

COMPUTER SUPPORTED ARGUMENTATION AND COLLABORATIVE DECISION MAKING: THE HERMES SYSTEM

NIKOS KARACAPILIDIS¹ and DIMITRIS PAPADIAS²

¹ Industrial Management Laboratory, University of Patras, 26500 Rion-Patras, Greece

² Department of Computer Science, Hong Kong University of Science and Technology, Clearwater Bay, Hong Kong

Abstract - Collaborative Decision Making problems can be addressed through argumentative discourse and collaboration among the users involved. Consensus is achieved through the process of collaboratively considering alternative understandings of the problem, competing interests, priorities and constraints. The application of formal modeling and analysis tools to solve the related processes is impossible before the problem can be articulated in a concise and agreed upon manner. This paper describes HERMES, a system that augments classical decision making approaches by supporting argumentative discourse among decision makers. It is fully implemented in Java and runs on the Web, thus providing relatively inexpensive access to a broad public. Using an illustrative example, we present the argumentation elements, discourse acts and reasoning mechanisms involved in HERMES. We also describe the integration of advanced features to the system; these enable users to retrieve data stored in remote databases in order to further warrant their arguments, and stimulate them to perform acts that best reflect their interests and intentions.

Key words: CSCW, Argumentation, Collaborative Decision Making, Information Sharing and Retrieval, World Wide Web

1. INTRODUCTION

Collaborative Decision Support Systems (CDSSs) can be 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 CDSS 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 removing communication impediments, and providing techniques for structuring the decision analysis and systematically directing the pattern, timing, or content of the related discussions. Major issues arising during the development of such a system are the effective work organization, in order to improve coordination, and the use of communication technology, to make decision making more efficient. Provision of rules and procedures for achieving consistency and automation of data processing, especially in data intensive decision making situations, are also of high importance.

Usually, Collaborative Decision Making (CDM) is performed through debates and negotiations among a group of people. 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 specific level. Opinions may differ about the relevance or value of a position when deciding an issue. Decision makers may have arguments supporting or against alternative solutions. In addition, they have to confront the existence of

insufficient or too much information. In other words, for some parts of the problem, relevant information that 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, factual knowledge is not always sufficient for making a decision. Value judgments, depending on the role and the goals of each decision maker, are among the critical issues requiring attention. A CDSS should provide users with appropriate means to assert their preferences, which often are expressed in qualitative terms. Finally, decision makers are not necessarily proficient in computer science and information technology. They need appropriate tools in order to easily follow and participate in the associated processes. Such tools should stimulate their participation giving them an *active* role. This is in accordance with the vision of the DSS community's pioneers: by supporting and not replacing human judgment, the system comes in second and the users first.

This paper describes HERMES, a system developed to address the above issues. HERMES aims at augmenting classical decision making approaches by supporting argumentative discourse among decision makers [Karacapilidis and Papadias, 1998a]. It is implemented in Java (applets) and runs on the Web, using any standard browser. The system can be used for distributed, asynchronous collaboration, allowing users to surpass the requirements of being in the same place and working at the same time.

The remainder of the paper is structured as follows: The next section outlines related research work and comments on the role of argumentation in CDM. Sections 3 and 4 describe the *argumentation elements* and the features of the system, respectively, through an illustrative example. Representative *discourse acts* are presented in Section 5, while tools that provide additional functionality to the system are discussed in Section 6. Finally, evaluation results and concluding remarks are given in Sections 7 and 8, respectively.

2. ARGUMENTATION AND COLLABORATIVE DECISION MAKING

CDM admittedly falls in the category of *wicked* problems [Kunz and Rittel, 1970; Rittel and Webber, 1973], a class of problems that can be addressed through argumentative discourse and collaboration among the users involved. In such a case, consensus emerges through the process of collaboratively considering alternative understandings of the problem, competing interests, priorities and constraints (see, for instance, [De Michelis and Grasso, 1994] and [Di Eugenio et al., 1997]). The application of more formal modeling and analysis tools is impossible before the problem can be articulated in a concise and agreed upon manner.

