Group Decision and Negotiation, 6:77–108 (1996)

An Intelligent Distributed System for Strategic Decision Making

SUZANNE PINSON
LAMSADE, Universite Paris-Dauphine, 75775 Paris Cedex 16, France

PAVLOS MORAÏTIS
LAMSADE, Universite Paris-Dauphine, 75775 Paris Cedex 16, France

Abstract

The decision-making process in strategic planning is often too complex to be handled by conventional methods. Strategic planning problems (building new plans, new product planning, etc.) belong to the class of problems called ill-structured by H. Simon. They involve a decomposition of the main problem into a set of subproblems, a reasoning process at the subproblem level, and then a coordinated and aggregated process to build a global solution. Because partial solutions are generated without having a complete view of the global objective, this type of decision-making process very often generates incoherent and contradictory hypotheses and actions. Therefore, the main problem is to find a way to achieve coherence and coordination among decisions made locally by different agents, at different levels.

Recent advances in Artificial Intelligence, particularly in the field of multi-agent theory, offer great promises in modeling strategic planning processes. In this article we present a general framework called "A Coherent Plan of Coordinated Actions (CPCA)" for building intelligent distributed strategic decision making systems which integrates advances in both distributed decision making and Distributed Artificial Intelligence. We then describe a multi-blackboard system, ARISTOTE, which is aimed at helping corporate managers address the feasibility and coherence of a plan of actions.

Key words: Distributed Artificial Intelligence, multi-agent theory, strategic planning, distributed decision making, blackboard system

1. Introduction

The decision-making process in strategic planning is often too complex to be handled by conventional methods. Strategic planning problems (building new plans, new product planning, quality assurance planning, etc.) belong to the class of problems called ill-structured by Simon (1969). They involve a decomposition of the main problem into a set of subproblems, a reasoning process at the subproblem level, and then a coordinated and aggregated process to build a global solution. Because partial solutions are generated without having a complete view of the global objective, this type of decision process very

Received 1994, Revised 1995.

often generates incoherent and contradictory hypotheses and actions. Therefore, the main problem is to find a way to achieve coherence and coordination among decisions made locally by different agents, at different levels.

A variety of knowledge-based systems have been developed to model strategic planning decisions (Berman and Kautz 1990; Mockler 1989; Greenley 1989). A good review of the literature can be found in Clark (1992). They are mostly decision-support systems integrating a database, a spreadsheet, financial analysis modeling, forecasting, and reporting systems. One of them (Paradice 1992) has a more sophisticated knowledge representation (i.e., object-oriented representation), works with qualitative information and causal models, and is able to support organizationally intelligent behavior. Other systems are expert systems for establishing organizational structures (Lehner 1992), but most of these projects were discontinued due to the high complexity of the tasks and the low involvement of representatives of classical organizational theory. Some other systems are Decision Support Systems (Badiru et al. 1993) or Group Decision Support Systems (Thietart 1988; Chung et al. 1989) which aim at simulating different strategies, either by incorporating probability information specified by the user in decision scenarios, or by requiring decision makers to be physically present and to communicate through a computer network. These kinds of systems do not encompass automatic task decomposition, task allocation, and reasoning processes. Furthermore, they have never addressed the problem of coordination and compatibility of actions proposed locally by different agents.

Recent advances in Artificial Intelligence, particularly in the field of multi-agent theory, offer great promise in modeling strategic planning processes (Holloway 1983). In this article we present a methodology for building distributed strategic planning systems, and we describe a multi-agent planning support system, ARISTOTE, which has been developed to support top-level managers in creating strategic scenarios and in assessing the feasibility and coherence of a plan of actions. ARISTOTE is an interactive planning system, meaning that both user and system contribute to the development of plans. In contrast to previous approaches, our system allows:

1. a computer representation of a set of agents which cooperate in the problem-solving process, an explicit representation of communication and control between agents, a knowledge-driven allocation of the subproblems to a set of agents, and a knowledge-driven creation of elementary actions;
2. detection of contradictory actions, and, consequently, the possibility for the user to modify the chosen global strategy;
3. the possibility of solving ill-defined problems by using the trial-and-error method and by testing several alternatives in order to find a satisfactory one;
4. the possibility of using incomplete, qualitative knowledge, taking into account each agent's subjective rationality, specific to strategic decision making problems.

This article is organized as follows. Section 2 gives a brief idea of the domain of strategic decision making. Sections 3 and 4 show that Artificial Intelligence methods, and especially the multi-agent paradigm, are able to bring a solution to the above problems. It also shows that the concepts of organization theory can be fruitfully used in the analysis of a strategic planning process. In section 5, we present a framework called "A Coherent Plan of Coordinated Actions" (CPCA) for a Distributed Strategic Decision-Making System (DSDMS), which integrates advances in both distributed decision making and Distributed Artificial Intelligence. The general architecture of the system is described in section 6. Section 7 presents an example of the operating and implementation of ARISTOTE.

2. Strategic decision-making process

2.1. Structure and characteristics

It is commonly assumed (Antony 1965) that decision-making processes are organized into three levels: strategic decisions, managerial decisions (optimal resource allocation, control), and operational decisions (task execution). Strategic planning specifies the objectives and resources needed to realize them. Early work in the field of strategic planning (Antony 1965; Mintzberg et al. 1976; Simon 1969; Cyert and March 1963; Armstrong 1982) has led to an exhaustive analysis of its characteristics:

Irregularity: Each problem is different; there is no general approach to the analysis of all types of strategic problems. The elements of strategic planning are not repetitive; they cannot be programmed into a "standing plan" procedure. *Feedback* is essential in strategic planning; each of the elements of strategic planning is subject to repeated evaluation and change.

Complexity: Simon (1969) considers a complex system as a system composed of a large number of elements which interact in a complex way. Simon considers that complexity often takes the form of a tree structure wherein a system is decomposed into interlinked subsystems. The solution to the main problem is replaced by the set of solutions to the subproblems. For Newell and Simon (1961), human problem solving of complex problems involves several *trial and error* steps. The more difficult and newer the problem is, the greater the number of trial and error steps.

Ill-structured problems: A strategy problem may be defined as a situation where it is not obvious that there is a solution. The "solution" to a business policy problem is often obtained in stages, which means that there is no algorithmic solution (Nutt 1993); it should not be a question of searching for the optimal solution but rather of being able to formulate the alternatives among which there may exist a *satisfactory solution*.

Uncertain and imprecise knowledge: Since strategic planning tries to realize long-range objectives, hypotheses have a high degree of uncertainty.

2.2. The actors

In the strategic decision-making process, four groups of individuals are usually involved (Thietart 1988):

1. One group is made up of top-level managers, such as division heads or members of the executive committee. They define the strategic orientations of the firm (i.e., international investment, niche concentration, etc.), decompose the main objective into subobjectives, and ask the lower levels to solve them. At the end of the strategic process, they decide whether the recommendations are acceptable, need to be modified, or should be rejected.
2. A second group consists of middle-level managers: finance, production, marketing, human resources, etc. Their objective is to deal with subobjectives and to assign them to specialists (or operational managers) while taking into account the internal constraints (financial, technical, managerial, etc.) of the firm.
3. Another group is composed of low-level managers (line managers, operational managers) in charge of an operational unit such as a product department or a Strategic Business Unit. Their main objective is to make recommendations and to propose feasible elementary actions based on the available information regarding the subobjectives and environmental constraints.
4. The fourth group is made up of staff specialists (lawyers, market researchers, R&D managers, etc.) who define all the environmental constraints, such as government regulations, historical and current industry information, historical and current information on competitors and customers, political data, demographic data, etc. These constraints are important information because they can prevent or facilitate the successful implementation of recommendations.

Given the above characteristics and the players involved, the design of decision support systems for strategic planning problems should be focused on the creation of an envelope which allows modeling of the players' behavior and simulation of their interaction. The framework of distributed decision making will help to define such a system.

3. Distributed decision making and strategic planning

In the existing literature, distributed decision making has been studied from different points of view:

- the cognitive engineering point of view;
- the system theory point of view; and
- the organization and management point of view.

The cognitive engineering approach to modeling distributed decision making aims at a model of human cognitive functioning in a complex cooperative work setting as a basis for the design of information and decision support systems (Rasmussen 1988). This cognitive approach tells us "that for the design of DSS, the structure of complex socio-technical systems should be modeled from at least four different points of view: (1) the content and structure of the basic work domain, (2) the structure of the decision-making task, (3) the level of cognitive control of decision agents, [and] (4) the allocation of control function to decision-making agents and the resulting *social organization*" (Rasmussen 1991). We will use this approach to define the level of decision functions of our framework (cf. section 5.3.2).

The system theory approach (Mesarovic et al. 1970) considers the structure of distributed decision making in action in control as a hierarchical, distributed and self-organizing control system. This multi-objective, multi-level approach will be developed in more detail in section 4.

Support for the modeling effort can also be found in the results of organization and management research (Thompson 1967). This research can supply adaptation heuristics guiding the formation of organizational structures, models of communication between decision makers, and allocation of roles to individuals reflecting the organizational structure.

According to Rasmussen (Rasmussen 1991), different structures of the social organization are possible for the coordination of activities:

(1) Autocratic coordination: One decision maker is responsible for the coordination of the activities of all the other agents.

(2) Hierarchical coordination: Coordination is distributed in the organization which is stratified such that one level of decision makers evaluates and plans the activities at the next lower level.

(3) Heterarchical planning: In bureaucratic organizations, the stratification is less pronounced, and decision makers invade the domain of subordinates and superiors for advice and monitoring.

(4) Democratic planning: Coordination involves interaction and negotiation among all the decision makers of the organization.

(5) Diplomatic planning: The individual decision makers negotiate only with the neighbors involved and the information traffic is locally planned (also called adhocracy or networked organization in organization theory).

These different architectures have also been proposed by Fox (1981) and Malone (1988) under different names. They imply different forms of communication between agents.

