8

A Unifying Multimodel Taxonomy and Agent-Supported Multisimulation Strategy for Decision-Support

Levent Yilmaz[1] and Andreas Tolk[2]

[1] Department of Computer Science and Software Engineering, Auburn University, Auburn, AL, U.S.A. yilmaz@auburn.odu
[2] Frank Batten College of Engineering & Technology, Old Dominion University, Norfolk, VA 23529, U.S.A. atolk@odu.edu

Summary. Intelligent agent technology provides a promising basis to develop next generation tools and methods to assist decision-making. This chapter elaborates on the emergent requirements of decision support in light of recent advancements in decision science and presents a conceptual framework that serves as an agent-based architecture for decision-support. We argue that in most decision-making problems, the nature of the problem changes as the situation unfolds. Initial parameters, as well as scenarios can be irrelevant under emergent conditions. Relevant contingency decision-making models need to be identified and instantiated to continue exploration. In this paper, we suggest a multi-model framework that subsumes multiple submodels that together constitute the behavior of a complex multi-phased decision-making process. It has been argued that situation awareness is a critical component of experience-based decision-making style. Perception, understanding, and anticipation mechanisms are discussed as three major subsystems in realizing the situation awareness model.

8.1 Introduction

Decision science involves understanding cognitive decision processes, as well as methods and tools that assist decision-making (Davis et al. 2005). Significant amount of research has been conducted on decision theory and associated processes. This chapter focuses on how intelligent agent technology can provide basis for a unified synthesis of deductive, practical, and experience-based mechanisms to constitute a multi-level decision support system. In this context, logical, practical, and experience-based decision-making are analogous to rational choice model (von Neumeann and Morgenstern 1953), heuristics and biases (Tversky and Kahneman 1974), and naturalistic decision-making (Klein 1997).

Decision-making involves making tradeoffs among competing attributes or goals, analyzing complex situations within constraints of time and resources,

L. Yilmaz and A. Tolk: *A Unifying Multimodel Taxonomy and Agent-Supported Multisimulation Strategy for Decision-Support*, Studies in Computational Intelligence (SCI) **97**, 193–226 (2008)
www.springerlink.com

projecting into future state of the environment despite uncertainty, and making judgments, even if they are heuristic (Zachary 1998). The evolution of decision-making theory can be viewed as a steady withdrawal from the rational choice model to bounded rationality, and most recently to naturalistic decision-making (NDM) theory. While rational choice model (Parsons and Wooldridge 2002) involves the maximization or optimization of the expected utilities, bounded rationality emphasizes the constraints of time, resources, and cognitive capacities. Bounded rationality worldview involves the use of heuristics and biases (Tversky and Kahneman 1974) to capture cognitive shortcuts used in decision-making. Naturalistic decision-making, on the other hand, is based on the premise that humans assess situations by using prior experience. Zsambok (1997) argues that situation assessment and experience-based decision-making is more appropriate than option generation under conditions that involve uncertain and dynamic environments, shifting or competing goals, time stress, and ill-structured problems. Note that decision-making styles can shift between analytic, heuristic, and experience-based several times within a single problem (Hamm 1988). Furthermore, Hammond (1986) demonstrates that task features, such as complexity of the task structure, ambiguity, and form of representation, determine the decision-making style. More specifically,

1. In most realistic decision-making scenarios, the nature of the problem changes as the situation unfolds. Initial parameters, as well as scenarios can be irrelevant under emergent conditions.
2. Our knowledge about the decision problem being studied may not be captured by any single decision-making style. Instead, the available knowledge is viewed as being contained in the collection of all possible decision-making experiments that are plausible given what is known and what is learned.
3. Dealing with uncertainty is paramount to making decisions within the context of complex evolving phenomena. Dynamic adaptivity in decision-making styles is necessary to deal with emergent conditions or evolving decision-making process in a flexible manner.

Based on these observations and a recent recommendation (Davis et al. 2005), the contributions of this chapter are two-fold.

1. An agent-supported multisimulation approach that aims to simultaneously analyze multiple alternative Course of Actions (COAs), and, if necessary, update the scenario to deal with new phases of problem.
2. Delineation of the design considerations for the agent-based naturalistic decision-making.

Intelligent agents are proven to be useful in decision-making, especially within the context of game theory (Parsons and Wooldridge 2002) and mechanism design (Wooldridge 2002). Designing mechanisms refers to developing agent interaction protocols, called strategies, which satisfy desirable properties such as Pareto efficiency, stability, and social welfare maximization among a collection of agents. Power (Power 2002) describes how model-based decision support can be supported by simulation systems in general and agent-based simulation systems in particular. Tolk (2004) enumerates a comprehensive list of military decision-making functions for which agents can provide valuable support.

Proper simulation-based decision support methodologies that are consistent with the way experts use their experience to make decisions in field settings could improve modeling for Course of Action (COA) analysis. Each COA is simulated faster than real time, the results are collected, and COA analysis can be performed. Additional requirements for simulation systems when being used for this sort of analysis are summarized in (Tolk and Kunde 2003). Exploring the effectiveness of alternative COAs at the tactical and operational levels requires dynamic updating, branching, and simultaneous execution of simulations, potentially at different levels of resolution. We propose a strategy in integrating human-centered decision-making with multisimulation-based COA analysis. Three modes are identified:

1. Human-in-the-loop with naturalistic decision-making approach,
2. Agent-augmented naturalistic decision-making,
3. Agent-based naturalistic decision-making.

The first mode involves an operator that interacts with the simulation to choose alternative COAs based on situational awareness gathered from the results obtained during the simulation. The second mode aims to augment the decision-making process of the operator with intelligent agents that carry out routine tasks that pertain experience the situation in a changing context, reasoning about and diagnosing the situation to make recommendations for plausible COAs. In the third mode, intelligent agents replace the operator, and they perform the perception, understanding, and anticipation functions to model the situation awareness capabilities of the operator.

In many situations simulation specialists build a simulation and then conduct the special study and report their results to management. Evan and Olson (2002) discuss examples of how simulation has been used to support business and engineering decision-making. Their examples are prototypical for our findings: simulation systems without agents designed for reliable decision support are not universal tools, but special – and often expensive – means of operations research. The methods and technology described in this chapter help to make simulation systems flexible and reliable enough to become decision support systems.

The rest of the chapter is structured as follows. Section 8.2 presents the major decision-making styles, the decision-making process, and intelligent agents. Section 8.3 enumerates a set of requirements for next generation for intelligent simulation-based decision support systems based on the nature and types of emergent problems in various application domains. Section 8.4 introduces the macro-architecture for the proposed decision-support system.

We show how alternative decision styles can be supported within a multi-level view of the decision-making problem. Section 8.5 focuses on the design of situation-aware agents that are capable of augmenting humans to make experience-based decisions. It also presents selected research domains for the next generation of such systems. Section 8.6 presents a case study to substantiate the utility of the presented decision support approach. Finally, Sect. 8.7 concludes by discussing potential avenues of research and application.

8.2 Decision Science and Intelligent Agents

For our approach, we view decision-making as a cognitive reasoning process. The first subsection presents the characteristics of major decision-making styles. The second subsection overviews the process and its phases. The last subsection characterizes the role that intelligent agents can play to support each phase of the process (Fig. 8.1).

8.2.1 Decision-making Styles

Decision-making is viewed as a process that entails two distinct activities. The first one is to decide what state of affairs is desired and second how this state will be achieved. In modern decision science, there are mainly three decision-making styles.

- **Rational Choice Model (RCM):** This model of decision-making emerged in such diverse fields as economics, political science, management science, and operation research.

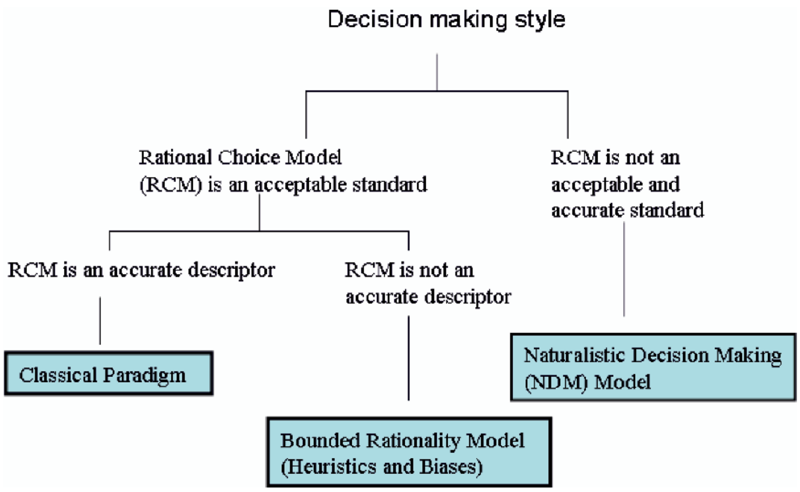

Fig. 8.1. Decision-making styles

von Neumeann and Morgenstern (1953) introduced the idea that rational choice should maximize expected subjective utility. From the perspective of game theory, this classic approach to decision analysis can be viewed as an analytical approach that optimizes the outcome of a decision. Building on the rationality principle, game theory has been applied to various problems (Geyer and van der Zouven 1998, Shubik 1964). However, evidence exists that classical game theory fails in cases where opponents have different value systems (Knight et al. 1991). Different types of game theories (e.g., sequential games, repeated games (Banks and Sundaram 1990, Leimar 1997)), differential games, evolutionary games, and hyper-games (Fraser and Hipel 1984), have been applied in the context of RCM.