Research on argumentative discourse has been performed through various viewpoints such as *philosophical*, where argumentation is considered as a way of theory formation [Lueken, 1991], *linguistic*, inspired by speech act theory and discourse analysis [Heidrich, 1982], *rhetorical*, where argumentation is studied primarily as a form of manipulation through language [Valesio, 1980], and *psychological*, where the relationships between arguments and cognitive processes are explored [Green, 1996]. As comprehensively described in [van Eemeren et al., 1996], most approaches follow two main methodologies, namely, formal and informal logic. According to the *formal perspective*, arguments are de-contextualized sets of sentences or symbols viewed in terms of their syntactic or semantic relationships. For instance, Brewka [1994] reconstructed Rescher's theory of formal disputation [Rescher,

1977], while Gordon's work [Gordon, 1995], was based on Geffner and Pearl's concepts of conditional entailment [Geffner and Pearl, 1992]¹. On the other hand, *informal logic* views arguments as pragmatic, i.e., their meaning is a function of their purposive context.

In addition to argumentation theories, implementation of argumentation and decision rationale support systems for different types of groups and application areas has been receiving growing interest in the last few years. 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 are placed on the “whiteboard”, and the relationships between them are graphically displayed. Users end up with a “map” that shows the history of an on-line conversation that led to key decisions and plans. QuestMap has been based on the *gIBIS* hypertext groupware tool [Conklin and Begeman, 1987], which aimed at capturing the rationale of a design process. *Euclid* [Smolensky et al., 1987] is another system of this category, which provides a graphical representation language for generic argumentation. Moreover, *JANUS* [Fischer et al., 1989] is based on acts of critiquing existing knowledge in order to foster the understanding of design knowledge. *SEPIA* [Streitz et al., 1989] is a knowledge-based authoring and idea processing tool for creating and revising hyper-documents that views authoring as a design process. *QOC* (Questions, Options and Criteria) is another interesting model to represent the rationale of reasoning in a decision making process [Shum et al., 1993; MacLean et al., 1991], in that it provides users the means to represent and integrate rationale of varying degrees of stability, at different stages in a design process. Finally, *Belvedere* [Suthers et al., 1995], originally designed to support students engaged in critical discussion of science issues, uses a rich graphical language to represent different logical and rhetorical relations within a debate. Although this category of systems provides a cognitive argumentation environment that stimulates discussion among participants, it lacks decision making capabilities.

In the same context, REMAP [Ramesh, 1992] is a conceptual model used to capture deliberations during requirements analysis, which relates process knowledge to the objects that are created during the requirements engineering process. It embodies the IBIS model of argumentation [Kunz and Rittel, 1970; Rittel and Webber, 1973] and is supported by a system based on the Telos language [Mylopoulos et al., 1990]. REMAP is useful in capturing deliberations and knowledge when transforming requirements into design and enhances previous approaches by adding explicit representation of decisions, constraints and assumptions, and by introducing knowledge process components. However, the model is not generic, in that it cannot efficiently capture and represent process knowledge across various phases of systems development. Interesting automated rationale based techniques in requirements engineering have been also proposed by Robinson and his colleagues [Robinson and Fickas, 1994; Robinson and Volkov, 1997]. Their *Oz* prototype automates the generation of resolutions in conflicts arising during the integration of multi-user requirements. Using planning techniques (such as back-propagation), Oz can detect conflicts and generate compromise resolutions through a requirements restructuring model. Once again, it is our belief that their approach is not generic enough to address alternative decision making environments; as also admitted in [Robinson and Volkov, 1997], requirements need always to be formally represented, and there are open issues concerning coverage of the restructuring methods and their controlling heuristics.

¹ A discussion on the use of alternative logics in argumentative discourse can be found in [Prakken, 1995].