4. Strategic planning and multi-agent systems

As stated previously, strategic planning problems belong to those class of problems for which a solution requires cooperation between several agents. In this section we will show that distributed decision making and, more precisely strategic decision making and multi-agent systems deal with similar concepts, and that a rapprochement of these two domains may be of mutual interest. Multi-agent systems theory may be enriched by human organization models and strategic planning problems, while problems which are essentially ill-structured may be modeled, satisfactorily, by multi-agent systems.

4.1. Problem decomposition

In order to deal with complexity, multi-agent theory proposes the kind of decomposition found in distributed decision making. A single "supertask" is decomposed into smaller subtasks, each of which requires less knowledge. Subtasks are allocated among a group of intelligent "agents." Relations among agents and role distribution define different types of agent organization. Among the most well known are (a) contract nets (Davis and Smith 1983), (b) the scientific community metaphor (Kornfeld and Hewitt 1981), and (c) hierarchical organization (Steeb 1988).

Since the strategic planning process is hierarchical in nature, Mesarovic's multi-level, multi-objective model and its multi-agent hierarchical counterpart could be fruitfully adapted to model this process. On each level a decision unit (controller) is concerned with a larger portion of the system, and its primary task is to coordinate the actions of the subordinate units. This decomposition corresponds also to Simon's hierarchy concept and to the "partial rationality" concept of Cyert and March (1963).

Another interesting point in Mesarovic's model is the existence of three types of levels: (a) the abstraction level, which facilitates modeling, (b) the decision's complexity level, which refers to the vertical decomposition of decision problems into subproblems of decreasing complexity, and (c) the organizational level, which refers to relations between the different decision entities hierarchically arranged in such a way that a higher level authority exists over the lower level.

In multi-agent theory, similar concepts of "level" are proposed. Lesser and Erman (1980) argued for three dimensions of problem decomposition. Problems could be decomposed along lines of location (e.g., spatial, temporal, logical), by information level or degree of abstraction, and by the "interest areas," which include the given partitioning of skills among knowledge sources.

4.2. Coordination and coherence

In Mesarovic's system theory approaches, in organization theory, and in multi-agent theory, two important factors are considered: coordination and coherence. According to Mesarovic and Simon, hierarchical multilevel systems have inevitably to tackle the problems of coordination and coherence.

Coordination refers to the fact that upper-level units influence lower-level decisions in such a way that the actions proposed to solve the problems at this level also constitute a satisfactory solution for the global objective.

In organization theory, March and Simon (1958) pointed out that coordination is necessary among subsystems when a high degree of specialization exists. The type of coordination used in such an organization depends on the situation. If the situation is stable and foreseeable, coordination is a simple deterministic planification called coordination by plans. The more varying and unpredictable the situation is, as in the case of strategic planning, the more a coordination by feedback (retroaction) is necessary. This coordination needs communication in order either to detect divergences between expected and proposed solutions, or to require modifications of actions to reduce the deviations.

In multi-agent theory, coordination is characterized by interaction among agents performing collective activities (Bond and Gasser 1988). Effective coordination implies some degree of mutual predictability and conflict resolution. Coordination does not necessarily imply cooperation, since antagonists can be coordinated.

Coherence designates the requirement that subsystem objectives must be compatible (Mesarovic et al. 1970).

In organization theory, Mesarovic and Simon observed that lower-level decision units must be autonomous to ensure the efficiency of hierarchical organizations. Hence, the problem is how to guarantee global coherence in a system where decisions are made by different persons at different hierarchical levels. These persons have a partial view of the main problem, which is limited and incomplete knowledge in their action domain. Their goal is to solve a local subproblem.

In multi-agent theory, the problem of coherence is addressed in a similar way. Although the agents are predisposed to work together towards network goals, they may compete or conflict with each other because each of them interprets the network goals locally (Durfee et al. 1987). Coordination is a way to solve these problems. Different types of coordination are proposed: distributed planning (Durfee et al. 1987), and relation-driven coordination scores (Von Martial 1992).

Lesser and Corkill (1987) have suggested that global coherence with decentralized control requires the achievement of three conditions: *coverage*: each necessary portion of the overall problem must be included in the activities of at least one agent; *connectivity*: agents must interact in a way permitting the covering activities to be developed and integrated into an overall solution; and *capability*: coverage and connectivity must be achievable within the communication and computation resource limitations of the network.

5. A coherent plan of coordinated actions approach (CPCA)

5.1. A cooperative approach

Due to the irregular and ill-structured nature of strategic planning problems the user (the strategic planner) has to cooperate with the system to solve the problem. Our approach is based on the paradigm of supporting the planning process rather than automating it. The model has to help the user assess the feasibility of a given solution by simulating different strategic planning scenarios (cf. Figure 1). It starts at the top of the hierarchy by asking the user to identify his/her global goal, then decomposes the global objective into a scenario composed of subobjectives and works down to more immediately achievable objectives.

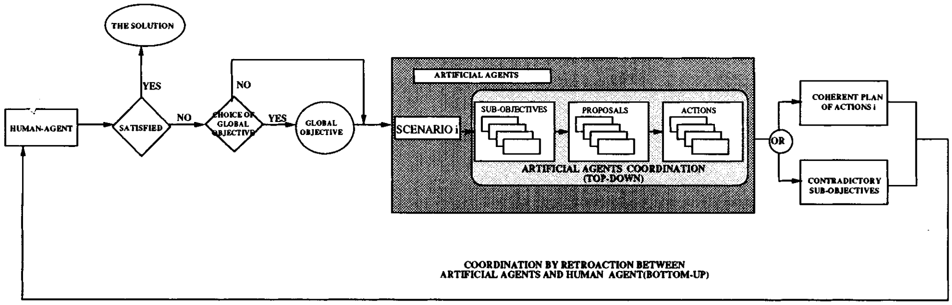

Figure 1. Cooperative environment for decision making.

A feedback has been designed to allow the user to ask for another scenario if the solution is not feasible or satisfactory, until a satisfactory solution is obtained. This, also called restructuring by Shakun (1991), may be supported using domain-independent methodological knowledge. The planning process is terminated by the user, who decides that the synthesized plan describes and solves the problem. This approach is called by March and Simon (1958) the principle of a "satisfying solution." This may imply that in general a search for a solution in an organization will return only an adequate, rather than the best, solution. In its present state the system uses predefined scenarios to decompose the main goal into subobjectives.

Our approach differs from traditional AI planners which are goal driven and fully automated, i.e., the user has no possibility to interact with the planner. A fully automated approach makes sense in domains with an underlying closed-world assumption, where the knowledge representing objects and actions is not very complex, e.g., the blocks-world domain. This assumption is not true in strategic planning domains. The planning process is not an automatic goal-driven process with predefined operators and search in a search space, but rather an interactive and iterative construction process.

5.2. An example

To illustrate the basic concepts of our approach and the general functionality of each element, we present the following example, which will be used throughout this article. Let us consider a company which manufactures three products A, B, and C, where the global objective G is to achieve business success. In order to achieve this objective, four simultaneous and coordinated subobjectives are defined by top managers: increased market share, increased profit, increased product quality, and better qualified people. These subobjectives are passed on to middle-level managers and become their goals (local goals). Several elementary actions, defined by specialists (lower-level managers), are proposed to achieve these subobjectives, such as "decrease price of product A," "extend brand to product B," "give discount on product C," "increase advertising expenses," "increase quality control," "replace obsolete machines," "train in new technology machines," etc. The relations which exist between these three levels of decision are by nature hierarchical.

Taking into account environmental constraints as stated in section 2.2, the implementation of these elementary actions results in keeping only feasible actions and in quantifying them. For example, a 5% discount is proposed on product C with regard to the competitors' prices.

Conflict may arise between elementary actions proposed to achieve two different subobjectives: an increase in the price of product A is necessary in order to achieve a "increased profit," whereas a decrease in the price of the same product is required to achieve a "increased market share." The first one is very important for the achievement of the subobjective "increased profit," while the second one is moderately important for the achievement of the subobjective "increased market share." At the level of the specialists, the conflict cannot be solved because actions may be proposed for each subobjective by different persons at different intervals of time. It will be the responsibility of the strategic manager to choose a set of coherent actions among all proposed actions in order to achieve the global objective. It is also important to keep all the actions proposed to achieve a single subobjective (they represent the local optimum as defined by Simon), in order for the system to give explanations to the user.

There is obviously a good deal more to strategic company planning than just specifying a plan of elementary actions. Such an approach is, however, a useful starting point to help a planner develop a focus and direction for the more detailed studies that follow. This general approach also provides a useful starting point, and conceptual foundation, for developing a prototype multi-agent system in the corporate strategy planning area.

5.3. Decomposition levels

In this section we present the conceptual framework. According to Mesarovic's model presented in section 4.1 (Mesarovic et al. 1970), we propose three types of decomposition levels: (a) abstraction levels, (b) decision function levels, and (c) organizational levels.

5.3.1. Abstraction levels. Abstraction levels deal with problem decomposition into different description levels. On the lower level, the description is more detailed than on the higher level. We propose three levels of abstraction that we formalize as follows:

Global objective level: Let G be a global objective. It can be decomposed into a set of subobjectives: $G = \wedge O_j$, $j = 1, \ldots, m$. A strategy (or a scenario) is a conjunction of subobjectives. In other words, the solution of the overall problem is the conjunction of the solutions of all the subproblems.

A coefficient called COP (*coefficient of objective priority*) is attached to each subobjective. It represents the relative importance of the subobjective in order to achieve a higher level objective.

This concept of priority is similar to that proposed by Saaty in the Multiattribute Hierarchical Method (MHM) (Saaty 1980). It shows how each subobjective is able to contribute to the achievement of the higher level objective. For example, the subobjective "increased profit" may be more important to achieve the global objective "business success" than the subobjective "increased market share."

Sub-objectives level: A sub-objective O_j is defined as the conjunction of a set of plans P_k^j, i.e., $O_j = \wedge P_k^j$, where P_k^j is a plan k necessary for the achievement of the subobjective O_j. A plan is the conjunction of a set of elementary actions a_i^j, $P_k^j = \wedge a_i^j$, $i = 1, \ldots, n$.