- **Bounded Rationality (BR):** In making decisions, humans operate within a complex and often changing environment with limited cognitive capabilities, time, and other resources. Hence, decision-making is only rational within the bounds imposed on decision makers (Simon 1982).

  Tversky and Kahneman (1974) identified a number of heuristics and biases that humans use to make decisions. These studies aim to bring classical and analytic decision theorists into conformity with findings in cognitive psychology. The premise of bounded rationally is based on the observation that heuristics (Davis et al.2005) often yield cost-effective compared to classical methods in terms of time and mental effort. Furthermore, changes in the environment will cause the judgment to be obsolete.
- **Naturalistic Decision-making (NDM):** The empirical work of Gary Klein (1997) on expert behavior in high-pressure environments resulted in a new school of thought in decision-making. The NDM paradigm argues that people assess situations by using prior experience and knowledge.

  Furthermore, unlike RCM and BR decision-making styles, in NDM situation assessment is considered to be more important compared to option generation. Hence, the approach is to perform pattern matching to match observed problem facets to the mental model of the problem formed by the decision maker. Sokolowski (Sokolowski 2003) discusses the application of NDM for agent supported decision-making.

8.2.2 Intelligent Agents

In the context of this chapter, we use the definition of Ferber (1999), who defines software agents as entities that are capable of acting in purely software and/or mixed hardware/software environments

1. can communicate directly with other agents,
2. are driven by a set of goals, objectives and tendencies,
3. possess skills to offer services,
4. perceive its environment, and
5. can generate autonomous behavior that tends toward satisfying its objectives.

An overview of additional views is documented in Murch and Johnson (1998). Furthermore, we assume that the environment will be

– not-accessible (versus accessible),
– stochastic (versus deterministic),
– dynamic (versus static),
– sequential (versus episodic),
– and continuous (versus discrete)

to represent the environments specifies in the last section for realistic decision-making problems.

In this context, we understand agents as autonomous software modules with perception and social ability to perform goal-directed knowledge processing over time, on behalf of humans or other agents in software and physical environments. When agents operate in physical environments, they can be used in the implementation of intelligent machines and intelligent systems and can interact with their environment by sensors and effectors. The core knowledge processing abilities of agents include: reasoning, motivation, planning, and decision-making. The factors that may affect decision-making of agents, such as personality, emotions, and cultural backgrounds can also be embedded within agents. Additional abilities of agents are needed to increase their intelligence and trustworthiness. Abilities to make agents intelligent include anticipation (pro-activeness), understanding, learning, and communication in natural and body language. In this chapter, we advocate the use of (1) practical situation-aware agents that diagnose the situation via perception, understanding, and anticipation capabilities and (2) agents that facilitate simulation-based analysis of alternative COAs.

8.3 Requirements for Developing Computational Frameworks for Decision Support

Advances in decision science and the nature of problems being tackled impose new requirements on next generation decision-support systems.

8.3.1 Decision Styles and Problem Domain Characteristics

The nature of the decision style further imposes constraints on the decision-making models within a multi-model. Table 8.1 depicts the three main decision styles discussed in the earlier section along with the problem domain characteristics they target.

For instance, the RCM style provides an acceptable and accurate framework for problems in which actors, their preferences, utilities for actions, and the outcomes are well-defined. The problem is expected to be stable, and the number of options and players are small. Furthermore, the cognitive limitations of the decision maker and the lack of resources are not considered

Table 8.1. Features of decision-making styles

Decision-making style	Problem Domain Characteristics	Tool Design Features
Rational Choice Model	1- Well-defined problems 2- Low uncertainty 3- Stable environment 4- Small number of players and options 5- Time is not a parameter/factor	a- High-level design templates for various recurring problems b- Graphical interfaces for specifying utilities, actors, preferences, and outcomes
Bounded Rationality	1- Resource limitations (cognitive, computational etc.) 2- Time stress is a factor 3- Medium level certainty 4- Incomplete information about the environment	a- Models that encode heuristics and biases such as availability, representativeness, and anchoring and adjustment heuristics [1]
Naturalistic Decision Making	1- Ill-structured problems 2- Uncertain, dynamic environments 3- Shifting, ill-defined, competing goals 4- Action/feedback loops 5- Time stress and high stakes 6- Multiple players 7- Organizational goals and norms are factors (Zsambok 1997)	a- Perceiving situations in an environment b- Matching perceptions against learned experiences c- Understanding the overall situation via comprehension mechanisms d- Exploring possible outcomes by emulating mental simulation d- Anticipating future state(s) of the environment before making a decision

to inhibiting factors in decision-making. Having decision-making tools that enable formal specification of the structure of decision-making problem is feasible under these conditions.

Therefore, interactive tools that provide graphical facilities to capture options, preferences, utilities etc. can be useful. On the other hand, NDM decision-making style is introduced for problem domains that are ill-defined. The level of uncertainty in the environment leads to shifting and possibly competing goals. The characteristics of the domain are common in decision-making environments where there is a time stress, high stakes, and continuous action/feedback loops.

To support experts in making decisions in such environments, a decision-support system needs to provide facilities to augment pattern matching for situation recognition, understanding of the overall situation from the perceived disconnected elements, and make projection to potential future states. The projection phase simply involves tool support for mental simulation of the plausible actions.

8.3.2 Multisimulation in Support of Naturalistic Decision-making

Many real-world decision-making phenomena can not be modeled by one single model; rather, they require the use of a set of complementary decision-making models representing multiple perspectives that are able to describe the whole process possibly at different resolutions and phases when applied orchestrated (Bigelow and Davis 2003, Ören (1987, 1991, 2001), Zeigler et al. 2000, Yilmaz and Ören 2004). We distinguish contribution of multimodels and multisimulation that are dealt with in the following in more detail.

Multimodels

Models are purposeful abstractions of reality. Complex challenges require the use of several different views – or abstractions – to cover the full spectrum. This motivates the use of multimodels. While one big model is feasible, it is likely that this model would be as complicated as the real problem and the modeling would not result in any advantage. Several smaller models combined with each other overcome both shortcomings. Basic definitions and brief explanations of the envisioned multimodel types – as they are shown in Table 8.2 – follow here:

A multimodel is a modular model that subsumes multiple submodels that together constitute the behavior of a complex multi-phased decision-making process. A multimodel encapsulates several aspects of reality (i.e., submodels) in one model. For instance, conflict resolution problems discussed in (Yilmaz et al. 2006) emphasized the importance of dropping the notion of decision-making using a single conflict management procedure for the management and resolution of complex conflicts. Tolk (2004) discusses similar issues for agent mediated decision support in the military domain. The discussion on the use of multi-aspect, multi-stage, multi-resolution multimodels implies a certain type of conflict dynamics; that is, a set of stages in the process associated with proper conflict management procedures for each stage.

Note however, that as a situation unfolds, the parameters of the decision and payoff matrices, the state space of the problem, the attitudes, and preferences may change. Therefore, the time path of a decision-making process should map onto a time path of decision-making styles embedded within the models. Critical questions that need to be answered include the issues pertaining to the mechanism by which decision-making styles are selected, when and how shifts occur in updating multimodels, and to whom the judgment to determine the shift should be given. In single aspect models only one aspect of reality can exist at a given time (to be represented by an appropriate submodel) and transitions can occur from one submodel to another one under monitored conditions. Special cases of multimodel formalism are the metamorphic model and the evolutionary model.

A metamorphic model has a fixed number of submodels with a predetermined transition order between the submodels. The transition conditions can include the processes of the metamorphosis.

Table 8.2. Synopsis of the envisioned mulimodel (MM) formalism

Based on	Additional Criteria				Type of Multimodel (MM)
Structure of submodels	Number of submodels active at a given time			Only one	Single Aspect MM (sequential MM)
				Two or more	Multi-aspect MM
	Variability of structure (variability of number of submodels)	Static			Static-structure MM
		Dynamic (Dynamic structure MM) (Variable structure MM)	Number of submodels	Extensible	Extensible MM
				Depends on model's stage	Multistage MM
			Alternations of submodels	No	Non-mutational MM
				Yes	Mutational MM
					Evolutionary MM
Behavior (activation) of submodels	Nature of knowledge to activate the submodels	Constraint-driven			Constraint-driven MM (Adaptive MM)
		Pattern-directed MM (Metamorphic MM)	Submodel selection is cyclic	No	Acyclic MM
				Yes	Cyclic MM
		Goal-directed			Goal-directed MM (Exploratory MM)
	Location of knowledge to activate the submodels	Within the MM (internal activation of submodels)			Active MM (Internally activated MM)
		Outside the MM (External activation of submodels)			Passive MM (Externally activated MM)

An evolutionary model can have several submodels. The number of submodels at the beginning may be fixed or unknown. Subsequent submodels are variant models of their predecessors. The transitions from a submodel to another one can be achieved as rule-based, pattern-directed, or goal-directed activities. Evolution, being an irreversible change in an open system, is important in the study of decision-making. Mutations, pathological or not, –including social mutations– can be modeled as evolutionary models.

A multi-aspect model consists of several submodels where two or more submodels can be active at a given time. Since each submodel can represent an aspect of reality, several aspects –even contradictory ones– can be represented at the same time. The multi-aspect modeling methodology appears to be very promising to encapsulate several aspects of phenomena and their mutual influences. In a multi-aspect model, submodel(s) inactive at a given time are latent or dormant submodels. In decision-making, for instance, anticipatory study of the effects of latent submodels may deter later catastrophes.

A multistage model is a set of variable number of submodels that can be used to represent reality at different emerging stages of a system. In conventional decision-making studies, one model is used for the duration of the lifespan of a system. However, in social systems, the fluidity of the situation may necessitate exploring with more than one model at every emerging stage of the analysis.