SIBYL [Lee, 1990] is another tool for managing group decision rationale, which has also been viewed as an extension of *gIBIS*; while the latter is mainly a hypertext system whose services focus on the representation of a discourse structure (with the purpose to make it easy for people to see the discourse's rationale), *SIBYL* is rather a knowledge-based system that provides services for the management of dependency, uncertainty, viewpoints and precedents. *SIBYL* implements DRL (Decision Representation Language) that supports among others goals, alternatives, claims and relations among them. In addition, the tool makes people articulate and become aware of the objectives against which alternatives are evaluated, thus targeting to an explicit representation of goals. However, decision making in *SIBYL* is only performed through simple decision matrices, where users evaluate alternatives with respect to the goals specified; arguments in favor or against an alternative are to be considered by the users before they submit their evaluation.

Increasing interest also develops in implementing Web-based conferencing systems, such as *AltaVista Forum Center*, *Open Meeting*, *NetForum* and *UK Web's Focus*, to mention some. Such systems exploit the platform-independent communication framework of the Web, as well as its associated facilities for data representation, transmission and access. 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*; this 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].

Contrary to the above systems, our approach focuses on aiding decision makers reach a decision, not only by efficiently structuring the discussion, but also by providing reasoning mechanisms for it. Our primary goal in HERMES was to develop a generic *active* system that efficiently captures users' rationale, stimulates knowledge elicitation and argumentation on the issues under consideration, while it constantly (and automatically) checks for inconsistencies among users preferences and considers the whole set of the argumentation items asserted to update the discourse status. In other words (following Conklin's terminology), the system not only "captures the informal organizational memory" embodied in such environments [Conklin, 1992], but also helps the users during the decision making process *per se* by integrating features based on concepts from well-established areas such as Decision Theory, Non-Monotonic Reasoning, Constraint Satisfaction, Truth Maintenance and Information Retrieval. As comprehensively described in the next sections, the reasoning mechanisms of the system can efficiently handle qualitative data and are automatically triggered each time a user inserts a new item in the discourse graph (insertion of a new item may change the status of numerous existing items and make another alternative look as more promising). As a last note, it should be noted here that HERMES 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 users [Karacapilidis and Papadias, 1998b].

3. ARGUMENTATION ELEMENTS

The argumentation framework of HERMES is an extension of ZENO [Gordon and Karacapilidis, 1997], which in turn has its roots to the informal IBIS model of argumentation [Kunz and Rittel, 1970; Rittel and Webber, 1973]. The basic argumentation elements in HERMES are *issues*, *alternatives*, *positions*, and *constraints* representing *preference relations*².