Let us take the example of section 5.2: the price and quality of product A have to be modified in order to achieve the subobjective "increased market-share." The conjunction of these two elementary actions, that is, "decrease price on product A" and "increase quality of product A" is called plan P_k^j on product A.

Elementary actions level: An action is defined as the process of giving a value to one attribute of the object under study. To take the example of section 5.2, sub-objective *Oj* would be "increased profit," a real world entity ω would be "product A," one of its attributes y_1 would be "price." At the elementary actions level, the value "$v_1^j = +5\%$" would be chosen with a *coefficient of action priority (CAP)* 0.9, which means that it would be very important to decide a ″5% increase in the price of product A (action a_1^j) to achieve the subobjective "increased profit".

More formally, let a_i^j be an action proposed in order to achieve the subobjective *Oj*, $i = 1 \ldots n$; Ω be the set of real world entities (i.e., product A, product B); $\omega \in \Omega$ be a real world entity; y_i^j be the attribute of ω modified by the action a_i^j; V a set of values of attribute y_i; $v_i \in V$ (the subscripts refer to the action proposed to achieve a sub-objective, the superscripts refer to the corresponding sub-objective).

We define a_i^j as a n-tuple

$$a_i^j = (\omega, y_i^j, v_i^j, \mathrm{CAP}_i^j, \mathrm{CGP}_i^g, \mathrm{IMP}_i^j)$$
$$\text{with } i = 1, \ldots, n; j = 1, \ldots, m; g = 1, \ldots, r$$

where

CAP_i^j is the coefficient of action priority attached to action a_i^j. This coefficient represents the relative importance of elementary actions proposed by specialists in order to achieve different subobjectives O_j. For example, an action a_1 "5% increase in the price of product A" may be proposed to satisfy the subobjective "increased profit" and at the sime time an action a_2 "2% decrease in the price of product A" may be proposed to satisfy the subobjective "increased market share". Since we consider that agents are able to evaluate the consequences and the importance of their proposed actions (see section 5.5), we are able to say that the first action is more important than the second in the achievement of the corresponding subobjectives.

CGP_i^g is the coefficient of global priority attached to action a_i^j. The coefficient of global priority is the product of the coefficient of action priority CAP_i^j and the coefficient of objective priority COP_j of the corresponding subobjective. It represents the relative importance of the elementary action a_i^j in the achievement of the global objective. It will also allow the system to detect and solve the conflicts between subobjectives (see section 5.5).

IMP_i^j is the impact of action a_i^j on the sub-objective O_j. The proposed actions may be positive or negative: a positive action is a necessary action to achieve a subobjective, a negative action is an action which has a negative impact on the achievement of a subobjective. For example, a 5% increase in the price of product A has a positive impact on the subobjective "increased profit" and a negative one on the subobjective "increased market share." The variable IMP_i^j takes two values "+" and "−".

5.3.2. Decision function levels. Decision function levels refer to the vertical decomposition of the decision-making process into different complexity levels. Decision complexity increases from the lowest to the highest level. Following the cognitive engineering approach (Rasmussen 1988) and the cognitive model of planning proposed by Hayes–Roth and Hayes–Roth (1979), we propose three levels of decision functions.

Strategic level: At this level agents assume decomposition of a global objective into a set of subobjectives and coordinate the allocation of subobjectives to decision centers; each decision center takes control of a subobjective. The subobjective allocation criterion is the functional decoupling criterion presented in the cognitive approach (Rasmussen 1988). This criterion serves to minimize the necessary exchange of information among agents. The basic principle is to identify subobjectives that can be separated and achieved by a decision center with a minimum of interaction and communication. The agents are allocated roles such as analysis, goal setting, planning, for which they are specialized.

Decision-center level: At this level agents prepare proposals for each subobjective given at the strategic level. A proposal refines the subobjective into a more precise goal. It quantifies the goal level and specifies the domain of action (for example, the subobjective "increased market share" is quantified into a "2% market share increase" and refined into several proposals as "take actions on price and on quality.") These proposals will be allocated to specialists and transformed into elementary actions. The proposal allocation criterion is the specialist's competency as presented in the cognitive approach (Rasmussen 1988). The activities of specialists are related to different parts of the work domain.

Decisions at the strategic and decision-center levels can be considered as "decision control." Rasmussen (1991) called them *levels of cognitive control of decision agents* and Mintzberg et al. (1976) called their use *meta-decision making*.

Specialist level: At this level, agents propose one or more elementary actions for each proposal. Each specialist has his own domain of competence. Each specialist assigns a coefficient of action priority to each proposed elementary action, as a function of his/her knowledge, intuition, experience, and the given context, thus following a subjective rationality adapted to strategic decision-making problems.

5.3.3. Organizational levels. Vertical decomposition, that is, the decomposition of decision functions, is accompanied by a horizontal decomposition forming an organizational hierarchy. At strategic and decision-center levels a decision unit (controller) is concerned with a portion of the main problem and coordinates the actions of the subordinate units.

5.4. Feasibility and compatibility conditions

Global coherence of the system, i.e., coherence of behavior of autonomous agents, is conditioned by the satisfaction of feasibility and compatibility conditions.

Feasibility: Each subproblem has to be feasible in its own domain. The feasibility concept allows the reasoning process to propose actions satisfying the economic and environmental constraints.

A constraint is written as:

$c_k = (\omega, \mu_k, b_k)$, $k = 1, \ldots, l$ with μ_k an attribute of the entity ω and b_k a constraint that must be satisfied by any value of μ_k.

The actions proposed by the specialist agents are matched against the contraint base to yield *feasible elementary actions*. More precisely, the matching process uses the knowledge of a specialist which is of the form:

$$\wedge\langle\text{pre-condition}\rangle \rightarrow \langle\text{elementary action}\rangle$$

The pre-condition part is of the following form:

$\langle\text{pre-condition}\rangle = (\omega, l_t, V_t)$, $t = 1, \ldots, n$ with l_t an attribute of entity ω and $V_t = \{v_1, \ldots, v_i\}$, $i = 1, \ldots, s$, a set of possible values of the attribute l_t. (An example of a pre-condition is (product-A, production-cost, {high, medium})).

The action part is the elementary action as defined in section 5.3.1.

Compatibility: The actions generated for a subproblem solution have to be compatible with the actions generated for other subproblems at the same level of decision making. Conflicts among incompatible or contradictory actions that exist very often in distributed decision making will be solved by applying Simon's compromise concept (Simon 1975). This concept means that in a situation where a system aims at the simultaneous achievement of a set of subobjectives, the chosen solution never allows the total or perfect achievement of the subobjectives. It only consists of the best possible solution given the context in hand. In our case this means that elementary actions chosen for the achievement of the main objective are not always the best actions for the achievement of the corresponding subobjective, but they are compatible. The compromise concept is represented by our compatibility criterion defined in the next section.

5.5. Conflict resolution mechanism

The conflict resolution mechanism is based on March and Simon's conflict theory and on the detection of individual conflicts applied to internal conflicts in an organization (March and Simon 1958). This method is based on the ability of an individual to evaluate the consequences of the choice of an action. The individual conflict arises during the decision-making phase. The choice of the best action or detection of conflicts between actions is done using the relative importance of the proposed actions.

In our CPCA method, when several actions on the same entity are proposed in order to solve several different objectives, the conflict resolution mechanism chooses the action with the highest coefficient of global priority, that is to say, the one which has the greatest perceived likelihood of enabling the global objective to be achieved.

Let $a_i^j = (\omega, y_i^j, v_i^j, \mathrm{CAP}_i^j, \mathrm{CGP}_i^g, \mathrm{IMP}_i^j)$ and $a_p^k = (\omega, y_p^k, v_p^k, \mathrm{CAP}_p^k, \mathrm{CGP}_p^g, \mathrm{IMP}_p^k)$ be two elementary actions proposed for two different subobjectives O_j and O_k, on the same attribute y of the entity ω; $\mathrm{CAP}_i j$, CAP_p^k, their coefficients of action-priority, and $\mathrm{CGP}_i g$, CGP_p^g, their coefficients of global priority.

If $\mathrm{CGP}_i^g \neq \mathrm{CGP}_p^g$, then the action with the greatest coefficient of global priority is chosen:

$$\langle a_i^j \; v \; a_p^k \rangle = \mathrm{Max}\,[\mathrm{CGP}^g(a_i^j), \mathrm{CGP}^g(a_p^k)]$$

With this mechanism, only a subset of the set of actions proposed by different specialists to achieve a particular subobjective will be chosen and will be present in the global plan of actions. The set of actions proposed by the specialists to solve a subobjective represents an optimal solution. Since only a limited number of actions are chosen by the conflict mechanism, at the end of the process the solution to the subobjective will no longer be optimal but satisfactory (this again illustrates the implementation of the compromise concept of Simon; cf. Section 5.4).

If $\mathrm{CGP}_i^g = \mathrm{CGP}_p^g$ and $v_i^j \neq v_p^k$ (same importance but different values), this means that the conflict cannot be solved, and thus that the subobjectives j and k are incompatible. As we will see later, this scenario is considered incoherent.

To illustrate the conflict resolution mechanism, let us take the following network of decisionnal units shown in Figure 2.