As shown in Table 8.2, there are various design decisions in multimodel design. Alternative names are given in parentheses.

Based on the completeness of submodels, there are two cases: (1) one can either know all the submodels at the beginning i.e., at modeling stage, or (2) there can be emergent conditions where the need for additional submodels.

Based on the number of active submodels, one needs to consider two cases: (1) only one submodel is active at a given time or (2) two or more submodels are active at a given time. Simultaneous existence of two or more model components would facilitate simulation of multiple aspects of the phenomena under study.

Based on the location of information necessary for the activation of submodels there are two cases: the necessary information can be (1) within the submodels or (2) it can be external to submodels.

The transitions between submodels can be goal-directed (goal directing the submodel transition rule and goal-directed submodel transition mechanism should be specified) or pattern-directed. Natures of information necessary for the activation of submodel(s) entail the selection conditions of a submodel.

Pattern-directed activation entails a meta-pattern to guide (1) selection of known submodels and (2) request of new submodels corresponding to an interruption of the decision-making process using a human-in-the-loop mechanism.

Multisimulation

We define multisimulation as a simulation of several aspects of reality in a study. It includes simulation with multimodels, simulation with multi-aspect models, and simulation with multistage models. Simulation with multimodels allows computational experimentation with several aspects of reality; however, each aspect and the transition from one aspect to another one are considered separately. (In special cases, multimodels can be metamorphic models or evolutionary models). Simulation with multi-aspect models (or multi-aspect simulation) allows computational experimentation with more than one aspect of reality simultaneously. This type of multisimulation is a novel way to perceive and experiment with several aspects of reality as well as exploring conditions affecting transitions. While exploring the transitions, one can also analyze the effects of encouraging and hindering transition conditions. Simulation with multistage models allows branching of a simulation study into several simulation studies; each branch allowing to experiment with a new model under similar or novel scenarios.

In our approach, there can be multiple strategy components that are qualified at the time of decision-making. Each different strategy component characterizes a distinct aspect. Multisimulation can be used to branch out multiple simulations, where each simulation uses a specific component configured with an exclusively selected strategy component. Similarly, multiple distinct stages of the problem can be qualified at a given point in time during the simulation by virtue of the evaluation of an updating constraint. In such a case multisimulation enables branching multiple distinct simulations each one which generates the behavior of distinct plausible stage within the problem domain.

Multisimulation with multimodels, multi-aspect models or multistage models needs mechanisms to decide when and under what conditions to replace existing models with a successor or alternative.

Staging considers branching to other simulation studies in response to a scenario or a phase change during experimentation. Graphs of model families facilitate derivation of feasible sequence of models that can be invoked or staged. More specifically, a graph of model families is used to specify alternative staging decisions. Each node in the graph depicts a model, whereas edges denote transition or switching from one model to another. Figure 8.2 depicts the components of the abstract architecture of a possible multisimulation engine.

A meta-simulator is a scheduler that generates staged composition of models by traversing the model stage graph and coordinates their simulation and staging within distinct simulation frames. Each frame simulates a distinct subset of models derived from the model stage graph. Note however, that not all staged compositions are feasible or useful. Hence, the meta-simulator needs to consult with the model recommender before model staging to determine if emergent trigger or transition condition in the simulation is consistent with

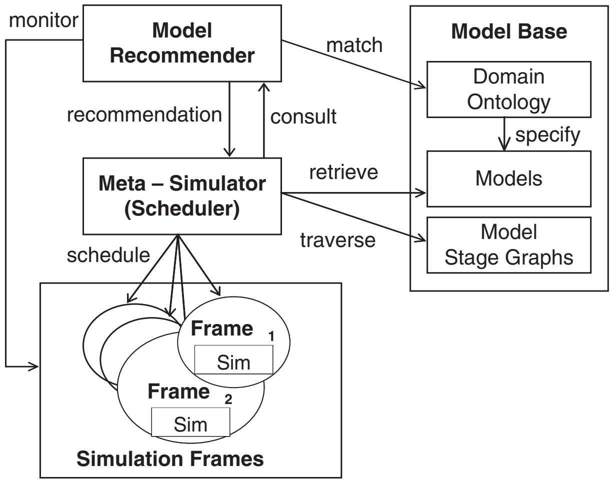

Fig. 8.2. Abstract components of the multisimulation engine

the precondition of the model to be staged. More than one model in a family can qualify for staging; in such cases separate simulation frames need to be instantiated to accommodate and explore plausible scenarios. Given a collection of models (or more generally, a family of models), a stage graph can be generated automatically by an optimistic approach that connects every available node (model) to every other node within the domain of problem. The edges in a model stage graph denote plausible transitions between models as the problem shifts from one stage to another. One can consider each model as a separate conflict management protocol (i.e., compromise over actions, compromise over outcomes, negotiation, and mediation) or a phase in the conflict process (i.e., escalation, resolution), where a phase (i.e., resolution) can constitute alternative models (i.e., mediation, negotiation, third-party intervention).

The subsets of staged models can be identified by traversing and enumerating the graph in some order (i.e., depth-first). Infeasible paths may be due to an unreachable node, or it may result due to conflicts between the transition condition and precondition of the target model. Infeasible paths due to incompatible sequences of models are common. Each edge (say from n_i to n_j) indicates that there is some legitimate solution that includes n_i followed by n_j; yet, it does not imply that every solution containing n_i followed by n_j is legitimate. As argued above, each model in a family of models is associated with a precondition. A precondition denotes the conditions required for a model to be instantiated. Hence, the feasibility of staging a successor model depends on the satisfiability of its precondition (relevance) by the condition of the transition and the post-condition of the predecessor model. As a result, not all enumerated staged sequences of model components are feasible.

Model recommendation in multisimulation can simply be considered as the exploration of the model staging space that can be computed by a reachability analysis of the graph. There are two modes for the usage: (1) offline enumeration of paths using the graph and performing a staged simulation of each model in sequence one after the other, unless a model staging operation becomes infeasible due to conflict between the transition condition and the precondition of the successor model and (2) run-time generation of potential feasible paths as the simulation unfolds. In both cases, an online model recommender plays a key role to qualify a successor model. The first case requires derivation of sequence of models using a traversal algorithm. The edges relate families of models. Therefore, the actual concrete models, the preconditions of which satisfy the transition condition need to be qualified, since transition to some of these model components may be infeasible due to conflict between a candidate model and inferred situation. Identifying such infeasible sequences is computationally intractable; otherwise, it would have been possible to determine if the conjunction of two predicates is a tautology by using a polynomial time algorithm.

Experience in the component-based simulation paradigm, however, indicates that for most model components preconditions are simple. Hence, it is possible to eliminate some models that violate the transition condition. For the remaining possible transitions it is possible to select one of the three strategies: (1) omit all difficult qualification conditions, (2) decide on an edge-by-edge basis which specific models of a model family to include, and (3) include all difficult edges. Omitting all difficult associations between transitions and model preconditions is conservative. This strategy excludes all infeasible models. The cost is the exclusion of some feasible edges. Hand-selecting those associations between transition conditions and models facilitate inclusion of feasible models. Nonetheless, the costs involved with this level of accuracy are the potential human-error and effort needed to filter out infeasible models. Choosing to include all difficult associations is liberal, in that it ensures inclusion of all feasible models. The cost is the inclusion of some infeasible models, hence the inclusion of some undesirable staged compositions that enforce models to be simulated even when their qualification conditions are violated. Nevertheless, it is possible to screen out such models using an online model recommender.

The second more ambitious yet flexible approach is to delay the enumeration process until a model is qualified at run-time. Runtime generation of feasible staging using the graph of model families requires monitoring and evaluation of transition conditions as the simulation unfolds. A planning layer connected to simulator would be capable of identifying, qualifying, and, if necessary, selecting and instantiating a model based on the specified preferences and options. Furthermore, in the case of an impasse or lack of knowledge on preferences among qualifying model switch strategies, a planning layer can guide exploring alternative contexts (games) in some order. The metascheduler follows the recommendations made by the planner to instantiate distinct simulation frames.

Candidate models and associated simulations are maintained by focus points. A focus point manages branch points in the simulation frame stack. Suppose that a goal instance (i.e., stage transition condition) is at the top of the stack. If only a single model qualifies for exploration, then it is pushed onto the stack. Yet, if more than one model matches the condition, a simulation focus point is generated to manage newly created simulation branching (discontinuity) points. Each one of these simulation focus points has his own context. When a path is exhausted, the closest focus point selects the next available model to instantiate the simulation frame or return to the context that generated the focus point. As simulation games are explored, a network of focus points is generated. Determining which focus point should be active at any given time is the responsibility of the meta-scheduler. When more than one model is qualified, then scheduler needs to decide which one to instantiate. Control rules can inform its decision. Three steps involve in deploying a new simulation frame in such cases: matching, activation, and preference. The matching step should both syntactically and semantically satisfy the request. The activation step involves running a dynamic set of rules that further test the applicability of models with respect to contextual constraints. Finally, the preference steps involve running a different set of rules to impose an activation ordering among the active frames.

8.4 Agent-Based Intelligent Decision Support – A Unifying Framework

We present a unified exploratory multisimulation technology, which suggests a simulation world-view shift. After evaluating general observations, we will focus on aspects of situation awareness and experience-based reasoning.

8.4.1 Architectural Constraints for a Unifying Framework

Experimentation with exploratory multisimulation contrasts sharply with establishing a base-case model and scenario to perform sensitivity and factor analysis, where the user is interested in understanding the variance of predictions under priory selected configurations. Exploration involves performing computational experiments under uncertainty to gain intuition about possible outcomes, if decisions on using certain models based on emergent conditions are true. The premise of exploratory multisimulation is based on the view that the results of a simulation are not viewed as a prediction of what we would expect to occur, but rather the results of a computational experiment. By making recommendations for staging and branching to alternative models as well as scenarios, dynamic simulation update mechanisms enable exploratory multisimulation.

