Integrating new information and communication technologies in a group decision support system

Nikos Karacapilidis*

Department of Computer Science, University of Cyprus, P.O. Box 20537, 1678 Nicosia, Cyprus

Received 31 March 1999; received in revised form 1 January 2000; accepted 12 March 2000

Abstract

We view group decision making as a collaborative process, where decision makers can establish a common belief on the dimensions of the problem by following a series of well-defined communicative actions. Having first defined these actions, this paper reports on the exploitation of recent advances in information and communication technology, which can be used to: (i) remove the communication impediments among spatially dispersed decision makers; (ii) efficiently elicit and represent the domain of knowledge; (iii) develop efficient mechanisms to structure and consistently maintain the decision analysis; and (iv) automate the decision making process itself. Automation concerns coherence and consistency checking, detection of contradictions, truth maintenance, and information retrieval techniques. © 2000 IFORS. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Group decision support system; Argumentation; Multi-criteria decision making; Decision modeling; Information and communication technologies

1. Introduction

Group Decision Support Systems (GDSSs) have been defined as interactive computer-based systems which facilitate the solution of ill-structured problems by a set of decision makers, working together as a team (Kreamer and King, 1988). The main objective of a GDSS is to augment the effectiveness of decision groups through the interactive sharing of information between group members and the computer. This can be achieved by (i) removing
communication impediments, and (ii) providing techniques for structuring the decision analysis and systematically directing the pattern, timing, or content of the discussion. Major issues arising during the development of such a system include effective work organization in order to improve coordination, and use of communication technology to make decision making more efficient.

Group decision making (GDM) usually raises a lot of intricate debates and negotiations among participants. Conflicts of interest are inevitable and support for achieving consensus and compromise is required. Each decision maker may adopt and, consequently, suggest his/her own strategy that fulfills some goals at a certain level. Opinions may differ about the relevance or value of a proposition when deciding an issue. Decision makers may have arguments supporting or against alternative solutions. In addition, they have to confront the existence of insufficient and excessive information simultaneously. In other words, for some parts of the problem, relevant information which would be useful for making a decision is missing, whereas for others, the time needed for the retrieval and comprehension of the existing volume of information is prohibitive. Furthermore, participants need appropriate means to assert their preferences, which often are expressed in qualitative terms. Due to the above, proper definition of all acts decision makers perform during such processes and provision of procedures for automation of data processing, especially in data intensive situations, are of high importance.

Decision makers are not necessarily proficient in computer science and information technology; they need appropriate tools in order to easily follow the processes involved. Such tools should stimulate their participation giving them an active role. This parallels the vision of the DSS community pioneers, that is, by supporting and not replacing human judgement, the system comes in second and the users first.

GDM admittedly falls in the category of ‘wicked’ problems (Rittel and Webber, 1973), a class of problems that can be addressed through discussion and collaboration among the agents involved. Consensus emerges through the process of collaboratively considering alternative understandings of the problem, competing interests, priorities and constraints. The application of more formal modeling and analysis tools is impossible before the problem can be articulated in a concise and agreed upon manner. Computer-mediated GDM has been receiving growing interest in the last few years. Proliferation of Internet technologies enables development of such systems on the World-Wide Web platform. This is mainly due to its platform-independent communication framework and associated facilities for data representation, transmission and access. In particular, attention focuses on the implementation of argumentation support systems for different types of groups and application areas. Such systems address the needs of a user to interpret and reason about knowledge during a discourse.

For instance, QuestMap (Conklin, 1996) captures the key issues and ideas during meetings and creates shared understanding in a knowledge team. All the messages, documents, and reference material for a project can be placed on the whiteboard, and the relationships between them can be graphically displayed. Users end up with a map that shows the history of an online conversation that led to key decisions and plans. QuestMap was based on the gIBIS hypertext groupware tool (Conklin, 1992) which aimed at capturing the rationale of a design process. Euclid (Smolensky et al., 1987) is another system that provides a graphical
representation language for generic argumentation. *JANUS* (Fischer et al., 1989) is based on acts of critiquing existing knowledge in order to foster the understanding of a design process. *SEPIA* (Streitz et al., 1989) is a knowledge-based authoring and idea processing tool for creating and revising hyperdocuments that views authoring as a design process. Finally, *Belvedere* (Suthers et al., 1995) uses a rich graphical language to represent different logical and rhetorical relations within a debate, originally designed to support students engaged in critical discussion of science issues. Although this category of systems provides a cognitive argumentation environment that stimulates discussion among participants, it lacks decision making capabilities.

Numerous web-based conferencing systems have also been deployed, such as *AltaVista Forum Center*, *Open Meeting*, *NetForum* and *UK Web’s Focus*, to mention some. They usually provide means for discussion structuring and user administration tools, while the more sophisticated ones allow for sharing of documents, on-line calendars, embedded e-mail and chat tools, etc. Discussion is structured via a variety of links, such as simple responses or different comment types (e.g., qualify, agree, example in *Open Meeting*) to a previous message. However, the above systems merely provide threaded discussion forums, where messages are linked ‘passively’, which usually leads to an unsorted collection of vaguely associated comments. As pointed out by the developers of *Open Meeting*, there is a lack of consensus seeking abilities and decision making methods (Hurwitz and Mallery, 1995). Furthermore, this category of systems is not based on a well defined set of users’ communicative actions.