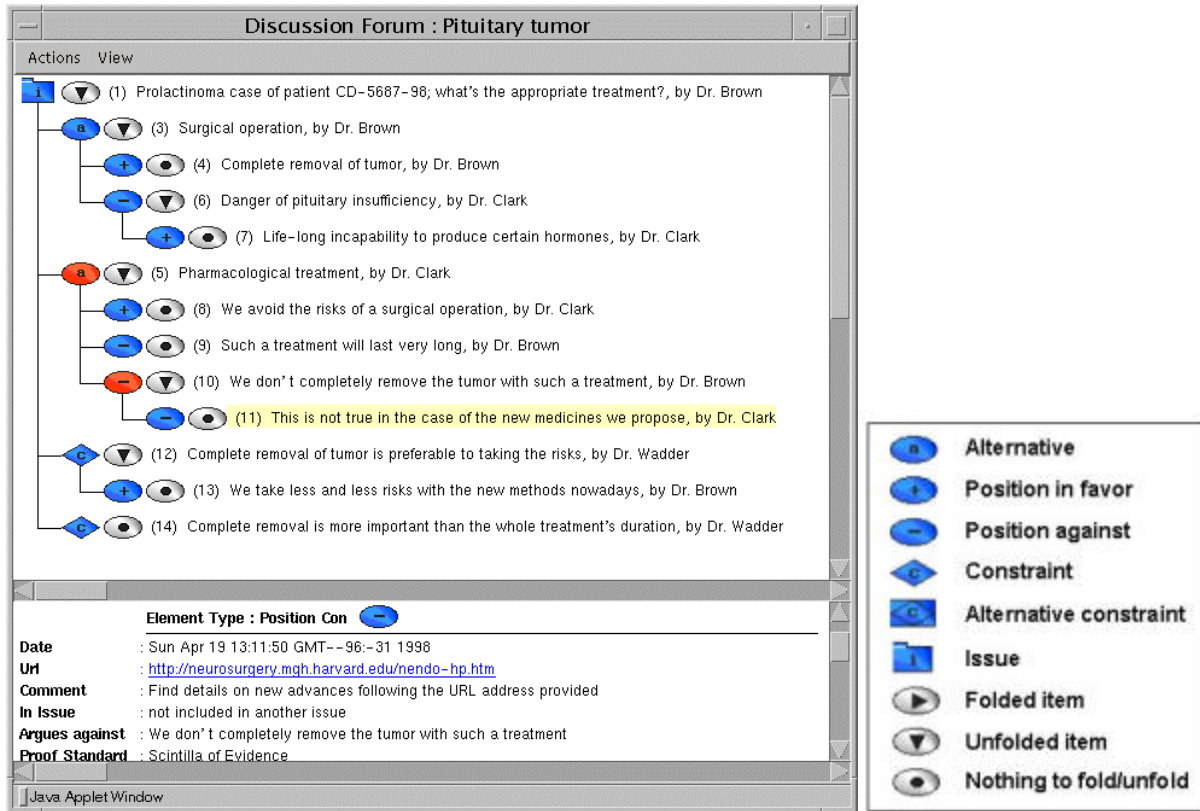


Fig. 1: An instance of the HERMES Discussion Forum (left) and an explanation of the full notation used in the system (right)

Throughout this paper, we use a real example from the medical decision making domain about the appropriate treatment for a patient case. Three medical doctors participate in the discussion, namely Dr. Brown, Dr. Clark and Dr. Wadder; they bring up the necessary argumentation in order to express their interests and perspectives. Figure 1 illustrates an instance 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 (upper pane). Each entry in the Forum corresponds to an argumentation element. In the applet window of Figure 1, each element is accompanied by two icons, one that indicates the element type and another serving for folding/unfolding purposes. Moreover, the numbers shown in parentheses (next to the icons) correspond to a unique ID of the element in the database (IDs are automatically assigned). Finally, each entry in the Forum may contain the username of the user who submitted it and the date of submission (alternative forms of each entry can be obtained through options provided under the View menu). The lower pane of the applet window provides all necessary information about a selected entry of the discussion graph (users can select an entry by clicking on it; for instance, position-11 has been selected in Figure 1).

Issues correspond to decisions to be made, or 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 users and are open to dispute (the root entity of a Discussion Forum has always to be an issue). 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).

An issue can be also “inside” another issue, in cases where some alternatives need to be grouped together. For instance, assume that there were two alternative pharmacological treatments, say an ordinary and an innovative one. Figure 2 shows the structure of the Discussion Forum in such a case. Alternative-5: “Ordinary pharmacological treatment” and alternative-7: “Innovative pharmacological treatment” have been put together in an internal *subissue* (issue-6: “Selection of the appropriate pharmacological treatment”). During the discourse, the selected (best) alternative of issue-6 is compared to alternative-3.