As exploration is based on a number of such recommendations, our knowledge about the problem being studied cannot be captured by any single model, scenario, or experiment. Instead, the domain knowledge needs to be viewed as being contained in the collection of possible modeling experiments and ensemble of models that are plausible given what is known or learned during the simulation experiment. Multisimulation subsumes multi-resolution simulation, where entities are capable of simultaneously operating at different levels, while maintaining consistency at each level of abstraction.

Embedding such a decision-centered simulation methodology into operational systems is a significant challenge. Operational necessity and integration concepts are discussed have been discussed among others by Daly and Tolk (2003). In decision-making situations, operators should be able to identify and investigate the impact of COAs to evaluate the effectiveness of decisions. To this end, a decision support system based on exploratory multisimulation technology that will operate within the framework of NDM. NDM is emerging as a field of research, providing a descriptive view of how people behave in dynamic, uncertain, and often fast-paced environments. This model focuses on experienced agents, working in complex, uncertain conditions, who face personal consequences for their actions. Figure 8.3 depicts the organizational layout of the components that constitute the solution. In the following sections we will clearly identify the technologies, (basic, applied research, or

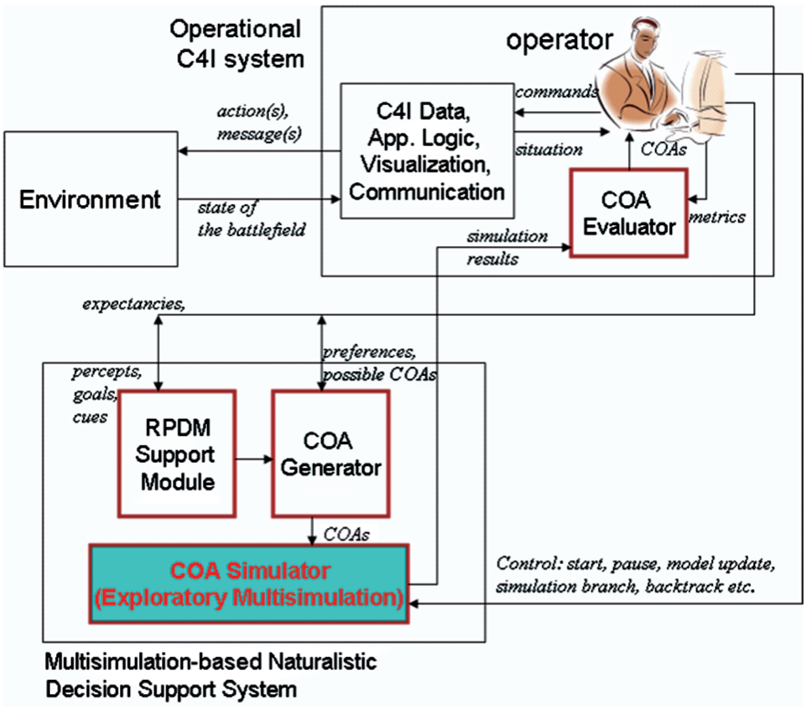

Fig. 8.3. Architecture of the decision-support system

exploratory development) forming the proposed solution. The premise of the approach is that decision makers (i.e., operators) need tools to augment their decision-making process. Such a decision support tool, however, needs to be consistent with how experts use their experience to make decisions in operational settings. To this end, we choose an NDM framework, which provides a descriptive view of how people behave in dynamic, uncertain, and often fast paced environments. NDM focuses on experienced agents, working in complex, uncertain conditions, who face personal consequences for their actions (Zsambok 1997). Development and insertion of this technology into operational systems forms the basis of the technical objective. The novel aspects of the approach are based on the following technologies.

- Exploratory multisimulation that realize the mental simulation component of Recognition-Primed Decision (RPD). Dynamic model and simulation updating is a novel strategy that enable evaluating multiple COAs via simulation branching.
- A computational model for situation-aware RPD, which is a special case of NDM, and
- Agent-supported COA generation based on practical agent reasoning technology.

The operational C4I system shown in Fig. 8.3 embodies a multisimulation-based decision support subsystem that aims to evaluate various COAs on behalf of the operator. The operator interprets the situation in consultation with the computational RPD model to generate valid and accurate percepts based on his experience. RPD component provides a computational mechanism for situation recognition and pattern recognition. The output of the RPD Making (RPDM) module is a set of goals, expectancies, and clues. This output is evaluated by the operator to generate a set of preferences and/or action(s) to be carried out by the simulation component of the decision support system. The preferences and actions are used by the COA generator component that deploys an agent–based planning algorithm to generate a set of plans. These plans are then simulated by the exploratory multisimulation engine. The simulation results are then evaluated interactively by the operator using the COA filter that uses the provided performance metrics.

8.4.2 Situation Awareness and Experience-based Reasoning

The decision support system is designed to support three modes of operation – operator-driven, agent-augmented, and agent-supported multisimulation.

Mode 1: Operator-driven Multisimulation

The first mode is the operator acting on his/her own interpretation of the situation to devise COAs. The strategy is as follows:

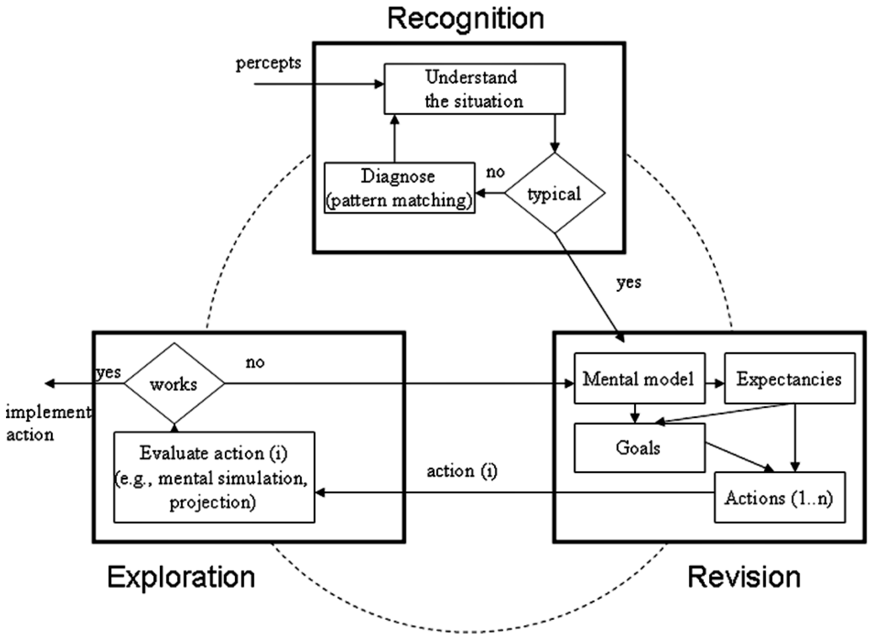

Fig. 8.4. Experience-based reasoning

1. Observe the C4I system
2. Perceive and understand the situation
3. Anticipate/project future status
4. Decide on plausible COAs
5. Update the simulation model to predict outcomes under alternative COAs.

The first three steps in the above strategy refer to diagnosis of the situation. The diagnosis activity is initiated in response to uncertainty about the nature of the situation. The life cycle for experience-based decision-making of the operator involves three main stages as shown in Fig. 8.4: the recognition, revision, and exploration phases. The architecture embodies an extended version of the RPD model (Klein 1997). The model, which is based on Recognition Primed Decision Model, is an example of NDM, and it attempts to emulate what people actually do under conditions of time pressure, ambiguous information, and changing conditions. According the architecture, the sensory input is processed by the experience the situation component to perceive the elements of the situation. If the situation is prototypical, the NDM submodel instantiates a skeleton mental model, from which expectancies and goals can be derived. Simple if-then rules can be used to derive plausible actions based on goal-action pairs. These goal-action pairs are based on prior experience, and they are encoded within the mental model. If the observed situation and perceived inputs are not categorized to be prototypical, then a diagnosis (i.e., pattern matching) procedure that synthesizes the features of the percepts to causal factors is enacted to facilitate comprehending the situation until a prototypical or analog case is identified.

The exploration phase of the life cycle requires evaluating the selected action. Humans often perform mental simulation of the possible outcomes if and when the decision is implemented. In our system, the evaluation is performed via multisimulation. If the action is found to be irrelevant to the goal as a result of the projection or mental simulation, the mental model is further revised to either update the goal or identify a different action. The challenge in this mode is in providing a front-end interface to multisimulation to pause, update, reconfigure, and restart the simulation with the new parameters, models, and even scenarios. In this mode, the operator will browse through the available COA in the library or query based on the perceived situation. The recognition and revision phases are manual, whereas evaluation is supported by multisimulation. However, the update operations over the multisimulation are still manual.

Mode 2: Agent-Augmented Multisimulation

In this mode, the operator is active in perceiving the situation, understanding it, and projecting the status for decision-making. However, unlike the operator-centered mode, intelligent agents are responsible for dynamically updating the model. Our design strategy for enabling this operation is based on an ontology-driven approach that provides introspective access to dynamic object patterns. More specifically, the multisimulation provides the facilities that

1. establish a self-representation of the system using dynamic object pattern ontologies,
2. offer means by which this representation can be updated, and
3. assure that the manipulations to the self-representation influence the behavior of the system.