A prerequisite for computer-mediated GDM tools is the ability for the computer to understand (at least partially) the dialogue in a decision-related argument between people, and the discourse structure used in presenting supportive material in a document. This requires a computational model of the discourse acts used in these cases. Although there has been work in Artificial Intelligence (AI) on dialogue and discourse in collaboration and negotiation (see, for instance, De Michelis and Grasso, 1994; Di Eugenio et al., 1997; Grosz and Sidner, 1990; Merin, 1997), that work is not sufficient for modeling dialogues in the GDM context. More specifically, it is rather general and not explicitly oriented towards real-life collaborative decision making environments.

The next section of this paper illustrates the discourse acts model that we suggest. In the sequel, we present the modules that constitute our framework for computer-mediated GDM. The argumentation-based group decision support module and its associated machinery for aiding decision makers to reach a decision are described next. Automation in our system includes mechanisms that not only structure the related discussion, but also provide truth maintenance, reasoning and consistency mechanisms. Then, we report on further assistance tools with information retrieval, natural language processing and argument building features. The last but one section comments on previous research work and various GDM aspects, while the last one concludes the paper.

2. Modeling the discourse

Attempts to model the dialogue process, mainly coming from the AI discipline, generally presume that all participants are being cooperative and honest. Agents act optimally with
respect to the information at hand, and do not consider how others might interpret their actions. For instance, Sidner (1994) presents a model of collaborative negotiation based on the idea of establishing mutual beliefs, that is, beliefs that agents hold in common. This model rests upon the non-existence of deception, and appears fragile in the presence of mutual misunderstanding. The work of Cohen and Levesque (1990) and of Smith and Cohen (1996) is very similar to Sidner’s work, but relies in addition on the primitive notion of joint goals. Based on Searle’s idea (Searle, 1969) that requesting something means that one is attempting to get an agent to perform an action, they define a set of illocutionary acts (the act performed as the result of a speaker making an utterance; its effect is a perlocutionary act) in terms of agent’s mental states. Core and Allen (1997) introduced a scheme for annotating communication acts in dialogue, which ignores the formation of opinions by hearers about speakers and gives a single coding for each utterance. It is generally the case, however, that utterances can and should be considered to perform multiple functions.

An understanding of the implications in the GDM process requires a model of the mental attitudes of the agents involved (their beliefs, desires, intentions, goals, etc.) as they pertain to the task at hand. Further, it requires a model for the particular form of discourse acts that agents use to communicate their knowledge and intentions, and affect the attitudes of others. In addition, it requires a model of the actions that relate to the argument process itself. The works described above take this rational agent approach, as we do. The model presented here, however, is explicitly oriented towards GDM.

2.1. Objects and relations, states and actions

The first question to consider is the primitive components of the model. This includes objects of discourse such as the alternatives among which agents must choose, the criteria for evaluating these alternatives, and methods for applying these criteria (i.e., the decision or objective function). Participants in a decision process are also ‘objects’ since their interactions must be modeled. A complete model should also allow comparisons between criteria and alternatives along numerical scales, so it can represent notions such as “cost is more important than speed” or “the total cost must be less than 5,000,” in the context of a car purchase discussion. The basic language that we propose is a sorted first-order predicate calculus with extensions for numerical comparison. We will use terms such as ‘agt1’ or ‘agtn’ to denote individual participants, and ‘group’ to denote the set of all agents.

To capture the dynamics and evolution of a decision, the model must include notions of time and states, and of actions that change states. States represent a coherent situation (they may be already past or current). In order to consider alternatives to the existing state of affairs, we introduce the notion of a hypothesis, which is an artificial state of the world that may not exist, and may not be possible. It provides a framework for hypothetical reasoning by an agent. If agent agt1 hypothesizes the state S1, then we write: (HYP agt1 S1).

Actual, future and hypothetical states, and temporal intervals are reified objects in our proposal, and all propositions must have them. However, for simplicity, they are only included in the examples here when pertinent. Actions are defined in terms of the changes that they evoke in states. The simplest acts in our system are the actions of telling something to an agent, and its complement of hearing what an agent is saying: (TELL agt1 agt2 P) and
Finally, in order to describe an action, we consider the effects it will have. These effects are the difference between a start state and the new state that the action produces. We represent that an act $A$ in state $S_0$ results in state $S$ as: $(\text{EFFECT } A S_0 S)$.

2.2. Primitive mental attitudes

When agents interact with each other in a dialogue, there is a number of mental attitudes that dictate the form of their interaction and their long-term behavior throughout the dialogue. The attitudes can concern states of the world or actions that affect the world. The most basic of these is the attitude of belief. In the context of AI and in this article, belief is used as a global term that covers the notion of knowledge as well as the notion of information in which we do not have complete confidence. Presuming that this information can be expressed as a proposition in, say, a first-order predicate calculus, then belief can be considered to be a modal operator relating an agent to a belief (see Ballim, 1992; Ballim and Wilks, 1991 for a more detailed description). So, for example, the belief held by an agent $agt_1$ that ISDN lines are fast might be expressed as: $(\text{BEL } agt_1 \forall x \ [\text{ISDN}(x) \rightarrow \text{fast}(x)])$. It is often convenient to refer to some belief as being commonly held between a group of agents, that is, we talk about a mutual belief. If $P$ is such a belief, this will be expressed as: $(\text{MB } agt_1 \ldots agtn P)$

Agents act in a world that is not always the same as the world in which they would like to be. Desires are used to express the wishes of the agent about the state of the world. They may also be expressed as modal operators over propositions. Thus, the desire that MBONE software will be used (for some purpose), might be expressed in the following way: $(\text{DES } agt_1 \exists x \ [\text{use}(x) \land \text{MBONE}(x)])$.