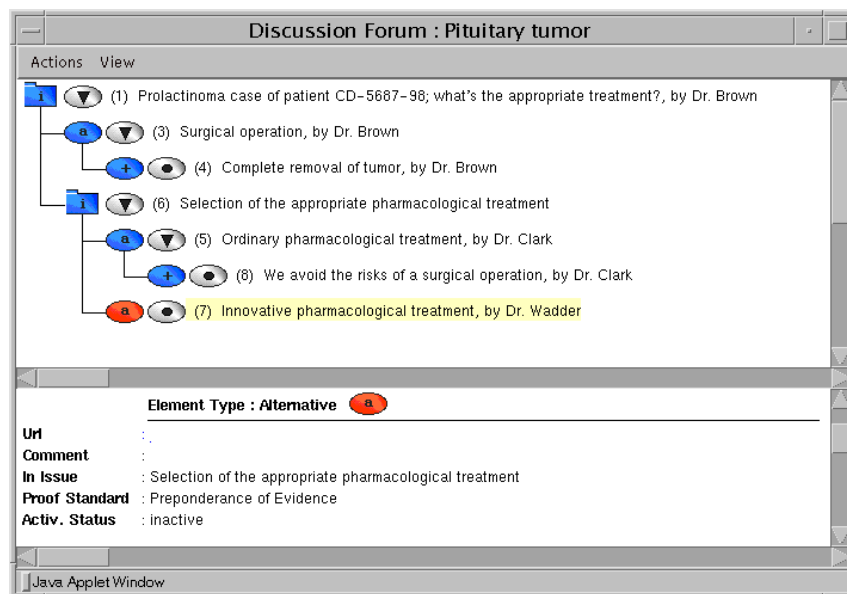


Fig. 2: Subissues

Positions are asserted in order to support the selection of a specific course of action (alternative), or avert the users’ interest from it by expressing some objection. For instance, in Figure 1, position-4: “Complete removal of tumor” has been asserted to support alternative-3, while position-9: “Such a treatment will last very long” to express Dr. Brown’s objection to alternative-5. 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 *refers to* a single other position or alternative, while an alternative is always *in* a single issue.

² The IBIS model has as elements *issues*, *positions* and *arguments*; Conklin’s variant uses the same set of elements under the terms *questions*, *ideas*, *pros* and *cons* [Conklin, 1996].

In decision making environments, one has often to define priorities among actions and weigh different criteria. Well-defined utility and probability functions regarding properties or attributes of alternatives (used in traditional approaches, see for instance [Saaty, 1980]), as well as complete ordering of these properties are usually absent. Furthermore, it has been extensively argued that such approaches are impractical for the majority of decision making tasks that involve common sense knowledge and reasoning [Tan and Pearl, 1994]. In argumentation studies, subjects like *priority relationships* and *preference orders* between arguments have been mostly handled through quantitative approaches [Simari and Loui, 1992; Pinkas, 1991].

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 [position, preference_relation, position], where the preference relation can be *more (less) important than* or *of equal importance to*. Constraints refer to an issue (or an internal subissue) and 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. In Figure 1, constraint-12: “Complete removal of tumor is preferable to taking the risks” expresses the preference relation “position-4 is more important than position-8”, refers to issue-1 and is supported by position-13.

More specifically, two types of preferences can be expressed by the system:

- *Local*, when a constraint refers to a position, or another constraint. In this case, the positions that constitute the constraint must refer to the same element (i.e., have the same father). In the example shown in Figure 1, a preference of this type could be expressed by the constraint “position-7 is less important than position-99”, assuming that position-99 has been inserted and refers to position-6. Note that a constraint of the form “position-9 is less important than position-11” is not permitted, since the positions compared through the constraint (called *constituent positions* hereafter) do not refer to the same father element.
- *Non-local*, when a constraint refers to an issue. In this case, its constituent positions must refer to alternatives (not necessarily the same one) of this very issue (both constraint-12 and constraint-14 fall in this category).

4. SYSTEM FEATURES

This section describes the reasoning mechanisms integrated in the system. Without resorting to formal approaches (such as qualitative probabilistic knowledge [Goldszmidt and Pearl, 1996] and logic of argumentation [Fox and Parsons, 1997]), such mechanisms determine the appropriate labeling for each entry of the discussion graph, the aim being to keep users aware of the discourse status. As highlighted in the first two sections of the paper, the features illustrated here provide the basis for an efficient interworking of multi-agent argumentation and decision making.

4.1. Proof Standards

Alternatives, positions and constraints have an *activation label* indicating their current status (they can be *active* or *inactive*). This label is calculated according to the argumentation underneath and the *type of evidence* specified for them. Activation in HERMES is a recursive procedure; a change of the activation label of an element (alternative, position or constraint) is propagated upwards in the discussion graph. For instance, 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*). In the sequel, we describe the ones implemented in HERMES (the names have the same interpretation as in the legal domain). We do not claim that the list is exhaustive; however, other standards, which match specific application needs, can be easily incorporated to the system.