In effect, the system's self-representation is connected to the behavior of the actual application. Hence, the structure of an application is divided into two components: (1) system level and (2) meta-system level. The system level includes the stable components of the model, application level software objects, and the structural and behavioral dependencies between the components it includes. The meta-system level includes components that are subject to change, and the ontology is based on the dynamic object pattern. The meta-system level provides an interface to facilitate configuring or updating the ontology that subsequently drive the simulation. The meta-system level provides three categories of functions:

- **Reflection:** System level can access information about the system via facilitator agents associated with the system. This information can then be used to guide the behavior of the system.

- **Introspection:** System level can access and update the parameters of existing meta-simulation entities. This enables seamless and transparent update of the behavior of the system, since the behavior is influenced by the meta-system entities.
- **Intercession:** System level can change, exchange, insert, or remove meta-system entities and their connections to the system level. This feature enables dynamically including or inserting new components into the application at run-time

Mode 3: Agent-Supported Multisimulation

This mode of the decision support system involves the exclusive use of agents, and there is no operator in the loop. That is, the recognition, revision, and exploration components of the decision-making lifecycle are supported by intelligent agents. This mode requires further research on developing means to facilitate situational awareness for implementing the recognition and revision components of the decision-making life cycle. The recognition, revision, and exploration phases of the situation awareness layer, shown in Fig. 8.3, suggest three main functional areas that revolve around a mental model of the problem domain. More specifically, a well-defined mental model provides

1. knowledge about the concepts, attributes, associations, and constraints that pertain to the application domain,
2. a mechanism that facilitates integration of domain elements to form an understanding of the situation, and
3. a mechanism to project to a future state of the environment given the current state, selected action, and the knowledge about the dynamics of the environment.

Endsley (1995) defines situation awareness as the perception of elements in a particular environment within time and space, the comprehension of their meaning and the projection of their status in the near future.

8.5 Considerations for the Design of the Situation Awareness Subsystem

Situation awareness, as depicted here, provides a set of mechanisms that enable attention to cues in the environment, expectancies regarding future states. In realistic settings, establishing an ongoing awareness and understanding of important situation components pose the major task of the decision maker. Therefore, situation awareness is the primary is basis of the decision-making process in experience-based decision-making process (i.e., NDM).

Situation awareness, the mechanisms of which are shown in Fig. 8.5, is an important cognitive skill that is essential for expert performance in any field

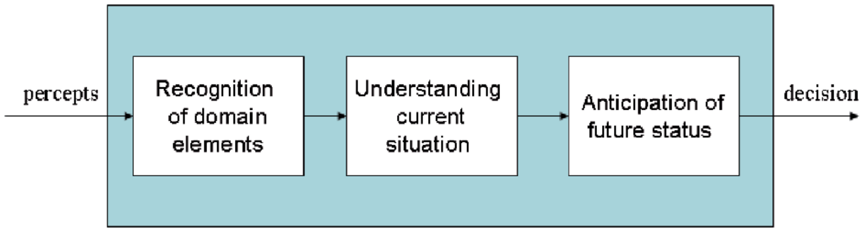

Fig. 8.5. Mechanisms for situation awareness

involving complexity, dynamism, uncertainty, and risk. The percepts are the interfaces to the environment; through them, the environment is perceived. The failure to perceive a situation correctly may lead to faulty understanding. Ultimately, this misunderstanding may degrade an individual's ability to predict future states and engage in effective decision-making (Gaba and Howard 1995). It is therefore an essential part of the NDM.

8.5.1 Perception

The way we perceive reality affects our feelings, decisions, and actions. Since Plato's allegory of the cave explained in Book 7 of "The Republic," it is well known that perception is very important (Bloom 1968). Wikipedia encyclopedia explains philosophy of perception as follows:

> "The philosophy of perception concerns how mental processes and symbols depend on the world internal and external to the perceiver. Our perception of the external world begins with the senses, which lead us to generate empirical concepts representing the world around us, within a mental framework relating new concepts to preexisting ones. Because perception leads to an individual's impression of the world, its study may be important for those interested in better understanding communication, self, id, ego –even reality." (Wikipedia (Phi-Per) 2004)

There are two types of perception, i.e., external and internal perceptions. Philosophy of perception is concerned with external or sensory perception.

> "External or sensory perception, tells us about the world outside our bodies. Using our senses of sight, hearing, touch, smell, and taste, we discover colors, sounds, textures, etc., of the world at large.
> Internal perception tells us what's going on in our bodies. We can sense where our limbs are, whether we're sitting or standing; we can also sense whether we are hungry, or tired, and so forth." (Wikipedia (Phi-Per) 2004)

Both types of perceptions can involve thought processes. Introspection is the detailed mental self-examination of feelings, thoughts, and motives.

Table 8.3. Categories of perception

	Current images of Past or current state	Future state
Others (people and/or events)	Perceived image of others and events	Behavioral anticipation of others and events
Self (decision maker(s), supporters, followers, and/or events related with one's own side)	Perceived image of self and/or events related with one's own side	Behavioral anticipation of self and/or events related with one's own side

> "In psychology and the cognitive sciences, perception is the process of acquiring, interpreting, selecting, and organizing sensory information. Methods of studying perception range from essentially biological or physiological approaches, through psychological approaches to the often abstract 'thought-experiments' of mental philosophy." (Wikipedia (Phi-Per) 2004)

A categorization of perception is given in Table 8.3. Perception of an entity at a time t gives an image of it at that time. At time t, we can refer to the perception as the current perception (or current image), if there is only one perception.

However, at a time t, based on the perspective, there may be different interpretations of an entity, hence several perceptions. From now on, for the sake of simplicity, unless it is specified otherwise, current perception (or current image) is considered to be unique. Current image can refer to external perceptions; hence it can be about others (people, groups, nations, events, facts, etc.). When current image refers to internal perceptions, then it is about the self (or own group of decision makers, supporters, followers; and/or events related with one's own side.) Current images may refer to past, current, or future states. There can be several current images, at different times $t_i,\ i = 1,\ 2,\ 3, \ldots,\ n$; until future becomes current.

This is similar to for example, seven day meteorological forecasts. At each day, there can be a forecast of a certain day until that day. And due to the variability of meteorological conditions, the forecasts may be different. When that specific day occurs, what we experience is the current image of the current state. If we are interested to interpret past events, current images of a certain past may be defined. However, there can be several images of a certain past based on the points of views of the people involved. Current images of (past, current, or future states) can reflect possibly different interpretations of the current perceptions. Hence, especially in a conflict situation, the opponents may even have antagonistic interpretations of the same situation. Furthermore, emotions such as anger affect the disposition of the decision makers.

8.5.2 Understanding

Understanding or comprehension of the situation is based on synthesizing the perceived disjoint elements to form a coherent representation of the entity, the elements of which are observed. For instance, the tactical commander of a military unit needs to comprehend that the appearance of enemy aligned in a specific pattern and in a particular location depicts certain specific objectives. Augmenting decision makers by providing capabilities that integrate perceived domain elements to facilitate comprehension of the situation requires taking the following design consideration. In the study of natural phenomena, the role of simulation is often cited as "to gain insight" which is another way of expressing "to understand." Understanding is one of the important philosophical topics. From a pragmatic point of view, it has a broad application potential in many computerized studies including program understanding, machine vision, fault detection based on machine vision as well as situation assessment. Therefore, systematic studies of the elements, structures, architectures, and scope of applications of computerized understanding systems as well as the characteristics of the results (or products) of understanding processes are warranted.

Dictionary definitions of "to understand" include the following:

– to seize the meaning of,
– to accept as a fact, believe,
– to be thoroughly acquainted with,
– to form a reasoned judgment concerning something,
– to have the power of seizing meanings, forming reasoned
– judgments,
– to appreciate and sympathize with, to tolerate,
– to possess a passive knowledge of a language

The following is a good starting point for the specification of the scope of machine understanding:

> "... if a system knows about X, a class of objects or relations on objects, it is able to use an (internal) representation of the class in at least the following ways: receive information about the class, generate elements in the class, recognize members of the class and discriminate them from other class members, answer questions about the class, and take into account information about changes in the class members." (Zeigler 1986)

From this point of view, knowing and computerized understanding can be taken as synonyms. However, one should remark here that knowing (something, somebody, some event, etc.) refers to the result of the process of acquiring knowledge and not the knowledge processing activity required to know. A system A can understand an entity B if three conditions are satisfied (see Fig. 8.6):

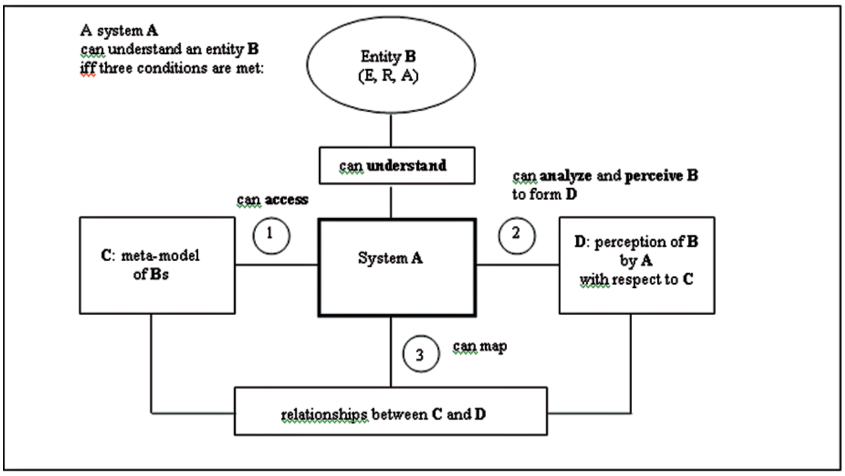

Fig. 8.6. Elements of an understanding system

1. A can access C, a meta-model of Bs. (C is the knowledge of A about Bs.)
2. A can analyze and perceive B to generate D. (D is a perception of B by A with respect to C.)
3. A can map relationships between C and D.