Although agents might desire a particular state of affairs to exist, they are not obliged to act upon those desires. For example, we might all desire a care-free life, but our responsibilities would stop us from acting upon this desire by (for example) dropping out of the society. States of affairs that we wish to exist, and, further, towards which we actively aspire are referred to as goals. If an agent had, as a goal that ISDN lines will be used in the context of a decision process, then this is written as: $(\text{GOAL } agt_1 \exists x \ [\text{use}(x) \land \text{ISDN}(x)])$.

The essential difference between desires and goals is that there is the notion of having an intention to act to achieve a goal, while this is not necessarily the case with desires. While it might be useful to thus define intention with respect to states of world ("I will work to achieve a certain state of the world") and then define a goal as desiring a state and intending that state, in our framework it is more reasonable that intention is related to actions, so that we intend to perform an action. So, if an agent intends to call another agent (the predicate call should be interpreted here as an action), we might represent this as: $(\text{INT } agt_1 \text{call}(agt_1, agt_2))$. The relation between goals and the methods of achieving these goals is a plan. A simple plan is a sequence of actions, but more complex plans can be defined which depend on contingencies and options.

For our domain, it is necessary to consider the relationships that the agents believe, hold between propositions; we introduce the notions of support and refute. If agent $agt_1$ believes that $P_1$ supports $P_2$, we write $(\text{BEL } agt_1 (\text{SUPP } P_1 P_2))$, while if $agt_1$ believes that $P_1$ disproves $P_2$, we write $(\text{BEL } agt_1 (\text{REF } P_1 P_2))$.

Such definitions are common in AI (see, for instance, Cohen and Levesque, 1990; Sidner,
While incomplete, they are sufficient to allow us to describe the model of dialogue acts in collaborative decision making.

2.3. Dialogue acts

The interactions between agents in a collaborative decision making process are codified using dialogue acts (note that multiple dialogue acts can be associated with a single utterance in a discussion). These can be used to interpret the discourse occurring between agents during the decision process, as well as to make inferences about their attitudes and to predict their likely future behavior. It should be made clear that the model of discourse acts we consider here (including the necessary conditions for them to occur) is not a complete model of human discourse, but an interesting subset which allows an analysis of the collaborative decision making process. While a detailed description of the model does not fall in the scope of this paper (see, instead, Ballim and Karacapilidis, 1998), in the rest of this section we outline most of the corresponding acts.

The start of any group decision process involves opening of an issue or, according to the model’s terminology, proposing a topic for consideration. To simplify matters, we will presume that the topic is always an action to be performed (e.g., buying a car, deciding between different medical treatments for a patient). According to the corresponding act (namely, consider), agt1 proposes (to all decision makers involved) an act $A$ which produces a desired state $S$; agt1 hypothesizes $S$ and has as goal that the same will hold for the group.

Another basic dialogue act is that of requesting something. We split up requesting into the following three sub-cases of agent agt1 making a demand: additional information is requested concerning a belief $P$, an opinion is demanded with respect to a belief $P$, and an act $A$ is requested concerning a belief $P$ (e.g. agt1 wants agt2 to agree or disagree with $P$).

It is often necessary (e.g., for inconsistency detection) to weigh alternative beliefs or to compare beliefs with values. In the dialogue, an agent may declare that he/she believes one criterion to be more important than another, or that some criteria must have values in a particular range. As such cases are very frequent, we introduce a compare dialogue act to treat them. In such cases, $\mathcal{R}$ is a comparison relation between beliefs (e.g., of the type “$P_1$ is more important than $P_2$”) or between a criterion $C$ and a value $V$ (e.g., of the type “$C$ should be equal to $V$”).

Using other acts, an agent agt1 may: (i) propose a belief to be accepted by another agent; (ii) express his/her agreement with a belief that was proposed by another agent; (iii) accept that a belief may be valid but he/she requires further proof before agreeing to accept it (acknowledge act); (iv) express that he/she disagrees with a belief; (v) agree with a belief $P_1$ and wish to give it further credence by proposing $P_2$ (corroborate act); (vi) disagree with a belief $P_1$ and wishes to further undermine it by proposing $P_2$ (challenge act); (vii) propose to another agent agt2 to stop considering a belief that has been proposed (discard act); (viii) propose (to agt2) a replacement belief $P_1$ for consideration as a clarification and replacement for belief $P_2$ which was previously proposed (clarify act); (ix) finally, agt1 may declare that he/she is not ready to accept the belief $P_1$ proposed by agt2 and, without explicitly disagreeing to $P_1$, he/she proposes a belief $P_2$ (counter-offer act).
3. Building a collaborative decision support system

Traditional decision making techniques, coming from areas such as mathematical economics, operations research, game theory and statistics, are built on the assumption of a predefined set of alternatives and criteria, and provide methods to quantify and aggregate subjective opinions (Keeney and Raiffa, 1976; French, 1988; Vincke, 1992). Everyday practices, however, make obvious that there is a lot of room for debate here. We view multi-agent decision making as a collaborative process, where agents may follow a series of communicative actions in order to establish a common belief on the dimensions of the problem (such dimensions may concern the choice criteria, the existing or desired alternatives, or the objective function, to mention some). Issues of knowledge elicitation and representation are inherent in these environments and appropriate machinery is needed. Furthermore, traditional approaches are built on a probabilistic view of uncertainty, where possible actions are evaluated through their expected utility (Dubois and Prade, 1997). The use of such crisp values has been extensively criticized; the specification of the complete sets of probabilities and utilities required renders such approaches impractical for the majority of decision making tasks that involve common sense knowledge and reasoning (Tan and Pearl, 1994). On the other hand, AI approaches basically attempt to reduce the burden of numerical information required, while paying much attention to the automation of the process (Perny and Pomerol, 1999).