- **Scintilla of Evidence** (SoE): according to this proof standard, a position p_i is active, if at least one active position argues in favor of it: $active(p_i) \hat{=} \bigvee_{p_j} active(p_j) \hat{=} \bigvee_{p_j} in_favor(p_j, p_i)$.
- **Beyond Reasonable Doubt** (BRD): according to BRD, a position p_i is active if there are not any active positions that speak against it: $active(p_i) \hat{=} \neg \bigvee_{p_j} active(p_j) \hat{=} \neg \bigvee_{p_j} against(p_j, p_i)$.

In the case of the third proof standard, each position has a $weight=(max_weight + min_weight)/2$, while an alternative with no positions linked underneath has a $weight=min_weight$. Both max_weight and min_weight are initialized to some predefined values (in the following, we assume that initially $min_weight=0$ and $max_weight=10$; this may be changed by preference constraints). The *score* of an argumentation element e_i is used to compute its *activation label*. 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:

$$score(e_i) = \sum_{\substack{in_favor(p_j, e_i) \wedge \\ active(p_j)}} weight(p_j) - \sum_{\substack{against(p_k, e_i) \wedge \\ active(p_k)}} weight(p_k)$$

- **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: $active(p_i) \hat{=} score(p_i) \geq 0$. Concerning alternatives, PoE will produce positive activation label for an alternative a_i when there are no alternatives with larger score in the same issue: $active(a_i) \hat{=} \neg \bigvee_{a_j} in_issue(a_i, a_j) \wedge score(a_j) > score(a_i)$. Note that, according to the above, more than one alternative can be active at a discussion instance (e.g., consider a case where three alternatives a_1 , a_2 and a_3 have scores 8, 5 and 8, respectively; a_1 and a_3 are both active, while a_2 is inactive).

In the discussion instance of Figure 1, it has been assumed that the proof standard is SoE for all positions and PoE for alternatives (for more on the selection of a proof standard, see Section 5). Position-10 is inactive (in the system's interface windows, the accompanying icons of inactive items appear in red color) since position-11 is active and speaks against it (note that, according to the definition of SoE, position-10 needs an active position that will argue

in favor of it in order to become active again). On the contrary, position-6 is active (in the system's interface windows, 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 (similarly to the above, position-6 will become inactive if a new position will argue against position-7). Active positions are considered "accepted" due to discussion underneath (e.g., strong supporting arguments, no counter-arguments), while inactive positions are (temporarily) considered as "discarded" or "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 (at the discussion instance shown in Figure 1) linked with a position in favor and a position against them (concerning alternative-5, the second position against it, namely position-10, is inactive). If there were not any constraints, the two alternatives would receive the same score. The insertion of constraint-12 and constraint-14 renders a higher score to alternative-3 (calculation of scores is described in detail in Section 4.3), hence this is the solution currently indicated 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 constituent positions. In Figure 1, both constraints have BRD as proof standard; therefore, they are currently 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 constituent positions (assuming that nothing else related to them changes). In such a case, the two alternatives will receive again the same score.

4.2. Detecting conflicts and inconsistencies

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 constituent 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. It should be noted here that both constituent positions of a new constraint have been already inserted in the discussion graph; thus, whenever a user is about to insert a new constraint, the system provides him/her with a list of all possible combinations to select from (see Figure 7 and the related discussion in the next section).

To illustrate the above, recall that in the discussion instance shown in Figure 1, issue-1 contains two constraints, namely constraint-12 and constraint-14. The latter has been asserted by Dr. Wadder and expresses the preference relation "position-4 is more important than position-9". Assume now that Dr. Clark has another opinion on this relation believing that "position-4 is less important than position-9". Figure 3 depicts the Discussion Forum applet window after the submission of the corresponding constraint by Dr. Clark (constraint-16: "Complete removal is less important than the whole treatment's duration"). Note that the icons of the two associated constraints have been changed now (these icons correspond to what we call *alternative constraints*). A new issue (issue-15) has been also