This network is composed of a strategic unit (STRU), of two decision-center units (DCU) and of three specialist units (SPU). One (SPU_1) of them belongs to two teams, the team of DCU_1 and the team of DCU_2. Let SPU_1 be a price specialist; it may propose two elementary actions ("4% decrease in the price of product A," 0.4) and ("2% increase in the price of product A," 0.6) for two different subobjectives: DCU_1 = "increased market-

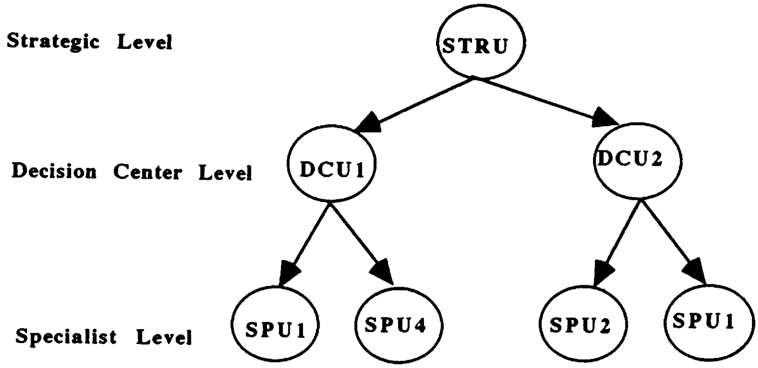

Figure 2. A network of decisional units.

share" and DCU_2 = "increased profit" with, respectively, the coefficients of action-priority 0.4 and 0.6. The specialist SPU_1 uses its causal map to say that the effects of the first action are more important than the effects of the second action in the achievement of the subobjectives. The coefficients of objective priority (COP) of the subobjectives are given by STRU when it creates the scenario. The global solution is represented in Figure 3:

If the COPs given to the subobjectives O_1 and O_2 by STRU are respectively 0.7 and 0.3, the coefficient of global priority of each action can be computing using our definition: CGP = COP*CAP. It gives CGP = (0.6*0.7) = 0.42 for action all, CGP = 0.21 for action a21, CGP = 0.12 for action a12, and CGP = 0.06 for action a22. According to this senario, actions a11 and a12 are contradictory. Using our compatibility criteria, action all will appear in the final solution.

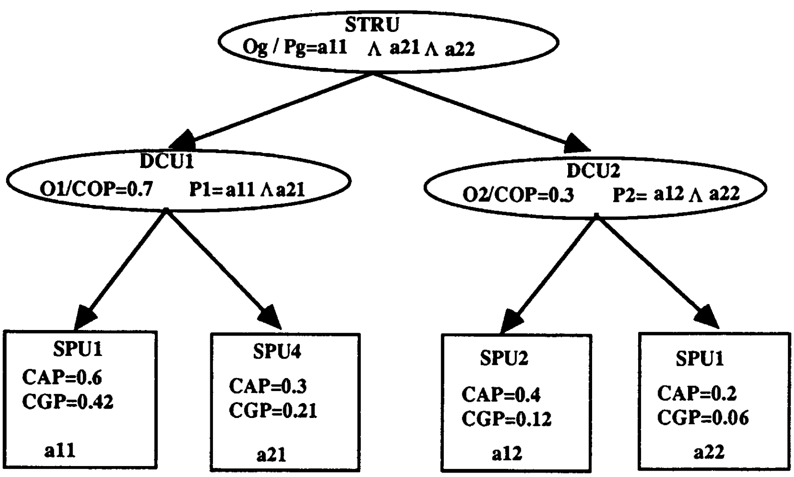

Figure 3. An example of conflict resolution in a distributed decision-making network.

6. ARISTOTE architecture

As we have shown in the previous sections, the general concepts of DAI and organization theory helped us to define an intelligent distributed strategic decision-making system. A prototype system has been developed called ARISTOTE. It has been implemented on a SUN SPARC II workstation using SMECI, an object-oriented, multi-task platform developed by ILOG. A window and icon interface, called AIDA, is used which allows users to simply "point and click" to verify strategies, the distribution of decision making between agents, actions already proposed by agents, etc. (for more detail, see Moraitis 1994).

The ARISTOTE system operates using a global objective, a set of subobjectives called scenarios and a set of actions proposed by the system to achieve these subobjectives under economic constraints. What the system does is (cf. Figures 4, 5, and 6):

- recommend a plan of actions to be performed to achieve the global objective;
- if the actions are found to be incompatible, it gives explanations. It looks for another scenario which is proposed to the user. Either the user accepts it and the system starts a new cycle or the user gives another global goal. He may also stop the process.

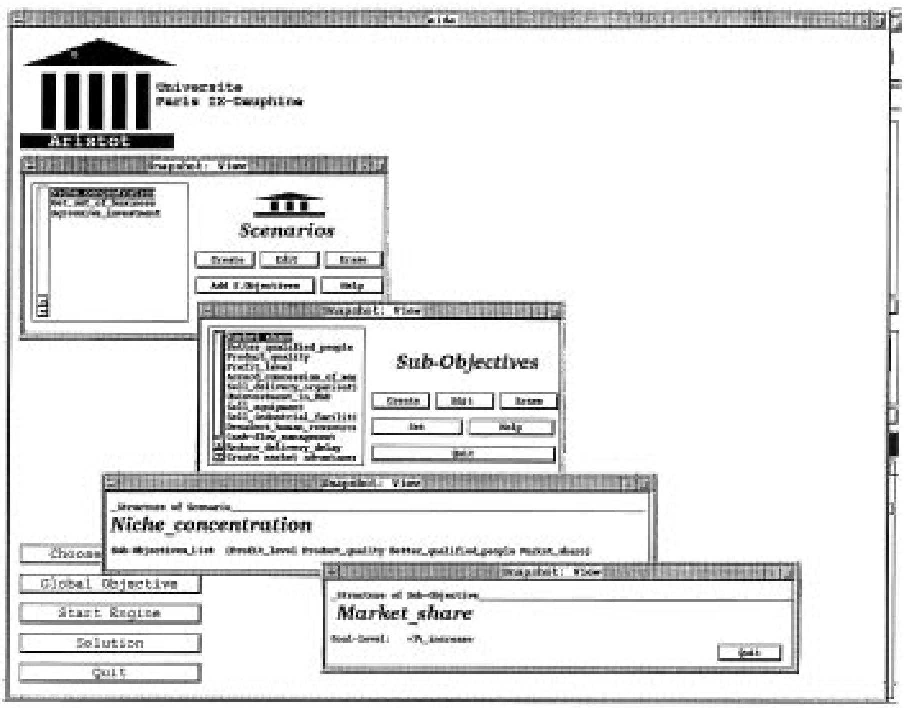

Figure 4. Input windows showing relations between Scenarios and Sub-Objectives.

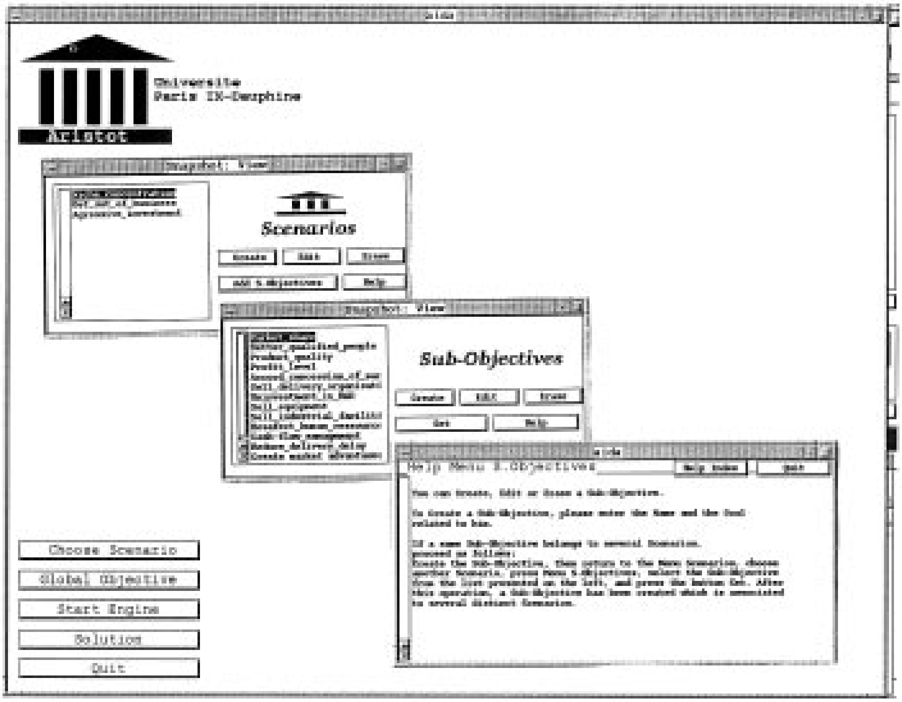

Figure 5. Help function of ARISTOTE.

Figures 4 to 6, which show screens from the system operation, illustrate man–machine interaction. Figure 6 illustrates how the system gives its recommendations and its explanations if asked by the user.

ARISTOTE architecture builds on previous blackboard architecture systems such as BB-1(Hayes–Roth 1988), ATOME (Laâsri 1988), distributed blackboard systems such as DVMT (Lesser and Corkill 1983) and multi-expert systems such as CREDEX (Pinson 1992). However these systems do not provide a powerful mechanism to handle the feasibility and compatibility criteria. The specific features of ARISTOTE are (1) direct communication between agents through message passing and indirect communication through shared memory (blackboard), (2) representation, combination and aggregation of hypotheses and actions using the compatibility criteria leading to a global coherent solution and (3) agent activation—agents may be activated in parallel without impeding convergence towards a possible solution. Potential conflicts between actions are solved by the conflict resolution mechanism (cf. section 5.5); this characteristic means there are no conflicts between KSs, only between actions.

The ARISTOTE multi-agent system consists of three types of modules: the agents, blackboards, and constraint base (cf. figure 7).