Using the model of discourse acts described in the previous section, our ultimate goal is the development of a complete system for supporting GDM in real-life applications. Information technology is essential for achieving proper communication and enabling information access to everybody. The system described in this paper is based on the World-Wide Web platform, while it meets three major practical requirements: (i) it provides relatively inexpensive access; (ii) it has intuitive interfaces in order to be easily usable by inexperienced users, and (iii) it can be available on all prominent operating systems and hardware platforms. In order to use the system, one only needs a Web browser and Internet access. We focus on distributed, asynchronous collaboration, allowing agents to surpass the requirements of being in the same place and working at the same time. The system is intended to act as an assistant and advisor, by facilitating communication and recommending solutions, but leaving the final enforcement of decisions and actions to the agents.

The system consists of four components (Fig. 1). The Argumentation-based Group Decision Support Module enhances decision making by supporting argumentative discourse among decision makers. Note that, depending on the type of the GDM process, the overall system may be ‘administrated’ by a discussion moderator which can intervene when needed (for instance, to verify the truthfulness of an assertion). The module enables decision makers to propose and discuss alternative solutions to a problem by electronically sending various discourse acts to the system’s server (for instance, they can use the consider act to open an issue, the corroborate act to further support an alternative, etc.). Users are then able to access the structured protocol of a discussion, which is based on a formal argumentation theory. The component is able to elicit and represent the domain knowledge; moreover, it efficiently structures and consistently maintains the decision analysis, and provides appropriate mechanisms for automating the GDM process itself. All discourse acts are recorded in the system’s relational database (mSQL has been used). Interactions with the database are
performed through *Java Database Connectivity* (JDBC) drivers. A first version of this component, namely *HERMES*, which supports a subset of the discourse acts described in the previous section, is fully implemented in Java (Karacapilidis and Papadias, 1998a) and has already received positive evaluation feedback.

The *Information Retrieval Module* aims at assisting decision makers to retrieve and reuse information stored in remote databases. It is able to access databases on any platform (e.g., MSAccess on PCs, Oracle on Unix, etc.) through the CORBA communication protocol. The module is also implemented in Java. The *Natural Language Processing (NLP) Module* provides techniques for the analysis of the user messages during a discussion. The idea is to identify the intended discourse acts (according to the model discussed earlier in this paper) and, in the sequel, automatically extract and link the corresponding argumentation items. Finally, the *Argument Builder Module* concerns perusal of a discussion’s patterns, the aim being to advise agents to perform discourse acts that best interpret their interests and intentions (for instance, to support or refute a certain argument).

Apparently, this module is highly associated with the Information Retrieval one. Having retrieved the required information, decision makers may construct more robust arguments to be then submitted in the corresponding GDM discussion. The NLP and Argument Builder modules are currently under implementation.

### 4. Supporting argumentative discourse in group decision making

This section discusses concepts and techniques involved in the Argumentation-based Group Decision Support module, in the context of the HERMES system. As said above, HERMES supports a subset of the discourse acts defined in Section 2 of this paper. However, integration of all such acts is rather straightforward.

The argumentation framework of HERMES is an extension of the ZENO one (Gordon and Karacapilidis, 1997; Karacapilidis and Papadias, 1998b), which in turn has its roots to the informal IBIS model of argumentation (Rittel and Webber, 1973). HERMES supports *issues*, *alternatives*, *positions*, and *constraints representing preference relations* as argumentation

![Fig. 1. The system’s architecture.](image-url)
elements. We consider here a real medical decision making example concerning the appropriate treatment to be followed for a patient case. Three medical doctors participate in the discussion, namely Dr. Brown, Dr. Clark and Dr. Wadder. Fig. 2 illustrates two instances of the corresponding HERMES Discussion Forum. As shown, our approach maps a multi-agent decision making process to a discussion graph with a hierarchical structure.

Issues correspond to the decisions to be made, or the goals to be achieved (e.g., issue-1: “prolactinoma case of patient CD-5687-98; what’s the appropriate treatment?”). They are brought up by agents and are open to dispute. Issues consist of a set of alternatives that correspond to potential choices (e.g., alternative-3: “surgical operation” and alternative-5: “pharmacological treatment” both belong to issue-1, and have been proposed by Dr. Brown and Dr. Clark, respectively). Issues can be inside other issues in cases where some alternatives need to be grouped together. For instance, if two alternative pharmacological treatments were proposed, they could be grouped in an internal issue, say the subissue “which is the most appropriate pharmacological treatment?” According to the related argumentation, the best of these alternatives will then be compared to alternative-3.

Positions are asserted in order to support the selection of a specific course of action (alternative), or avert the agents’ interest from it by expressing some objection. For instance, position-4: “complete removal of tumor” has been asserted to support alternative-3, while position-6: “danger of pituitary insufficiency” to express Dr. Clark’s objection to it. Positions may also refer to some other position in order to provide additional information about it, e.g., position-11: “this is not true in the case of the new medicines we propose” (arguing against position-10), and position-7: “life-long incapability to produce certain hormones” (arguing in favor of position-6). A position always refers to a single other position or alternative, while an alternative is always in an issue.

