF	Resource Description Framework
RF	radio frequency
RHS	right-hand side
RLP	recognized land picture
RMP	recognized maritime picture
RM	resource management
RPD	recognition-primed decision
SAAP	situation-analysis application
SA	situation analysis
SAW	situation awareness
SEUT	subjective expected utility theory
SHOR	stimulus, hypothesis, option, and response
SIGINT	signal intelligence
SME	subject matter expert
SOAP	Simple Object Access Protocol
SRK	skill-rule-knowledge
SSP	single-source processing
STDF	state-transition data fusion
TADIL	tactical digital information links
TADMUS	tactical decision-making under stress
TBH	time before hit
TBM	transferable belief model
TBS	Theatre Broadcast System
TCPA	time-to-closest point of approach
TOI	target of interest
TRP	threat reference point
TTS	text to speech
U.S.	United States
UDF	unified data fusion
UE	universal elimination
UI	universal introduction
URI	universal resource identifier
VBS	valuation-based system
VDF	visual data fusion
VF	visual fusion
ViPR	Virtual Planning Room

VM	virtual machine
VN	valuation network
VPN	virtual private network
VR	virtual reality
VSAW	visual situation awareness
WAN	wide-area network
WSDL	Web Service Description Language
XML	Extensible Markup Language
λJDL	unified data-fusion

About the Authors

Éloi Bossé received a B.A.Sc. and an M.Sc. from Laval University in 1981 in electrical engineering and Ph.D. degrees in 1990 from Ottawa/Carleton and Laval Universities. In 1981 he joined the Communications Research Centre in Ottawa, Canada, where he worked on high-resolution spectral analysis and radar tracking in multipath. In 1988 Dr. Bossé was transferred to the Defence Research Establishment Ottawa (DREO) and in 1992 he was transferred to the Defence Research and Development Canada Valcartier (DRDC Valcartier), where he worked on information fusion and decision support. He has published over 130 papers in journals, conference proceedings, and technical reports and has been very active in international (NATO, TTCP) research groups. Dr. Bossé is an adjunct professor at Laval University (Québec) and McMaster University (Hamilton). He now heads the Decision Support Systems Section at DRDC Valcartier.

Jean Roy received a B.A.Sc. and an M.A.Sc. in electrical engineering from Laval University. In 1987 he joined DRDC Valcartier, where he worked on the modeling and simulation of radar wave propagation effects. In 1988, he was posted at the John Hopkins University/Applied Physics Laboratory where he worked on a multisensor integration experiment under a NATO project. In Valcartier in 1989, he initiated a research program in sensor data fusion for situation analysis and then acquired a deep and recognized expertise of this field through the development of a number of test beds and successful applications and through active participation in a number of national and international research groups. From 2000 until 2006, he conducted numerous national and international R&D activities covering advanced aspects of data fusion, situation analysis, and command and control. He is the author or coauthor of 73 publications.

Steve Wark received a B.Sc. (Hons) from the University of Melbourne and was later awarded a Ph.D. in physics from the University of Melbourne in 1992. He joined the Microwave Radar Division at the Defence Science and Technology Organisation, Australia, in 1996. There he worked on modeling microwave radar and multisensor fusion systems and contributed to the development of the ATTITUDE multiagent system for radar control and data fusion. In 2000 Dr. Wark transferred to the Command and Control Division, where he works on the Future Operations System Analysis Laboratory (FOCAL) project, information fusion systems, and situation awareness displays for command and control. This has involved him in the development of distributed multiagent systems for information fusion and novel interfaces for situation awareness such as embodied conversational agents, immersive battlespace displays, and 3D animation. He has been actively involved in a number of national and international research activities, including the TTCP and the DARPA Coalition Agents eXperiment.

Index

M

N

O

P