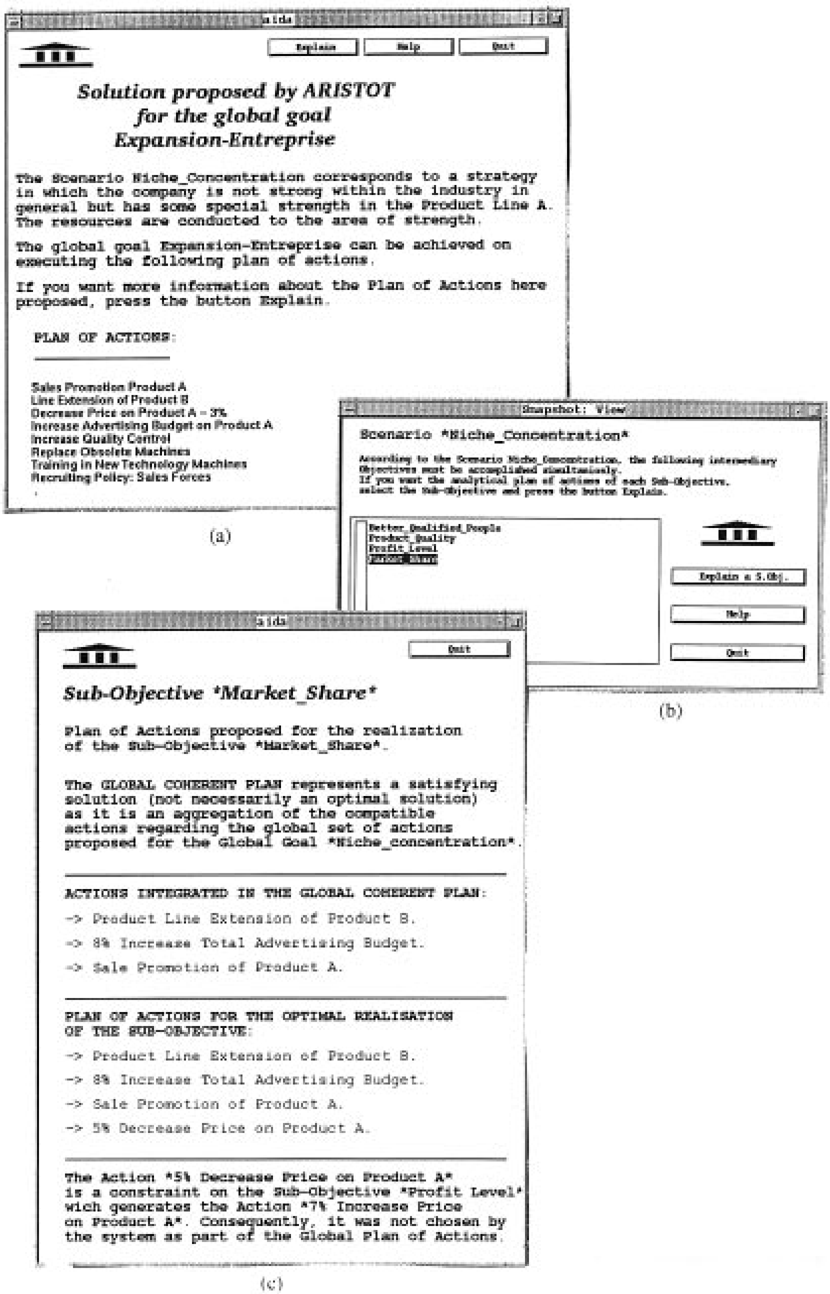

Figure 6. ARISTOTE Output.

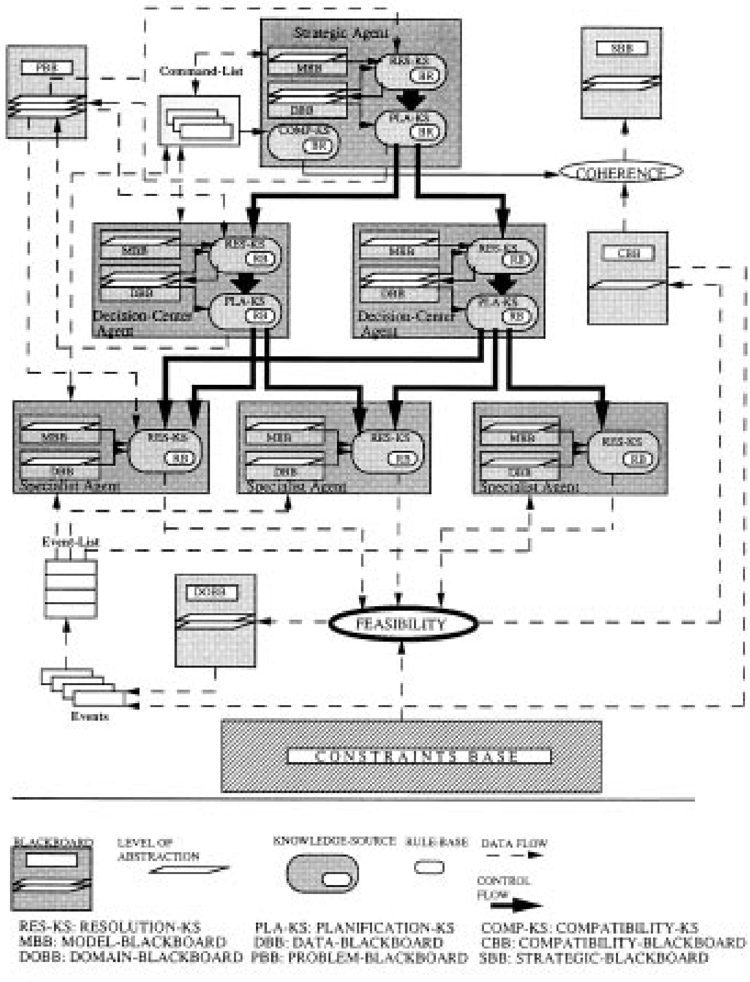

Figure 7. The architecture of ARISTOTE.

- Three types of agents—strategic agents, decision-center agents, and specialist agents—cooperate at three different hierarchical levels. These levels correspond to the three levels proposed in our framework (cf. section 5.3).
- Four types of blackboard are represented to allow communication between agents: the problem blackboard (PBB), the domain blackboard (DOBB), the compatibility blackboard (CBB), and the strategic blackboard (SBB).
- The constraint base contains the economic and environmental constraints of the domain.

6.1. Agent architecture

Each agent is considered as having two different parts: the Intelligent System (IS) and the Cooperative Layer (CL). Figure 8 illustrates the agent architecture. Each agent is seen as the union of the Intelligent System (IS) and the Cooperative Layer (CL).

The Intelligent System is responsible for the useful work of the agents (e.g., elementary action generation for Specialist Agents, proposal generation for Decision Center Agents, skeleton plan generation for the Strategic Agent). It is composed of the Resolution KS and two local blackboards: the Data Blackboard (DBB) and the Model Blackboard (MBB).

The Cooperative Layer is responsible for cooperation with other agents and for the control of the Intelligent System tasks. The Planning and Coordination module is a Knowledge Source, called Planification KS. It represents the knowledge about other agents of the community (its acquaintances) and the tasks they are able to do. It is also responsible for deciding when and how to cooperate with other agents. The Competences Module supports the knowledge that the agent has about itself.

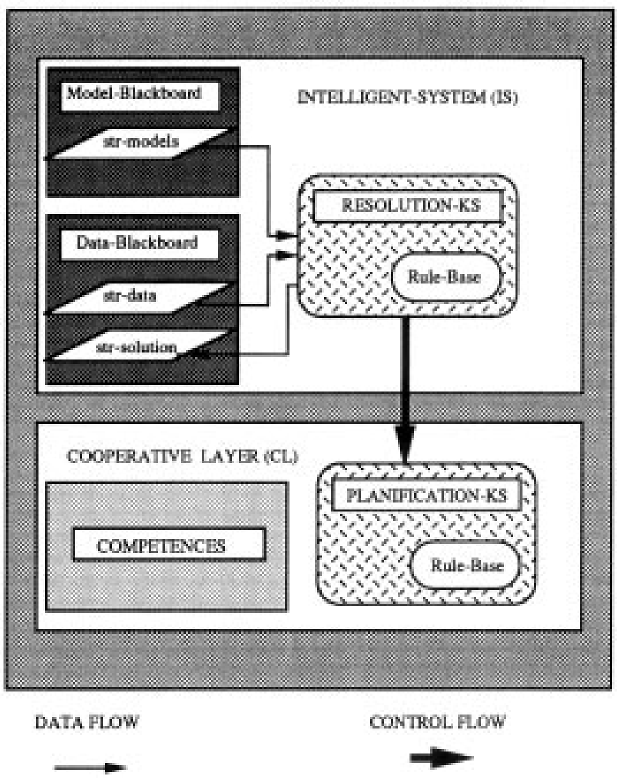

Figure 8. Agent structure.

In the ARISTOTE system each agent is implemented as an object which is an instance of one of the three classes of agents.

In a *Strategic Agent* (STA) the IS module is divided into two modules (cf. Figure 9). The first module, the Resolution KS, allows the STA to decompose the global objective into a set of subobjectives and to propose a skeleton plan (or scenario) to solve the problem. In its present state, the Resolution KS chooses among several predefined scenarios using its rule base. The other module, the Compatibility KS, is triggered at the end of the whole process, during the bottom-up coordination phase. It uses the compatibility criteria to test compatibility between elementary actions stored in the Compatibility Blackboard.

The Cooperative Layer is responsible for allocating the subobjectives to the Decision-Center Agents (DCA). The Planification KS is triggered at the end of the first IS module task. It looks for Decision-Center Agents able to achieve the corresponding subobjective. It is done by looking for the right competence attribute of the DCAs using a pattern-matching process. For example, the subobjective "market share" is matched against the value "market share" of the attribute "competence" of the corresponding DCA. It then generates a command to be sent to the chosen DCA. This command is also stored in the DCA mailbox and in the command list.