In decision making environments, one has usually to define priorities among actions and

![Fig. 2. Two instances of a Discussion Forum.](image-url)
weigh different criteria. Unfortunately, well defined utility and probability functions regarding properties or attributes of alternatives (used in traditional approaches), as well as complete ordering of these properties are usually absent. In HERMES, constraints provide a qualitative way to weigh reasons for and against the selection of a certain course of action. A constraint is a tuple of the form \([\text{position}, \text{preference relation}, \text{position}]\), where the preference relation can be more (less) important or of equal importance. Constraints may give various levels of importance to alternatives. Like the other argumentation elements, they are subject to discussion; therefore, they may be ‘linked’ with positions supporting or challenging them (see, for instance, position-13 in the right window). In Fig. 2 (right window), constraint-12: “complete removal of tumor is preferable to taking the risks” expresses the preference relation: “position-4 is more important than position-8”, while constraint-14: “complete removal is more important than the whole treatment’s duration” represents the relation “position-4 is more important than position-9.”

Alternatives, positions and constraints have an activation label indicating their current status (it can be active or inactive). This label is calculated according to the argumentation underneath and the type of evidence specified for them. Activation in our system is a recursive procedure; a change of the activation label of an element (alternative, position or constraint) is propagated upwards. When an alternative is affected during the discussion, the issue it belongs to should be updated since a new choice may be made. In general, different elements of the argumentation, even in the same debate, do not necessarily need the same type of evidence. Quoting the well-used legal domain example, the arguments required to indict someone need not be as convincing as those needed to convict him (Farley and Freeman, 1995). Therefore, a generic argumentation system requires different proof standards (work on AI and Law uses the term ‘burdens of proof’). HERMES uses three different proof standards (we do not claim that the list is exhaustive; other standards, that match specific application needs, can be easily incorporated to the system): (i) Scintilla of Evidence (SoE): according to this proof standard, a position is active, if at least one active position argues in favor of it; (ii) Beyond Reasonable Doubt (BRD): according to BRD, a position is active if there are not any active positions that speak against it; (iii) Preponderance of Evidence (PoE): according to this standard, a position is active when the active positions that support it outweigh those that speak against it.

Without resorting to many details here (see instead Karacapilidis and Papadias, 1998a), each position has a weight = \((\text{max}_\text{weight} + \text{min}_\text{weight})/2\), while an alternative has weight = \(\text{min}_\text{weight}\). \text{Max} _\text{weight} and \text{min}_\text{weight} are initialized to some pre-defined values (initially, let \(\text{min}_\text{weight} = 0\) and \(\text{max}_\text{weight} = 10\); this may be changed by preference constraints). The score of an element \(e_i\) is used to compute its activation label (the score is calculated on-the-fly). If an element does not have any arguments, its score is equal to its weight; otherwise, the score is calculated from the weights of the active positions that refer to it:

\[
\text{score}(e_i) = \sum_{\text{in-favor}(p_j, e_i) \land \text{active}(p_j)} \text{weight}(p_j) - \sum_{\text{against}(p_k, e_i) \land \text{active}(p_k)} \text{weight}(p_k)
\]

In the case of alternatives, PoE produces positive activation label when there are no alternatives with larger score in the same issue. It uses constraints to determine the activation level by comparing the relative importance of supporting and counter arguments. The
associated **weighing schema** and **scoring mechanism** are illustrated through a comprehensive example in (Karacapilidis and Papadias, 1998a); the basic idea is that the weight of a position is increased every time the position is more important than another one (and decreased when is less important), the aim being to extract a total order of alternatives.

In the discussion instance of Fig. 2, the proof standard is SoE for all positions and PoE for alternatives. Position-10 is inactive (the accompanying icons of inactive items are shown in red color), since position-11 is active and speaks against it. On the contrary, position-6 is active (the accompanying icons of active items appear in blue color), since there is at least one active position (position-7) that speaks in favor of it. Active positions are considered 'accepted' due to discussion underneath (e.g., strong supporting arguments, no counter-arguments), while inactive positions are (temporarily) 'rejected'. Similarly, active alternatives correspond to 'recommended' choices, i.e., choices that are the strongest among the alternatives in their issue. Note that both alternatives are actually linked with a position in favor and a position against them (concerning alternative-5, the second position against it, namely position-10, is inactive).

In the left window of the figure, both alternatives receive the same score. The insertion of constraint-12 (right window) renders a higher score to alternative-3, hence this is the solution indicated (at this state) by the system.

The activation label of constraints is decided by two factors: the discussion underneath (similarly to what happens with positions) and the activation label of their consistent positions (i.e., the positions compared through the constraint). In Fig. 2 (right window), both constraints are active. If during the evolution of the discussion, a new position inactivates position-4, this will result in the inactivation of both constraints, since position-4 is one of their consistent positions.