Therefore, an understanding system needs to have the following three basic elements: a meta-model of the entities to be understood, a perception element and an analyzer and a comparator to map a perception of an entity to be understood with the meta-model.

8.5.3 Role of Anticipation in Decision-Making

Anticipation is an important characteristic of intelligence. Pro-active behavior requires anticipatory abilities. Without anticipation a system can only be reactive; but a dead frog can also be reactive. A seminal work on anticipatory systems is the one written by Rosen (1985). A brief introduction to and serious concerns about anticipation follows:

> "Strictly speaking, an anticipatory system is one in which present change of state depends upon future circumstances, rather than merely on the present or past. As such, anticipation has routinely been excluded from any kind of systematic study, on the grounds that it violates the causal foundation on which all of theoretical science must rest, and on the grounds that it introduces a telic element which is scientifically unacceptable. Nevertheless, biology is replete with situations in which organisms can generate and maintain internal predictive models of themselves and their environments, and utilize the predictions of these models about the future for purpose of control in the

> present. Many of the unique properties of organisms can really be understood only if these internal models are taken into account. Thus, the concept of a system with an internal predictive model seemed to offer a way to study anticipatory systems in a scientifically rigorous way." (Rosen 1985)

A systematic review of 12 definitions of anticipation is available from Berkley Initiative in Soft-Computing, Special Interest Group (BISC-SIG) in Anticipatory Systems with the following warning:

> "The following 12 definitions, or descriptions, of anticipation should be understood as working hypotheses. It is hoped and expected that the knowledge community of those interested in anticipation will eventually refine these definitions and suggest new ones in order to facilitate a better understanding of what anticipation is and its importance for the survival of living systems."(BISC-SIG 2004)

An important aspect from the point of view of BISC-SIG is the emphasis on soft computing requirements in anticipation. Perception ability is a required characteristic of agents. Hence, they can be designed to perceive current state of self and others. They can also be designed to create current images of future states. An anticipatory system is a system whose next state depends on its current state as well as the current images of its future states. This definition is a radical departure from the original definition given by Rosen (1985): *"An anticipatory system is a system determined by a future state. The cause lies in the future."* Nonetheless, our definition is in line with the following definition also given by Rosen:

> "An anticipatory system is a system containing a predictive model of itself and/or of its environment that allows it to change state at an instant in accord with the model's predictions pertaining to a later instant." (Rosen 1985)

However, we would like to stress the distinction on dependency of next states on current images of future states rather than the future value of the states.

Perception requires mechanisms that enable interpretive capabilities. Perception invariably involves sensory qualities, and introspection entails accessing sensations and perceptions the agent would introspect. Perceptions are derived as a result of interpretation of sensory inputs within the context of the current world and agent's self model. The prototype inference, orientation accounting, and situational classification mechanisms (Sallach 2003). could be used to realize the interpretation capabilities of an agent. The interpretation process results in perceptions. An anticipatory agent needs to deliberate upon perceptions through introspection and reflection to anticipate.

Introspection is deliberate and attentive because higher-order intentional states are themselves attentive and deliberate. An introspective agent should have access mechanisms to its internal representation, operations, behavioral potentials, and beliefs about its context. Reflection uses the introspective

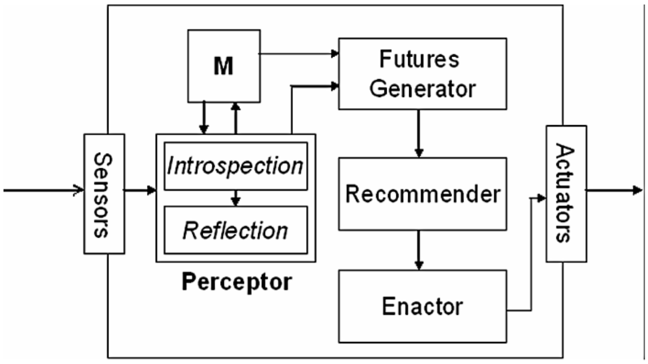

Fig. 8.7. Basic components for anticipatory agents

mechanisms to deliberate its situation in relation to the embedding environmental context. These features collectively result in anticipation capabilities that orient and situate an agent for accurate future projections. Figure 8.7 presents interpretation and introspection as critical components within the micro-architecture of an anticipatory agent. A computationally anticipatory agent needs to incorporate interpretation facilities as a precursor to (1) comprehend and draw accurate inferences about the world, (2) have social pragmatism by considering the likely responses of others in its context in response to a communication or act, and (3) have situational definition [40] as a direct input to action recommendation. An anticipatory agent uses a domain model M, as the internal representation of the environment and agent's self in order to project to the future. The model and the anticipation that results from the introspection and reflection processes are used to derive a number of realities by the futures generator. The generator is a function that maps environmental parameters and past vector of states onto a set of future states of the environment.

Naturally, an inductive process would be used to realize the function, as the generation of future plausible realities (environmental contexts) results in a set of new models that vary from each other based on assumptions on different plausible events or possible interactions between the environment and the agent itself. This perspective is consistent with the definition of anticipation process that is given in (BISC-SIG 2004). According to the definition, anticipation (1) is a realization within the domain of possibilities and/or (2) involves the generation of a multitude of dynamic models and the resolution of their conflict. As such, the recommender subsystem is responsible for evaluating alternative anticipated models and to decide on choosing a specific strategy based on the goals and motivations of the agent. Next, a recommender system should select a desirable future state upon which the agent would make decisions and react using its enactor component.

Developing anticipatory agents with run-time recommenders is difficult, because interpretation of emergent conditions requires mining the state of the simulation to recognize situations within the domain theory (schema) of an application. That is plausible and desirable future states need to be qualified based on the motives and goals of the agents. Learning takes place as recommendations are made. Adaptive models that assume certain discernible patterns in the recommendations may be used to discover situations and associated relevant models so as to reinforce qualification of specific future states based on previous experience. Various domain specific representational issues and inadequacies make this very difficult for particular applications. One form of representational inadequacy pertains to intrinsic difficulty of determining (and utilizing) the features that are potentially relevant for model selection. Another form of representational inadequacy involves on deciding the right level of detail. A major difference between traditional deliberative agents and an anticipatory agent is that an anticipatory agent makes guesses about the future state of the environment to guide its behavior, whereas conventional deliberative agents make their decisions based on the observed conditions within the current context.

8.5.4 Additional Research Domains

So far, we focused on decision makers as individuals. In the netted organizations supporting complex systems of today, this is no longer the rule. What is needed are good models for shared situation awareness, which in turn request good communication models between decision makers, representing agents, or supporting agents. Tolk and Gaskins (2006) published some tentative results in the light of the development of the Global Information Grid, a highly interconnected web-based infrastructure to support operations in the defense and security domains.

Recent work shows the challenge of building human behavior models in complex and cognitive domains. Cannon-Bowers et al. (1993) introduced the concept of shared mental models to describe the fluid, implicit interaction often observed in successful teams. Teams must predict and cope with task difficulty and change by altering their strategies. Shared mental models are the mechanisms that help teams make sense of situations and facilitate coordinated team performance and decision-making. Team members typically do not share a single mental model. Rather, there are likely multiple mental models co-existing among team members. Such shared mental models are characterized by a variety of factors including the characteristics of the team, the nature of the task, the type of equipment, and the interaction among the team members. However, these factors are generally categorized as either task work or teamwork mental models. Task work mental models include the understanding of activities and action sequences of the task, whereas teamwork mental models refer to the understanding of communication needs, compensatory behaviors, performance monitoring, and internal coordination strategies of the team. It has been shown that shared mental models relate positively to team processes, in particular decision-making, as well as performance.

Furthermore, team processes were found to fully mediate the relationship between shared mental models and performance. Although empirical support is limited, emerging findings suggest that appropriate team mental models have positive effects on team processes and effectiveness. Such findings suggest that the development of shared mental models is a promising leverage point for distributed learning techniques aimed improving team effectiveness. How these research results can be incorporated into agents in the light of these findings, is the subject of current research.

One of the most critical aspects of distributed decision-making environments is the role of information transfer between team members, i.e., communication. Researchers have studied the communication process for many years, and have constructed models to depict that process. Since Shannon and Weaver (1949) proposed one of the earliest models of the communication process based on telephone communications in 1949, research has focused on how information is transmitted and what are disturbing factors, such as noise or external events. A critical component of the model is noise, which may serve to confound the message. Noise may consist of any unwanted stimulus that renders the message less comprehensible. For example, on the modern battlefield, noise may occur because of conflicting information, irrelevant information, or competing sources of information.

Since Shannon and Weaver's early work, other models of the communication process have been proposed, addressing the weaknesses of the five-step process. Some of these models reflected the increasingly complex nature of team communication. As time went on, network models of communication emerged, further increasing the complexity (and therefore the model validity) of representations of the human communication process. When dealing with distributed decision-making in teams, these models must replace the presumably perfect connections between communicating agents. However, as with shared situational awareness, the research on this topic is just in its beginnings.