A *Decision-Center Agent* (DCA) has three functions: (1) It refines the partial solution, generates proposals in order to achieve the corresponding subobjective, and stores them in the "specialist level" of the Problem Blackboard. The Resolution KS contains *knowledge*

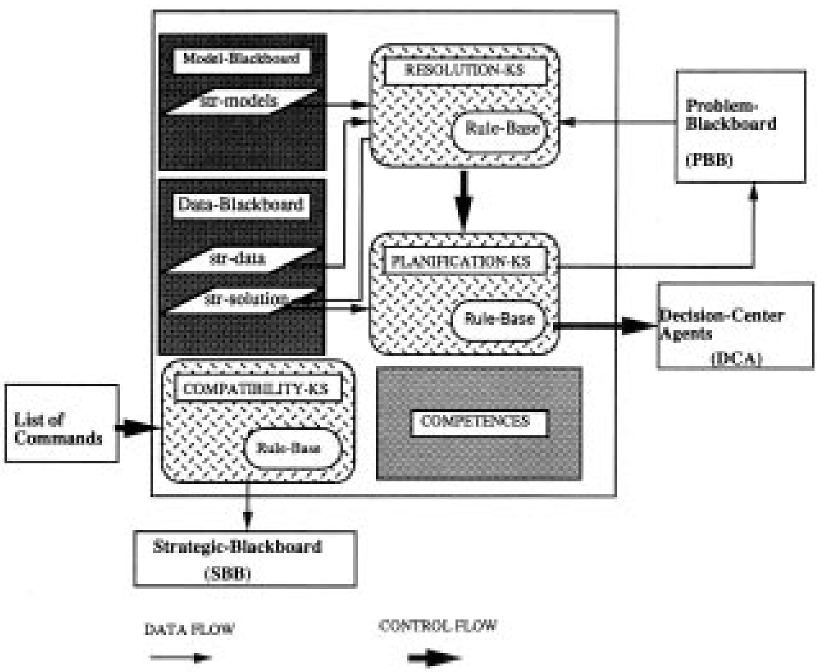

Figure 9. Strategic agent structure.

for proposal creation. It may use the local Data Blackboard and Model Blackboard to do so. (2) It chooses the appropriate Specialist Agent (in our example, the price specialist) by reading the "competence" attribute of this type of agent; and (3) it creates the "sub-objective" level of the corresponding Domain Blackboard and sends a command to the chosen Specialist Agent with the proposal location for it to pick it up. A proposal is represented as an object which is an instance of the class called "proposal" presented below:

proposal

name	=	⟨proposal name⟩
sub-objective	=	⟨subobjective name⟩
goal	=	⟨goal level⟩
COP	=	⟨coefficient of objective priority $\in [0, 1]$⟩
specialist	=	⟨competence of a specialist⟩
hypothesis-name	=	⟨"subobjective" level hypothesis of the DOBB to which elementary actions must be linked⟩

An example of a proposal is given below:

proposal

name	=	proposal-4
sub-objective	=	market share
goal-level	=	+3%
COP	=	0.6
specialist	=	price
hypothesis-name	=	subobjective-1

In proposal-4 the subobjective is detailed: to achieve a market share increase of 3%, the price specialist has to be activated. The attribute "hypothesis-name" indicates that the elementary actions will be linked to the hypothesis "subobjective-1" in the Domain-Blackboard. 0.6 is the coefficient of objective-priority of the subobjective "market-share".

A *Specialist Agent* (SPA) has the general structure of Figure 8 except for the planification KS that does not exist. The two local blackboards contain part of the domain knowledge. The Intelligent System (IS) proposes elementary actions corresponding to proposals created by the decision centers that have chosen this specialist to participate in a subobjective solution. It also has *knowledge* to propose a coefficient of action priority CAP which represents the importance of the elementary action for the achievement of the corresponding subobjective. For example, an elementary action proposed by the agent specialized in pricing policy is represented by the following object "elem-act-1." It belongs to the class "elem-act".

An elementary action is represented as an instance of the "elem-act" class. The structure of this class is represented follows:

elem-act		
name	=	⟨action name⟩
objective	=	⟨objective to be achieved⟩
COP	=	⟨coefficient of objective priority⟩
entity	=	⟨name of the entity⟩
attribute-name	=	⟨an attribute of the entity⟩
attribute-value	=	⟨value for this attribute⟩
impact	=	⟨impact of the action on the objective $\in \{+, -\}$⟩
CAP	=	⟨coefficient of action priority $\in [0, 1]$⟩
CGP	=	⟨CAP*COP⟩

An example of a elementary action is given below:

elem-act-1		
name	=	elem-act-1
sub-objective	=	market-share increase
COP	=	0.4
entity	=	product-A
attribute-name	=	price
attribute-value	=	-10%
impact	=	"+"
CAP	=	0.6
CGP	=	0.24

The values of the attributes "entity," "attribute-name," "attribute-value," "CAP," and "impact" are inferred by the specialist using its rule base; the value of the attributes "objective" and "COP" come from the proposal sent by a Decision-Center agent; the value of the attribute "CGP" is the product of COP and CAP computed by a demon method attached to this attribute.

A Specialist Agent may be chosen by several Decision Center Agents at the same time. The "Competence" module states which task the SPA is able to do. The agent writes proposed elementary actions in the Domain Blackboard and in the Compatibility Blackboard. Its knowledge is represented by rules that are triggered by matching premisses against the constraints of the constraint base. Some of the constraint values, such as product cost, are computed by a mathematical model (stored in the Model Blackboard of the BDD) as a function of the purchase cost and the production cost (stored in the Data Blackboard). As an example, this function is written in SMECI as follows:

```
(defSmethod {Product}:product-cost (product) ()
        (+ production-cost.product   purchase-cost.product))
```

To take an example, a rule of the Specialist Agent "price" corresponding to the sub-objective "increased market share" is written as follows:

rule: R1
if x is a proposal and y a product
if
objective (x) = market-share-increase
and goal-level (x) = +3%
and name (y) is product-A
and product-cost (y) is high
and competition (y) is strong
then
create an elementary-action z
with

objective (z)	*= objective (x)*
COP (z)	*= COP (x)*
entity (z)	*= product-A*
attribute-name (z)	*= price*
attribute-value (z)	*= −10%*
impact (z)	*= +*
CAP (z)	*= 0,6*
CGP (z)	*= CAP (z)* COP (x)*

end-rule

This says that in order to solve the subobjective "increase market share by 3%" and considering the economic constraints (product-cost and competition) on product A, the price of product A should be decreased by 10%; it also gives the coefficient of action priority of this elementary action. *This knowledge is given by the experts during the acquisition phase*.

6.2. Blackboards

Communication between agents is carried out by means of two centralized structures called blackboards (the Domain Blackboard and the Compatibility Blackboard) and by decentralized messages sent by agents to each other and stored in their mailboxes of the Problem Blackboard.

The *Problem Blackboard* (PBB) contains the problem's initial data (the global objective) which is given interactively by the user. It contains the global objective name, the mailboxes of Decision-Center Agents and the mailboxes of the Specialist Agents (cf. figure 10).

The Problem Blackboard allows communication and distribution of decision making among agents of the three levels of decision functions. This data structure is specific and important because it shows at any time of the problem-solving process who is doing what, when (more details are given in sections 6.4.3 and 7), which agents are active, and what the skeleton plan of actions is. The structure of the distributed decision making process is thus available at all times during the problem-solving process.

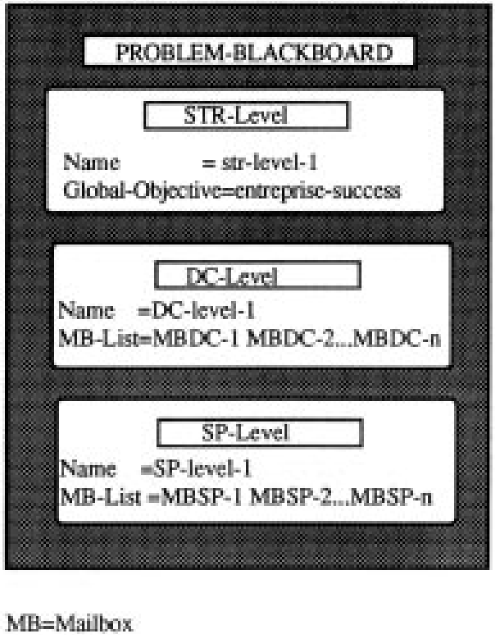

Figure 10. The Problem-Blackboard.

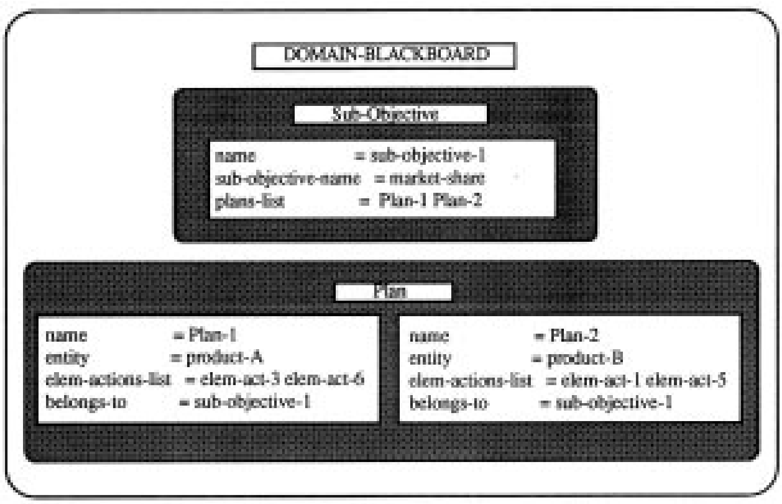

Figure 11. The Domain Blackboard.

The *Domain Blackboard* (DOBB) contains feasible elementary actions proposed by the Specialist Agents to achieve a subobjective (cf. Figure 11). An elementary action is feasible if it satisfies the domain's constraints. The set of elementary actions represents part of the solution to the global problem and a local optimum for the corresponding subproblem, as is explained in section 5.4.

The DOBB is created dynamically. A DOBB is divided into two abstraction levels: (1) the "subobjective" level, which contains the names of the subobjectives and the list of plans; the hypotheses of this level are created when the DC Agents are activated; (2) The "plan" level, which contains the decomposition of each plan into a set of elementary actions. Elementary actions are sorted by entity and subobjective. The hypotheses of this level are created when the Specialist Agents are activated. Inverse "belongs-to" links exist between the two levels to allow updating of the first level when instances of actions are created at the second level.