Apart from an activation label, each constraint has a **consistency label** which can be **consistent** or **inconsistent**. Every time a constraint is inserted in the discussion graph, the system checks if both positions of the new constraint exist in another, previously inserted, constraint. If yes, the new constraint is considered either **redundant**, if it also has the same preference relation, or **conflicting**, otherwise. A redundant constraint is ignored, while a conflicting one is grouped together with the previously inserted constraint in an issue automatically created by the system, the rationale being to gather together conflicting constraints and stimulate further argumentation on them until only one becomes active. If both positions of the new constraint do not exist in a previously inserted constraint, its consistency is checked against previous active and consistent constraints belonging to the same issue. In Hermes, this process is handled by a polynomial (O(N^3), where N is the number of the associated positions) **path consistency** algorithm (Mackworth and Freuder, 1985). Although path consistency interacts with the mSQL database where the discussion graph is stored, the algorithm is very efficient. Even for cases involving issues with numerous alternatives and positions linked to them, execution time is negligible compared to communication delay.

The argumentation framework outlined in this section combines concepts from various well-established areas such as Decision Theory, Non-monotonic Reasoning, Constraint Satisfaction, and Truth Maintenance (the abovementioned module’s features are presented in much more detail in (Karacapilidis and Papadias, 1998a, 1998b)). The Web platform facilitates access to the current knowledge by making available all relevant data and documents. When highlighting an argumentation item in the upper pane of the Discussion Forum window, all related
information is given in the lower pane. By clicking on the Url entry, the associated HTML file appears in a new Web browser’s window. In the example shown in Fig. 2, the Url corresponds to a detailed presentation of the effects of the new medicine proposed (position-11 is highlighted). In such a way, participants may ‘attach’ to their discourse elements, useful multimedia information (text, images, sound, etc.).

5. Providing further assistance

This section reports on theoretical and implementation issues concerning the next three modules of the GDM system presented in this paper. While not of minor importance, we view their role as enhancing the Argumentation-based Group Decision Support module; they augment its effectiveness providing decision makers with advanced capabilities. To our knowledge, there is no other GDM system that integrates all these features.

5.1. The Natural Language Processing module

Using only the Argumentation-based Group Decision Support module, the user has to choose the location in the graph where he/she wants to add an argumentation element and the type of the discourse act. That is, the module takes into account only the form of an argument and not its content (this advocates the supervising of such acts by the discussion moderator). The goal of the NLP module is to automate the process of message insertion in the discussion graph using appropriate techniques. For example, in the discussion of Fig. 2, assume that a user wants to assert an argument of the type: “I don’t think we should perform a surgical operation for the patient since he is over-aged and this admittedly involves serious risks; I can give you extra evidence for that which I have collected from our lab’s records.” Such a message cannot be attached to any particular element; so it must be analysed to extract the element(s) that it refers to and the type(s) of the discourse act the user has in mind. Such a process involves three steps: (i) element extraction, which deals with the segmentation of the message; (ii) theme identification, concerning a conceptual association of the extracted elements, and (iii) attitude link-up, which appropriately lays elements and relations on the discussion graph.

The type of methods used here include information extraction technologies and employ segmentation, tagging, and shallow parsing techniques to extract the most salient information from natural language sentences (Wu, 1997). Although accuracy is imperfect, performance is sufficient to achieve the practical goal of speeding up processes that would otherwise require tedious manual labor. Unrestricted natural language understanding is certainly a long way from being solved; however, information extraction methods can work because they depend on strong a priori restrictions on the kinds of patterns they need to search for. In our case, the constraining assumption is that the user’s sentence carries themes, a set of argumentation items, and a set of attitudes. The strength of this restriction makes the task feasible; concerning users of a GDM system, the restriction is not a burdensome one. In fact, the NLP module could relieve them of much of the tedious burden of manually linking each relation; moreover, it could eliminate misinterpretations and mistakes during the assertion of messages.

Without getting into details in this paper, implementation of the module involves integration
of off-the-shelf segmenters, taggers, probabilistic partial parsers, and broad-coverage grammars, and development of additional core NLP technologies in the context of GDM (among others, element extractor, phrasal similarity evaluator, and thesaural phrasal similarity evaluator).

5.2. The Information Retrieval module

In order to reach an understanding and/or negotiate and resolve conflicts through a GDM system, agents may want to consult various types of information which can be useful in further justifying their arguments towards the selection or rejection of a statement/action (Lochbaum et al., 1990; Labrou and Finin, 1994). Such information may be stored either in external, remote databases with legacy data or in the system’s proprietary one. In the GDM domain, case-based reasoning and learning techniques have been particularly useful due to their resemblance to the way people evaluate a potential future action by using past experience, the scarcity (or even absence) of explicitly stated rules, and the ill-structured definitions of the associated problems (Aamodt and Plaza, 1996; Karacapilidis et al., 1997).

The Information Retrieval module allows agents to ask various types of queries regarding the discussion and related material. Each time agents want to add a position in the discussion graph, they are able to retrieve auxiliary information from a pre-specified set of databases. Fig. 3 (left) illustrates the applet window for adding a new position in favor of position-4: "complete removal of tumor." This act may correspond to Dr. Brown's wish to further support position-4 by bringing up information concerning previous cases. Clicking on the ‘Consult Database’ button, a new window applet appears; one can then select the database and which of its tables he/she wants to consult, and then submit a query. Fig. 3 (right) shows such a query to an external medical database. Note that this applet window is dynamically created,
depending on the user’s selection of database and table (this is achieved by exploiting metadata about the structure of the selected database).

Communication with remote databases is achieved through the CORBA (Common Object Request Broker Architecture) protocol, which has become a standard for interoperability in heterogeneous computing environments. The CORBA specification defines a software bus, the Object Request Broker (ORB), that provides the necessary infrastructure to enable cross-platform communication among distributed objects and client programs. Visigenic’s ‘VisiBroker for Java’ environment has been used in our implementation; it is a complete CORBA 2.0 ORB environment for building, deploying and managing distributed Java applications that interoperate across multiple platforms.