(automatically) brought up, which consists of these two constraints. The subject of such an issue is standard, namely “issue contains alternative (inconsistent) constraints”. Alternative constraints are obviously open to discussion; users may submit appropriate argumentation items to speak in favor or against them (actually, constraints are now handled as proper alternatives; in other words, the system calculates their scores taking into account the weights of the active positions that argue in favor or against them). In Figure 3, the insertion of position-17: “this isn't true in the case of an aged person” has resulted to the inactivation of constraint-14 (since constraint-16 has a larger score); therefore, issue-15 is “resolved”, by proposing constraint-16 (issue resolution is performed through the weighting mechanism, which is comprehensively described in the next section). In turn, constraint-12 and constraint-16 are now the ones taken into account for the resolution (selection of the best alternative) of issue-1.

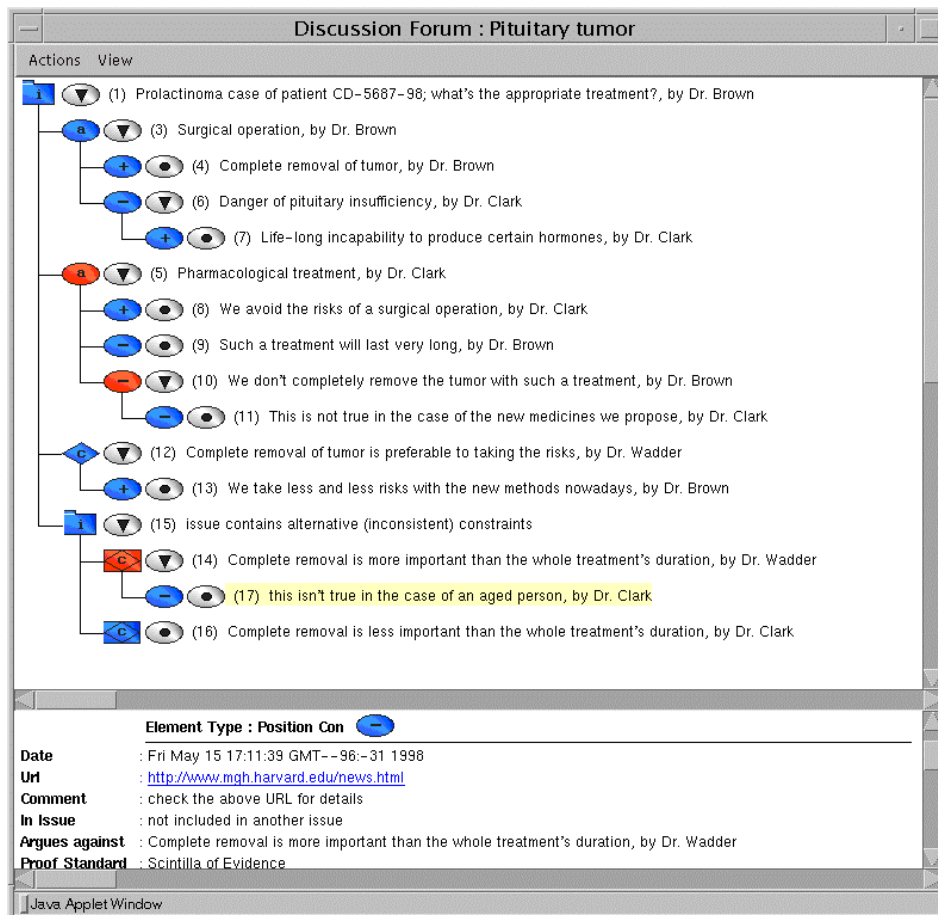


Fig. 3: Detecting inconsistent constraints

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 referring to the same element (or belonging to the same issue). Consider for instance a situation, where there exist two constraints “position-x is more important than position-y” and “position-y is more important than position-z”. A new constraint “position-z is more important than position-x” would be inconsistent with respect to the first two ones, although it is not directly conflicting with either one. Inconsistency checking is performed through a polynomial ($O(N^3)$, N the number of the associated positions) *path consistency* algorithm [Mackworth and Freuder, 1985]. Although path consistency, as most discourse acts described in the sequel, interacts with the database where the discussion graph is stored (the “public-domain” *mSQL* is used),

the algorithm is efficient; even for non-local preferences involving issues with numerous alternatives and positions linked to them, execution time is negligible compared to communication delay.