The *Compatibility Blackboard* (CBB) contains the set of feasible elementary actions proposed by all the Specialists Agents (cf. Figure 12). The elementary actions are sorted by entity and attribute in order to facilitate the compatibility mechanism. The objective of this blackboard is to allow the compatibility criteria to be applied on elementary actions.

As pointed out previously, conflicts are not solved at the specialist level because (1) we want to keep the local optimum for each subobjective (for explanatory purposes), and (2) the actions proposed by the Specialist Agents are generated asynchronously at different periods of time.

The *Strategic Blackboard* (SBB) is divided into two levels: the "coherent solution level" and the "non-coherent solution level." It contains either a plan of elementary actions representing a feasible and coherent solution to the global problem (stored in the "coherent solution" level) or incompatible actions, which means that there is no coherent plan of actions. In this case the system stores in the "non-coherent solution level" the plan, as well as the corresponding subobjectives that are consequently considered as incompatible.

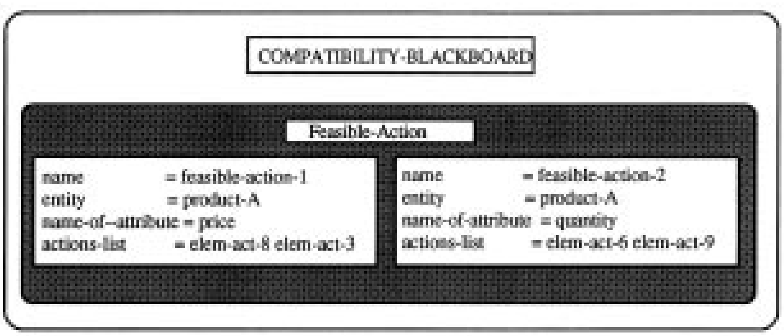

Figure 12. The Compatibility Blackboard.

6.3. The constraint base

This base contains all the environmental constraints of the domain: government regulations, historical and current industry information, historical and current information on competitors and customers, political data, demographic data, etc. The Specialist Agents match their knowledge against these constraints to generate feasible elementary actions (cf. section 5.4). In its present state, only information on competitors and customers is recorded.

6.4. Communication between agents

6.4.1. Events. Creation or updating of a hypothesis in the Domain Blackboard or in the Compatibility Blackboard creates an event which is stored in the event list. In our system events are not used in the same way as in a blackboard system, that is to say, to help select the next knowledge source to be activated. In ARISTOTE an event allows, us (1) to sort elementary actions by subobjective and attribute and store them in the "plan level" of the Domain BB, and (2) to sort elementary actions by entity and attribute (for example, all the actions proposed on the price of product-A) and store them in the "feasible-action level" of the Compatibility BB. The specificity of this event concept allows the specialist agents to self-organize their own elementary actions, given the local objectives and the global objective.

An event is an object which is an instance of the following class:

event

name	=	⟨event name⟩
level	=	⟨level in the blackboard (CBB or DOBB)⟩
hypothesis	=	⟨hypothesis name to be created or updated⟩
sub-objective	=	⟨subobjective name⟩
entity	=	⟨entity name⟩
attribute-name	=	⟨attribute of the entity⟩

6.4.2. The event list. The event list contains events which indicate changes in the Domain Blackboard and in the Compatibility Blackboard. A specialist agent checks this list before updating the Domain Blackboard and the Compatibility Blackboard in order to decide the type of action to be done: creating a new hypothesis or updating an existing one (cf. Figure 13).

6.4.3. Messages. When an agent at the strategic level or the decision-center level sends a request to a lower-level agent, it enters a command in the command list. A command is an object, an instance of the class called "command." For example, the command sent by the Strategic Agent to a Decision-Center Agent in order for it to achieve the sub-objective "increased market share increase" is called "command-23" and is as follows:

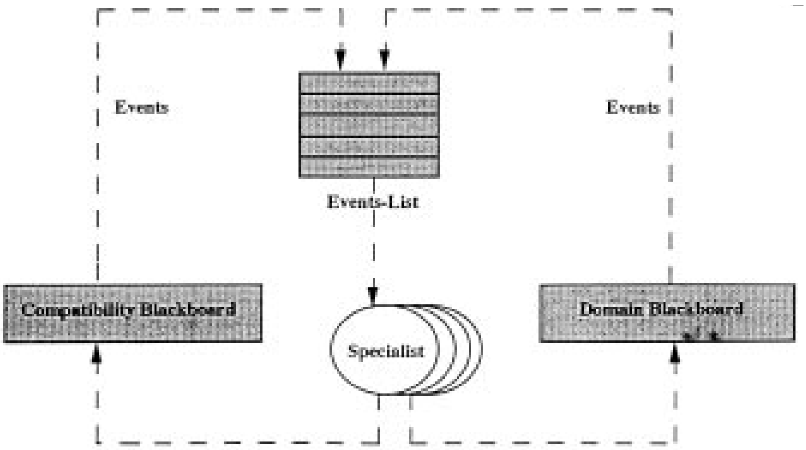

Figure 13. Domain and Compatibility Blackboard updating.

command-23

name	=	command-23
from-agent	=	strategic-agent
to-agent	=	decision-center-1
work-to-do	=	market-share-increase

The agent puts the command name in the command list and in the slot "no-command" of the appropriate agent mailbox. Communication between the agents at each level is done through mailboxes which are listed in the corresponding levels of the Problem Blackboard.

The command is removed from the list and from the mailbox when an agent has finished its job. When the list is empty, which means that the top-down phase is over, and that all possible elementary actions have been generated, a demon is triggered, and activates the compatibility procedure. This procedure is implemented as a set of rules and methods in the Compatibility-KS of the Strategic Agent.

7. Problem solving process

The system operates through different phases (cf. Figure 14). Phases are sequential, but, at each level, the agents may be activated in parallel. In addition, the compatibility phase is triggered at the end of the planning process during the bottom-up coordination.

Phase 1: User-computer interaction. The system asks the user to define the global objective. The global objective is stored in the strategic level of the Problem Blackboard (PBB).

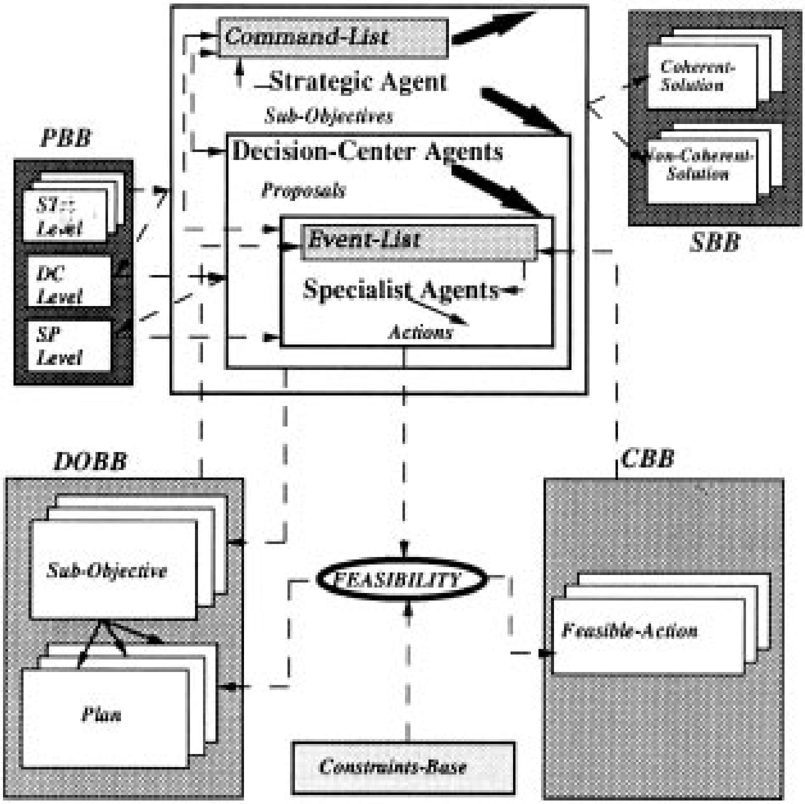

Figure 14. How ARISTOTE runs.

Phase 2: Activation of the Strategic Agent. (1) The Resolution KS retrieves the global objective name from the strategic level of the PBB and helps the user to generate a list of subobjectives which must be achieved in order to achieve the global objective (i.e., scenario 1 = increase market share, decrease financial expenses; scenario 2 = increase market share, improve notoriety, etc). (2) The Planification KS looks for the Decision-Center Agents able to solve the subobjectives (more precisely, it looks at the "competences" attribute of the DC Agent) and assigns a subobjective to the corresponding DC Agent. The names of the subobjectives are sent by a command to the DC mailboxes located at the "decision-center level" of the PBB. This command is also stored in the command list.

Phase 3: Activation of Decision-Center Agents. (1) The Resolution KS retrieves the name of the subobjective from the DC mailbox located in the PBB. According to its knowledge, it generates proposals. (2) The Planification KS allocates proposals to Specialist Agents. It looks for the Specialist Agent able to propose an elementary action ("competences" attribute) and passes on the proposal related to the selected agent. Proposal names are sent by a command to the mailboxes at the "specialist-level" of the PBB. This command is also stored in the command list. The command sent by the Strategic Agent is removed from the command list and from the mailbox.

Phase 4: Activation of Specialist Agents and feasibility phase. The Resolution KS of the Specialist Agent generates elementary actions and tests their feasibility against the constraint base. The feasible elementary actions are stored simultaneously in the Domain BB and in the Compatibility BB. The commands sent by the DCA are removed from the list of commands and from the mailbox.

Phase 5: Compatibility phase. When the list of commands is empty the Compatibility KS of the Strategic Agent is activated. The elementary actions stored in the CBB are tested for compatibility using the coefficients of global priority.