8.6 Case Study

In this study, a multi-resolution coordinated mission for Unmanned Air Vehicles (UAV, which are airplanes that are flying without a human pilot on board) is being considered. The C4I system is represented with yet another simulation developed in Matlab/Simulink environment. The model is called MUAV, which is a collaborative UAV testbed (Niland 2006). The agent-augmented multisimulation based decision-making scenario examined in this scenario involves an operator that interacts with both the MUAV software that represents the C4I system and the multisimulation. Figure 8.8 presents the major components of the simulation, which is based on the High Level Architecture (HLA) and its common information infrastructure,

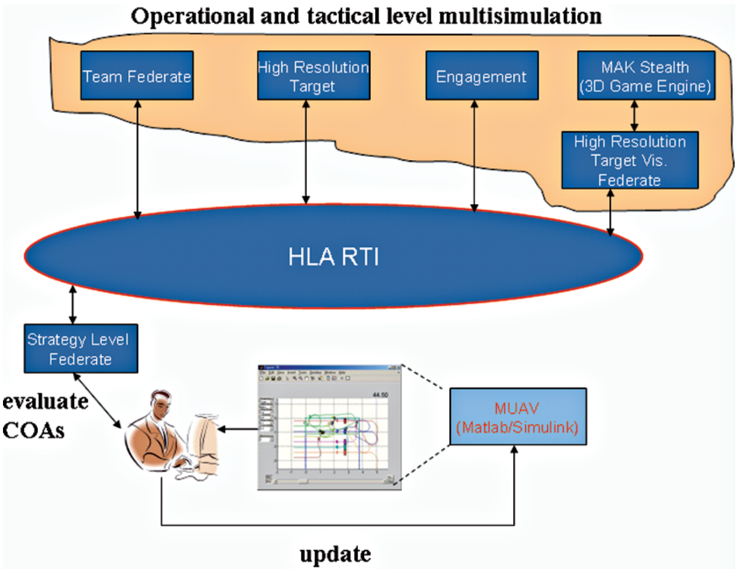

Fig. 8.8. UAV coordination mission study

the Run-Time Infrastructure (RTI). HLA is an international standard for distributed simulation (IEEE 1516–2000).

The scenario starts at the low resolution with a number UAVs sweeping an area that contains multiple targets. Targets are classified as low resolution (i.e., tank battalions) and high-resolution entities (i.e., individual tanks). Individual UAVs can detect and destroy high-resolution entities such as tanks. However, in the case of a detection of an aggregate entity such as a battalion, UAVs aggregate into teams by virtue of a team formation strategy to establish multi-resolution entities, called Teams. The strategy level federate uses inputs of from the operator to (1) cluster entities to identify aggregates and (2) uses agent based team formation protocol, called contract-net, to establish teams. Next, applicable strategies or COAs are recommended by the operator so that teams at the operational and tactical simulation level can be configured by the appropriate behavioral model. If more than one COA is applicable then multiple simulations are initiated, as shown in Fig. 8.9. The multiple simulations at the operational level include behavior from High-resolution Team (HRT), the engagement that represents the tactical strategy used to engage with the targets at the high resolution simulation, the targets, and the visualization behavior. For the low-resolution on higher tactical level, a Matlab/Simulink simulation was used. For the high-resolution simulation of HRT, the MAK Stealth (3D Game Engine) off-the-shelf software was used.

The tactical federate uses intelligent agent support to configure the HRT of a given operational simulation with any of the following strategies. As such, the Coordination Strategy lets the COA protocol vary independently

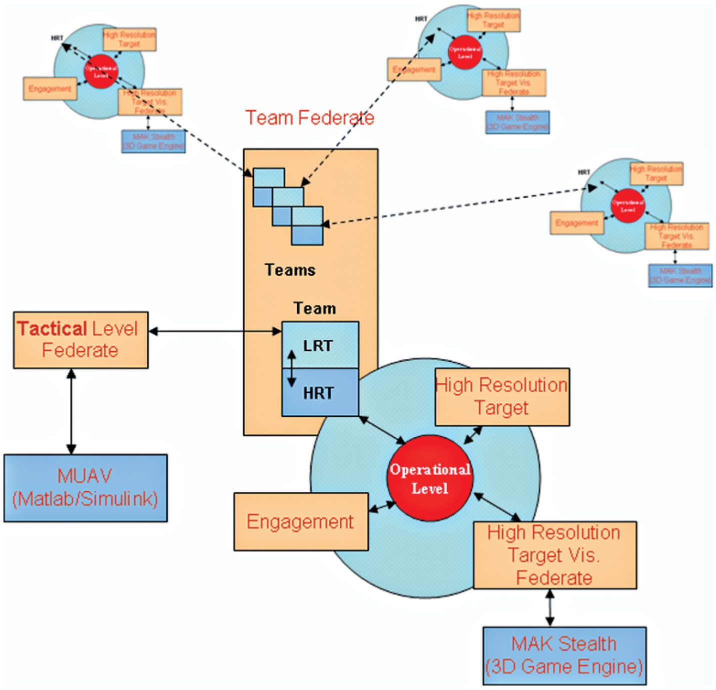

Fig. 8.9. Multiple simulations at the operational level

of the team that uses it. The possibility of configuring teams with multiple COAs enables performing multisimulation, where each simulation facilitates exploring the efficiency and effectiveness of a specific COA. For instance, in our case study we considered two COAs for sweeping the battlefield: Region and Fringe strategy. Figure 8.10 presents the rules of the region strategy, whereas Fig. 8.11 illustrates the rules of Fringe Point Strategy.

In our study, staging from one strategy to another based on the observed conditions is as critical as initiating multiple simulations in the first place. Fig. 8.9 presents demonstrates the connections between HRT and Strategy Federate via a Low Resolution Team (LRT) that coexists with HRT encapsulated within a Multi-resolution Team entity. LRT uses observer agents to monitor the HRT to evaluate the state of the engagement. Corresponding to the time path of the change of a problem should be a time path of the appropriate submodel families. But, the question is what should be the sequence of this shift pattern of models of family? Or should there be trigger mechanisms indicating when a shift should occur? The tactical federate uses an anticipator agent defined in terms of a Bayesian model to decide the correct strategy and instructs the Multi-resolution team to reconfigure its HRT with the selected strategy.

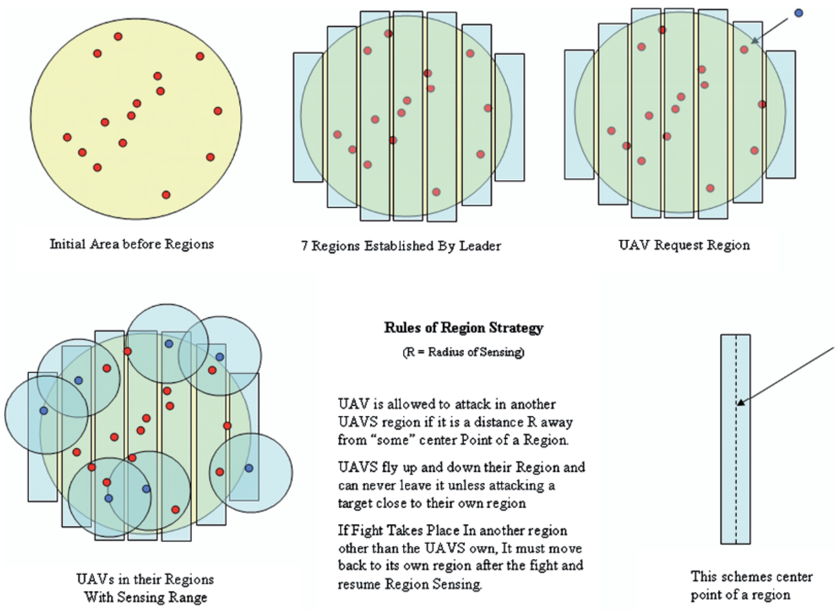

Fig. 8.10. Presentation of the rules of the region strategy

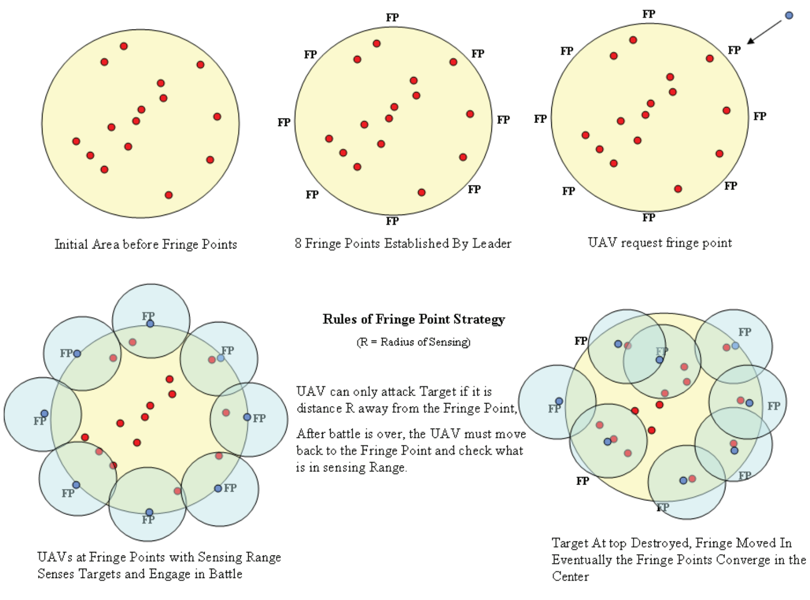

Fig. 8.11. Illustration of the fringe point strategy

8.7 Conclusions

The use of intelligent agents in decision support is common. However, existing work on agent-based decision support mostly focuses on rational choice models, where agents are programmed to seek optimal utilities during negotiation and bargaining. Recent advancements in decision science suggest that pursuing synthesis of alternative decision styles within a coherent framework could have profound effects on the approach to decision-support. Empirical studies of Eisenhardt and Zbaracki (1992) involving mid-to high-level strategic decision makers found that context and environmental circumstances effect the decision-making style employed by the decision makers. In most decision-making scenarios, the nature of the problem changes as the problem unfolds. Initial parameters, as well as scenarios can be irrelevant (i.e., real-time training scenarios) under emergent conditions. Relevant contingency models need to be identified and instantiated to continue exploration. Another aspect that is currently under research, in particular in the ontological community and composability researcher, is the question how model families and multimodels that comprise multi-resolution models (which are models that vary in scope, structure, or resolution) can be used in an orchestrated way in support of decision support. First results are summarized in (Tolk et al. 2007, Tolk et al. 2008), but the research and discussion is ongoing.