5.3. The Argument Builder module

This module aims at assisting agents constructing robust arguments. Since, the overall system cannot be expected to make complete judgement decisions itself, the module must be able to evaluate the arguments being presented by participants, to extract as much information concerning their beliefs and argumentation as possible (also taking into account the whole discussion graph), to provoke the appropriate queries to the related databases, and finally, to point out discourse acts that successfully reflect their interests and intentions. The module interoperates with both the NLP and the Information Retrieval modules and should be viewed as a users ‘advisor’.

The module follows a set of rules, concerning the grammar and syntax of a GDM-related discourse, in order to match patterns of an on-going discussion with pre-specifed ones. For instance, if one’s belief is supported by only one piece of evidence, it will suggest him/her to take appropriate actions to further support it; in cases of conflicts between agents, the module will advise them on actions for conflict resolution (such actions try to achieve a deeper understanding of the issue to be solved). Related work is recently performed in the context of the Belvedere system (Toth et al., 1997).

6. Discussion

Having described the modules which our approach to a collaborative decision support system consists of, this section first outlines related work, the aim being to extract the pros and cons of our contribution. In the sequel, we comment on some interesting aspects of GDM concerning the relationships between the key elements involved in such processes.

6.1. Related work

Various approaches have been proposed and several systems/prototypes have been already developed to address diverse aspects of argumentation modeling and decision analysis in GDM. Starting from Toulmin’s early argumentation theory (Toulmin, 1958), these works have been built on different theories, such as defeasible reasoning (e.g., Pollock, 1988; Geffner and Pearl, 1992), formal disputation (e.g., Rescher, 1977; Brewka, 1994), multi-criteria decision
making and multi-attribute utility theory (e.g., Bui and Jarke, 1986; Jarke et al., 1987; Bui, 1993; Sycara, 1987; Stewart and Van den Honert, 1988), knowledge management and organizational memory (e.g., Conklin, 1992; Bui et al., 1998) and negotiation models (e.g., Kersten and Szapiro, 1986; Jelassi and Foroughi, 1989). A detailed comparison of the above works against the system and its underlying argumentation framework proposed in this paper is given in (Karacapilidis and Pappis, 1997) and (Karacapilidis et al., 1998). It should be noted, however, that this comparison only concerns the argumentation-based group decision support module of our system (see Fig. 1). Functionalities offered in the information retrieval, argument builder and NLP modules, described in the previous section, do not exist in the approaches and systems mentioned above (with the exception of Sycara’s PERSUADER which keeps track of previous cases, thus increasing the efficiency of the system by reusing previous instances of similar argumentation).

Tools supporting collaborative argumentation also rely on different ‘languages’, in that they provide the means to express concepts of different degrees of precision and explicitness when manipulating and experimenting with the existing knowledge. In addition, various representation and support technologies have been proposed. The two most successful systems up to date are probably QuestMap (Conklin and Yourdon, 1993), by Soft Bicycle, and Decision Explorer (Eden and Ackermann, 1998), now provided by Banxia Software. The key component of QuestMap is the use of a hypermedia display system (much like an on-line ‘whiteboard’) that captures the key issues and ideas during meetings and creates shared understanding in a knowledge team. All the messages and artifacts (documents and reference material) for a project can be placed on the ‘whiteboard’, and the relationships between them can be graphically displayed. Users end up with a ‘map’ that shows a history of the on-line conversations that led up to key decisions and plan. On the other hand, Decision Explorer is a tool for managing the qualitative information existing in a complex setting. By building ‘cognitive maps’, users may capture in detail thoughts and ideas, and explore them aiming at gaining a better understanding. Links between ideas add structure to the model, while they reveal inter-relationships (e.g., contradictory and opposing views) and the cause and effect of different elements.

Compared to the above two systems, our approach focuses on aiding decision makers reach a decision, not only by efficiently structuring the discussion, but also by providing reasoning, consistency checking, and ‘weighing’ mechanisms for it. Also, it is fully Web-based, thus providing easy access to the parties involved, and supports distributed (both synchronous and asynchronous) argumentation. Moreover, our approach enables users to retrieve data stored in remote databases in order to further justify their arguments (information retrieval module), and guides them to perform acts that best reflect their interests and intensions (Argument Builder module).

As a last point, we highlight the centrality of communication in our approach. Other decision making software tools have only partially addressed this issue. For instance, using Team Expert Choice (by Expert Choice., see http://www.expertchoice.com), participants need to use hand-held radio keypads to vote, compare and prioritize the relative importance of the decision variables. On the contrary, our system’s client-server and web-based architecture enables users communicating just by using their favorite web browser. Ergo (a bayesian network editor and solver by Noetic Systems, see http://www.noeticsystems.com/ergo.shtml)
allows users to interface directly with popular relational databases, while the reports provided can be exported to HTML web pages, spreadsheets, presentation and word processor formats. However, such databases have to be ‘in-house’ and running on the same platform, not remote ones and platform-independent as allowed in our framework. In addition, as discussed earlier in this paper, our system provides direct links to web pages, where all related information about decision making elements can be placed.