- If elementary actions are incompatible, the system stores the incompatible actions along with the corresponding subobjectives in the "non-coherent solution level" of the Strategic BB.
- If elementary actions are compatible, the system gives its recommendations as a plan of actions to be performed (cf. Figure 6a for the output). The elementary actions of this plan are stored in the "coherent level" of the Strategic BB.

Phase 6: The ARISTOTE system asks the user if he/she wants to study another global objective or if he wants to evaluate another scenario for the same global objective. The system starts a new cycle again (as described in section 6.1).

8. Implementation

The system has been implemented using the SMECI environment based on LISP. In this environment, the graphic library AÏDA enables the developer to use a totally object-oriented approach to create sophisticated graphic interfaces (SMECI and AÏDA are trademarks of ILOG). The formalism of SMECI is object-oriented. The concepts handled by the shell are defined in terms of *categories*, which represent classes of objects. Specific methods are associated with each category. The reasoning processes of SMECI use *rules* that are grouped to form *tasks* where a task can be thought of as an object that describes a special process. SMECI deals with the various tasks according to an *agenda*, which can be controlled and modified dynamically by the user.

9. Concluding remarks

The objective of this article was to show the feasibility of a Multi-Agent Decision Support System for automated Strategic Decision processes. One of the main problems was to achieve coherence and coordination among decisions made locally by different agents at different levels. In order to formalize this problem, we defined two basic concepts, feasibility and compatibility of decision processes, and then we implemented these processes by means of a hybrid architecture. The ARISTOTE system involves three types of independent agents which cooperate to propose a plan of coherent actions to achieve a global objective given by the user. Communication between agents is carried out by means of a list of commands, the cooperative layer of the agents, and four types of blackboard. In order to reach a solution the system performs decomposition and coordination from the highest level to the lowest level (top-down coordination) and coherence control from the lowest level to the highest one (bottom-up coordination).

The use of a multi-agent system based on a hybrid communication offers three kind of advantages:

- Software engineering characteristics such as modularity and flexibility are guaranteed by a multi-agent system allowing easy evolution of the application.
- Distribution of knowledge: each problem is clearly separated from the others and the knowledge involved is well defined.
- Justification of the decision making: at any moment the state of the whole system is available—current position in the plan, current step, current context, significant value of objects, and local optimum for the achievement of a subobjective. The system can justify its advice to the user, which is fundamental for efficient man–machine cooperation.

One future area of study involves adding more parallelism to the system. We have been assuming that the user gives one global objective at a time and that the system evaluates one scenario at a time. The coordination process becomes more complex if scenarios are evaluated in parallel with Decision-Center Agents taking care of a subobjective whatever the origin scenario. Efficiency would be improved because fewer Specialist Agents would be idle, and the same subobjectives would only be evaluated once, generating elementary actions which could be used in parallel in different scenarios.

Another area of study deals with introducing learning facilities in the cooperative process, using case-based reasoning. When the system detects contradictory actions, it could suggest another decomposition based on its experience and history. It could review its past experiences in a search for similar patterns that might be useful in solving present situations. It could also store the found solutions in the case base.

Another future research area involves comparing various organizational structures. The multi-blackboard approach offers several advantages such as an efficient way to model the hierarchical distributed decision making process, but a completely decentralized structure with messages passing between agents may improve the modularity aspect of the system. System performance could be improved by learning to identify the utility of different organizational structures.

References

Antony, R. N. (1965). *Planning and Control Systems. A Framework for Analysis*. Boston: Harvard Business School Press.

Armstrong, S. J. (1982). "The Value of Formal Planning for Strategic Decisions: Review of Empirical Research," *Strategic Management Journal* 3, 197–211.

Badiru, A. B., P. S. Pulat, and M. Kang. (1993). "DDM: Decision Support System for Hierarchical Dynamic Decision Making," *Decision Support System* 10, 1–18.

Berman, S. G., and R. F. Kautz. (1990). "Compete: A Sophisticated Tool that Facilitates Strategic Analysis," *Planning Review* (July) 35–39.

Bond, A. H., L. Gasser, eds. (1988). *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann Publishers, Inc.

Chung, C. H., Lang, Jr., and K. N. Shaw. (1989). "An Approach for Developing Support Systems for Strategic Decision Making in Business," *Omega International Journal of Management Science* 17(2), 135–146.

Clark, D. N. (1992). "A Literature Analysis of the Use of Management Science Tools in Strategic Planning," *Journal of the Operational Research Society* 43(9), 859–870.

Cyert, R. M., and J. G. March. (1963). *A Behavioral Theory of the Firm*. Englewood Cliffs, NJ: Prentice-Hall.

Davis, R., and R. G. Smith. (1983). "Negotiation as a Metaphor for Distributed Problem Solving," *Artificial Intelligence* 20(1), 63–109.

Durfee, E. H., V. R. Lesser, and D. D. Corkill. (1987). "Coherent Cooperation Among Communicating Problem Solvers," *IEEE Transactions on Computers* C-36, 1275–1291.

Fox, M. S. (1981). "An Organizational View of Distributed Systems," *IEEE Transactions on Systems, Man and Cybernetics* SMC-11(1)(January).

Greenley, G. E. (1989). *Strategic Management*. London: Prentice-Hall.

Hayes-Roth, B., and F. A. Hayes-Roth. (1979). "A Cognitive Model of Planning," *Cognitive Science* 3(4), 275–310.

Hayes-Roth, B., and M., Hewett. (1988). "BB-1: An Implementation of the Blackboard Control Architecture," *Blackboard Systems*. 297–313. Reading, MA: Addison Wesley.

Holloway, C. (1983). "Strategic Management and Artificial Intelligence," *Long Range Planning* 16(5), 89–93.

Kornfeld, W. A., and C. E. Hewitt. "The Scientific Community Metaphor," *IEEE Transactions on Systems, Man and Cybernetics* SMC-11(1)(January), 24–33.

Laâsri, H., B. Maître, T. Mondot, F. Charpillet, and J. P. Haton. (1988). "ATOME: A Blackboard Architecture with Temporal and Hypothetical Reasoning." In *Proceedings of the 8th European Conference on Artificial Intelligence (ECAI-88)*, Munich, Germany, August 1–5, pp. 5–10.

Lehner, F. (1992). "Expert Systems for Organizational and Managerial Tasks," *Information & Management* 23, 31–41.

Lesser, V. R., and L. D. Erman (1980). "Distributed Interpretation: A Model and Experiment," *IEEE Transactions on Computers* C-29(12) (December), 1144–1163.

Lesser, V. R., and D. D. Corkill. (1983). "The Distributed Vehicule Monitoring Testbed: A Tool for Investigating Distributed Problem Solving Networks," *AI Magazine* (Fall), 15–53.

Lesser, V. R., and D. D. Corkill. (1987). "Distributed Problem Solving." In Stuart C. Shapiro (ed.), *Encyclopedia of Artificial Intelligence*. New York: John Wiley & Sons, pp. 245–251.

Malone, T. W. (1988). "Modeling Cooperation in Organizations and Markets." In A. H. Bond and L. Gasser (eds.), *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann Publishers, Inc.

March, J. G., and H. A. Simon. (1958). *Organizations*. New York: John Wiley & Sons.

Mesarovic, M. D., D. Macko, and Y. Takahara. (1970). *Theory of Hierarchical, Multilevel, Systems*. New York: Academic Press.

Mintzberg, H., D. Raisinghani, and A. Théorêt. (1976). "The Structure of 'Unstructured' Decision Processes," *Administrative Science Quarterly* 21.

Mockler, R. J. (1989). *Knowledge-Based Systems for Strategic Planning*. (1989). Englewood Cliff, NJ: Prentice-Hall.

Moraïtis, P. (1994). “Multiagent Paradigm and Distributed Decision Making.” PhD Dissertation (in French), University of Paris-Dauphine, Paris.

Newell, A., and H. A. Simon. (1961). “Computer Simulation of Human Thinking,” *Science* 134 (December), 2011–2017.

Nutt, P. C. (1993). “The Identification of Solution Ideas During Organizational Decision Making,” *Management Science* 39(9), 1071–1085.

Paradice, D. B. (1992). “SIMON: An Object-Oriented Information System for Coordinating Strategies and Operations,” *IEEE Transactions on Systems*, *Man and Cybernetics* 22(3), 513–525.

Pinson, S. (1992). “A Multi-Expert Architecture for Credit Risk Assessment: the CREDEX System.” In D. E. O’Leary and P. R. Watkins (eds.), *Expert Systems in Finance*. New York: Elsevier Science Publishers.

Rasmussen, J. (1988). “A Cognitive Engineering Approach to the Modelling of Decision Making and its Organization in Process Control, Emergency Management,” CAD/CAM, Office Systems, Library Systems). In W. B. Rouse (ed.), *Advances in Man-Machine System Research*, Vol. 4, Greenwich, CT: JAI Press.

Rasmussen, J. (1991). “Modelling Distributed Decision Making.” In J. Rasmussen, B. Brehmer, and J. Leplat (eds.), *Distributed Decision Making*: *Cognitive Models for Cooperative Work*. John Wiley & Sons Ltd.

(Saaty 80) Saaty, T. L. (1980). *The Analytic Hierarchy Process*. New York: McGraw-Hill.

Shakun, M. F. (1991). “Airline Buyout: Evolutionary Systems Design and Problem Restructuring in Group Decision and Negotiation,” *Management Science* 37(10), 1291–1303.

Simon, H. A. (1975). *Administrative Behavior*. New York: Macmilan Company.

Simon, H. A. (1969). *Science of the Artificial*. Cambridge: MIT Press.

Steeb, R., S. Cammarata, F. A. Hayes-Roth, P. W. Thorndyke, and R. B. Wesson. “Architectures for Distributed Air-Traffic Control.” In A. H. Bond and L. Gasser (eds.), *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann Publishers, Inc.

Thietart, R. A., and M. Bergadaa, (1988). “STRADIN: A Strategic and Dynamic Inter-personal Decision Making Process.” Conference of Strategic Management Society, Amsterdam.

Thompson, J. D. (1967). *Organizations in Action*. New York: McGraw-Hill.