In this paper, we suggested a multi-model framework that that subsumes multiple submodels that together constitute the behavior of a complex multi-phased decision-making process. Three distinct decision styles are embedded within a horizontal agent-based decision-support system architecture. Strategies and design considerations for developing experience-based, practical reasoning, and deductive rational choice models of decision-making are examined. It has been argued that situation awareness is a critical component of Naturalistic Decision-making style that is based on experience based reasoning. Perception, understanding, and anticipation mechanisms are discussed as three major subsystems in realizing situation awareness model. These methods and technology will contribute to make agent-based simulation a valuable tool for decision support systems, as their support will become more flexible, credible, and configurable to the users needs.

References

Banks JS and Sundaram RK (1990) Repeated games, finite automata, and complexity. Games and Economic Behavior, 2, pp 97–117

Bigelow JH and Davis PK (2003) Implications for model validation of multi-resolution, multi-perspective modeling (MRMPM) and exploratory Analysis. California, CA: RAND Publication

BISC-SIG (2004) Special Interest Group in Anticipatory Systems. BISC (Berkeley Initiative in Soft Computing) Available online via <http://www.anticipation.info/> (accessed on April 9, 2004)

Bloom A (1968) The republic of Plato (translated with notes and an interpretive essay), New York, NY: Basic Books

Cannon-Bowers J, Salas E and Converse S (1993) Shared mental models in expert team decision making. In Castellan NJ (Ed) Individual and group decision making: Current issues, Hillsdale, NJ: Erlbaum, pp 221–246

Daly JJ and Tolk A (2003) Modeling and simulation integration with network-centric command and control architectures, Fall Simulation Interoperability Workshop, IEEE CS Press

Davis PK, Kulick J and Egner M (2005) Implications of modern decision science for military decision-support systems. California, CA: RAND Publications

Eisenhardt KM and Zbaracki JM (1992) Strategic decision making. Strategic management journal, 13, pp 17–37

Endsley MR (1995) Measure of situation awareness in dynamic systems. Journal for Human Factors, 37(1), pp 65–84

Evan JR and Olson DL (2002) Introduction to Simulation and Risk Analysis (2nd Edition), Prentice Hall, pp 161–179

Ferber J (1999) Multi-agent systems: An introduction to distributed artificial intelligence. New York, NY: Addison-Wesley

Fraser MN and Hipel WK (1984) Conflict analysis models and resolutions. Amsterdam: Elsevier Science Publishers

Gaba DM and Howard SK (1995) Situation awareness in anesthesiology. Journal for Human Factors, 37, pp 20–31

Geyer F and van der Zouven J (1998). Bibliography on sociocybernetics (3rd edition). (http://www.unizar.es/sociocybernetics/bibliografia.html, 1998)

Hamm RM (1988) Moment by moment variation in expert's analytic and intuitive cognitive activity. IEEE Transactions on Systems, Man, and Cybernetics,18(5), pp 757–776

Hammond KR (1986) A theoretically-based review of theory and research in judgement and decision making. Technical report CRJP 260, Center for Research on Judgement and Policy, University of Colorado

IEEE 1516–2000 Family of Standards for Modeling and Simulation (M&S) High Level Architecture (HLA) – (a) IEEE 1516–2000 Framework and Rules; (b) IEEE 1516.1–2000 Federate Interface Specification; (c) IEEE 1516.2–2000 Object Model Template (OMT) Specification

Klein G (1997) The recognition-primed decision (RPD) model: Looking back, looking forward. In Zsambok EC and Klein G (eds) Naturalistic decision making, Mahwah, NJ, Lawrence Erlbaum Associates Publishers, pp 285–292

Knight DE, Curtis WH and Fogel LJ (eds) (1991) Cybernetics, simulation, and conflict resolution, Proc. of the 3rd Annual Symposium of the American Society for Cybernetics (Spartan Books, New York)

Leimar O (1997) Repeated games: A state space approach. Journal of Theoretical Biology, 184, pp 471–498

Murch R and Johnson T (1998) Intelligent Software Agents. Englewood Cliffs, NJ: Prentice Hall

von Neumeann J and Morgenstern O (1953) Theory of games and economic behavior. (3rd ed) Princeton, NJ, Princeton University Press

Niland MW (2006) The migration of a collaborative UAV testbed into the FLAMES simulation environment. In Proceedings of the 2006 Winter Simulation Conference, pp 1266–1272

Ören TI (1987) Model update: A model specification formalism with a generalized view of discontinuity. In Proceedings of the summer computer simulation conference, Montreal, Quebec, Canada, 1987 July 27–30, 689–694

Ören TI (1991) Dynamic templates and semantic rules for simulation advisors and certifiers. In Fishwick AP and Modjeski BR (eds) Knowledge-based simulation: Methodology and application, Berlin, Heidelberg, New York: Springer, pp 53–76

Ören TI (2001) Towards a modelling formalism for conflict management. In Sarjoughian SH and Cellier EF (eds) Discrete-event modeling and simulation: A tapestry of systems and AI-based theories and methodologies, Berlin, Heidelberg, New York: Springer, pp 93–106

Parsons S and Wooldridge M (2002) An introduction to game theory and decision theory. In Parson et al. (eds) Game theory and decision theory in agent-based systems, Kluwer Academic Publishers, pp 1–28

Power D (2002) Decision support systems: Concepts and resources for managers; Chapter 10: Building Model-Driven Decision Support Systems, Quorum Books division Greenwood Publishing, pp 172ff

Rosen R (1985) Anticipatory systems – Philosophical, mathematical and methodological foundations, New York, NY: Pergamon

Sallach D (2003) Interpretive agents. In Proceedings of the Agent 2003: Challenges in Social Simulation Conference, University of Chicago

Shannon, CE and Weaver, W (1949) The mathematical theory of communication. Urbana: University of Illinois Press

Shubik M (ed) (1964) Game theory and related approaches to social behavior. New York, NY, Wiley

Simon H (1982) Models of bounded rationality, vol 1, Cambridge, MA, MIT

Sokolowski JA (2003) Enhance Decision Modeling Using Multiagent System Simulation, SIMULATION, vol 79 no 4, pp. 232–242

Tolk A (2004) An agent-based decision support system architecture for the military domain. In: Phillips-Wren GE and Jain LC (eds) Intelligent decision support systems in agent-mediated environments, Volume 115 Frontiers in Artificial Intelligence and Applications, IOS Press, pp 187–205

Tolk A and Kunde D (2003) Decision support systems in the military environment., In Tonfoni G. and Jain L. (Eds) Innovations in decision support systems, International Series on Advanced Intelligence, Advanced Knowledge International, Magill, Adelaide, Australia, pp 175–210

Tolk A and Gaskins RC (2006) Challenges and Potential of Service-oriented Architectures for Net-Centric Operations. NATO Report MP-HFM-136, Research and Technology Organization: Human Factors and Medicine Panel, Brussels

Tolk A, Diallo SY and Turnitsa CD (2007) Model-Based Data Engineering: Preparing a Paradigm Shift towards Self-Organizing Information Exchange. Proceedings of the Summer Computer Simulation Conference (SCSC-07), pp 1112–1119

Tolk A, Turnitsa CD and Diallo SY, "Implied Ontological Representation within the Levels of Conceptual Interoperability Model," International Journal for Intelligent Decision Technologies 2 (1), (in press)

Tversky A and Kahneman D (1974) Judgment under uncertainty: Heuristics and biases. Science 185, pp 1124–1131

Wikipedia: (Phi-Per) (2004) Philosophy of Perception. Available online via <http://en.wikipedia.org/wiki/Philosophy_of_perception>

Wooldridge M (2002) An introduction to multi-agent systems, New York, NY, Wiley

Yilmaz L and Ören T (2004) Dynamic model updating in simulation with multimodels: A taxonomy and generic agent-based architecture. In Proceedings of the Summer Computer Simulation Conference (SCSC-04), pp 3–8

Yilmaz L, Ören TI and Ghasem-Aghaee N (2006) Simulation-based problem solving environments for conflict studies: Toward exploratory multisimulation with dynamic simulation updating. Simulation, and Gaming Journal 37(4), 534–556

Zachary W (1998) Decision-support systems: Designing to extend the cognitive limits. In Helander GM (ed) Handbook of Human-computer interaction, Amsterdam: Elsevier Science Publishers

Zeigler BP (1986) Toward a simulation methodology for variable structure modeling, In Elzas MS, Ören TI and Zeigler BP (eds) Modeling and simulation methodology in the artificial intelligence era, Amsterdam: North Holland

Zeigler PB, Praehofer HP and Kim GT (2000). Theory of modeling and simulation: Integrating discrete event and continuous complex dynamic systems, New York, NY: Academic

Zsambok EC (1997) Naturalistic decision making: Where are we now? In Zsambok EC and Klein G (eds) Naturalistic decision making, Mahwah NJ, Lawrence Erlbaum Associates Publishers, pp 3–16