6.2. Argumentation and GDM

Generally speaking, approaches to decision making may be divided into two large classes. In the first one, a set of alternatives is determined a priori and the task is to select one of them. In the second class, an ideal case is decided upon first, and a subsequent task is to find a real case that best approximates the ideal one. In both approaches, however, there are a number of common elements: (i) an overall task goal is specified; (ii) a set of alternatives is selected (this set may not be exhaustive); (iii) a collection of choice criteria must be determined by the participants; (iv) a decision function must be composed which combines criteria to decide between alternatives.

The overall task goal is generally not a subject of debate, although it may be ill-defined and the decision process may involve sub-processes to clarify the goal. The sub-processes themselves may be considered to be GDM processes. In fact, each element of the decision process may itself be the subject of a sub-decision process. Decision making can, therefore, be recursive.

The set of alternatives may be a predetermined, closed set (no further alternatives can later be considered), a predetermined open set (leeway is given to allow integration of new alternatives), or a postdetermined set (in the case of finding a match to an ideal case). Interesting conclusions concerning the implicit goals, a priori positions, and biases of the participants in the process may often be inferred from the manner in which they present alternatives. It is often the case that participants have applied unspecified choice criteria before proposing alternatives (eliminating what they consider to be useless alternatives). In cases where the participants are of unequal stature in the GDM process (as, for instance, when a mixture of middle and upper management are involved) this can have a profound implicit effect on the collaborative aspects of the decision process, and modeling of the hierarchical relations between participants may be necessary to understand what may appear to be illogical or contradictory decisions.

The choice criteria are the basis of any decision process. They provide the metrics upon which alternatives are compared, and accepted or rejected. As the foundations of the GDM process, they may be the subject of much debate. The inclusion of particular criteria may cause one to consider alternatives that would otherwise not figure in the process, while the exclusion of certain criteria may automatically eliminate certain alternatives that would prima facie be included. They can, therefore, be a preliminary battleground for power struggles between factions involved in the process. The decision function, however, is where most argumentation is centred, since it is here that the relative value of choice criteria is established and applied to select between the alternatives. The argumentation used is often authoritative or based on voting.
The multitude of goals and arguments, which are often expressed in qualitative or ill-structured forms, is inherent in multicriteria decision making environments, and advocates the use of approximation models. Fuzzy sets theory (Zadeh, 1965) provides a conceptual framework that may prove to be useful for dealing with situations characterized by imprecision due to subjective and qualitative evaluations (Zimmermann, 1991; Delgado et al., 1992; Roubens, 1997). The argumentation-based GDM module described in this paper can also be applied when using an approximation model to represent the related objectives and criteria. As shown in (Karacapilidis and Pappis, 2000), one can first elicit the decision makers’ knowledge aiming at specifying the ideal (desired) solution to the problem, and then apply similarity measures (Pappis and Karacapilidis, 1993, 1995) to decide which of the existing solutions is closer to the ideal one.

Finally, an important issue in decision making is the assessment of the degree with which a particular alternative satisfies a specific criterion. One way of determining this degree of satisfaction is based on the Analytic Hierarchy Process (AHP; see Saaty, 1980), where such information is derived from pairwise comparison of alternatives on a ratio scale. AHP has been applied successfully in many fields, while several multi-criteria decision support software tools have been based on it (the market-leader probably being Expert Choice, again by Expert Choice). Yet, there has been some criticism of AHP’s decision method and many modifications/improvements have been suggested (Saaty, 1989; Ved den Honert and Lootsma, 1995) leading to long discussions about its merits (see, for instance, Petkov and Mihova-Petkova, 1998; Edgar-Nevill and Jackson, 1998). So far, there does not seem to be a general consensus about the way the theory is applied or about the methods that are used in particular stages of AHP. The advantages of the AHP (and of tools such as Expert Choice) include the use of a flexible modeling and measurement approach to evaluation; the application of structure to facilitate decision making through the use of a model which imposes strict independence, ordinality, or homogeneity of preferences, thus allowing the decision maker to arrive at consistent, objective evaluations; and the simplicity of the use of the model (especially when performing what-if and sensitivity analysis). Limitations arise due to the imposed structure of decision making. For example, a rational decision is assumed to be derived from a linear, additive, and compensatory model structure whereas human preferences are often the result of irrational comparisons (Raiffa, 1970). The model also does not consider judgmental heuristics and biases. The approach proposed in this paper addresses such limitations, in that it allows for a very flexible decision making structure; the process followed is argumentation-driven, thus it may represent various forms of human reasoning.

7. Conclusion

It is our belief that the integration of the modules discussed in this paper with a properly specified model of discourse acts is the most reasonable way to cross the divide between, on the one hand, the simple systems for remote GDM that provide no more than a communication channel and archiving facility between participants (e.g., newsgroups and web forums) and, on the other hand, fully-blown reasoning systems that attempt to automatically solve the decision making problem. This latter approach appears to us to be infeasible at the
moment. However, the mix of human and machine reasoning that forms the basis of our proposition appears to be feasible. We view human intervention as necessary to determine certain instances of the propositions being asserted, and relations holding between them. Accepting that the process of interpreting dialogue acts from utterances is fallible and non-exclusive, the objective of our approach is to establish the most likely and coherent set of dialogue acts associated with each intervention by a participant, and to maintain the set of hypotheses being supported or considered by the participants. The participants can use this to better understand the arguments of their fellow decision makers, and advance more pointed argumentation of their own.

Acknowledgements

The author thanks Alexis Tsoukias and the anonymous referees for their helpful comments and suggestions towards improving this paper.

References