4.3. The weighting mechanism

Active and consistent constraints participate in the weighting mechanism (only constraint-12 and constraint-16 in the example of Figure 3). In order to demonstrate how the algorithm for altering weights works, we use the example of Figure 4. There exist five positions and four constraints that relate them as illustrated in Figure 4a. The arrowed lines correspond to the “more important than” ($>$) relation (e.g., $p_1 > p_2$) and the dotted line to the “equally important to” ($=$) relation (e.g., $p_3 = p_4$). Initially (with no constraints inserted in the system), each position would have a $weight = (max_weight + min_weight)/2 = 5$ (as stated in Section 4.1, we have assumed that $max_weight=10$ and $min_weight=0$). *Topological sort* [Knuth, 1973] is applied twice to compute the possible maximum and minimum weights for each position (Figure 4b). The new weight of each position (having taken all constraints into account) is the average of its new max_weight and min_weight . More specifically, it is: $weight(p_1)=6$, $weight(p_2)=4.5$, $weight(p_3)=5$, $weight(p_4)=5$ and $weight(p_5)=4$.

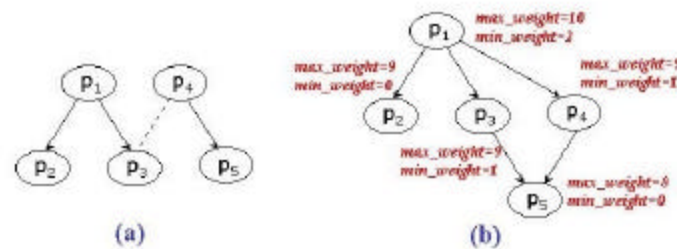


Fig. 4: The weighting mechanism

For instance, the scores of alternative-3 and alternative-5 in Figure 3 are 0 and -2, respectively, since position-4, position-6, position-8, and position-9 have scores 5, 5, 4 and 6, respectively (position-10 is inactive). Therefore, alternative-3 is active and recommended by the system. If position-9 were also inactive (assuming that nothing else had changed), the scores of the above two alternatives would be 0 and 4. In this case, alternative-5 would be the recommended one.

The basic idea behind the above mechanism is that the weight of a position is increased every time the position is more important than another one (and decreased when it is less important), the aim being to extract a total order of alternatives. Since only partial information may be given, the choice of the initial maximum and minimum weights may affect the system’s recommendation. Furthermore, this weighting scheme is not the only solution; alternative schemes, based on different algorithms, are under implementation.

5. DISCOURSE ACTS

Argumentation in our framework is performed through a variety of *discourse acts*. These acts may have different functions and roles in the argumentative discourse. We classify them in two major categories: *agent acts* and *internal (system) acts*.

5.1 Agent acts

Agent acts concern user actions and correspond to functions directly supported by the user interface. Such functions include the opening of an issue, insertion of a new alternative to an issue, addition of a new alternative to an existing alternative, insertion of a new position in favor or against an existing position/constraint/alternative, and insertion of a new constraint to an existing issue. There is also another agent act, which is not immediately connected to the Discussion Forum, that allows users to ask the system's database queries of the form "give me all messages by a *specific user* on an issue" or "what was the recommended solution on a *given date*". In the sequel, we present the pseudo-code for a few representative agent acts:

```
add_Alternative_to_Issue (alternative alt, issue iss) {  
    In(alt) = iss;          /* alt is attached in iss */  
    update(iss);          }  
}
```

```
add_Alternative_to_Alternative (alternative alti, alternative altj) {  
    issj = In(altj);  
    create new issue issi;  
    In(issi) = issj;  
    In(alti) = In(altj) = issi;  
    update(issi);      }  
}
```

The screenshot shows a Java Applet window titled "Alternative". Inside the window, there is a blue button labeled "Add a new Alternative". Below the button, there are instructions: "The new alternative will be included in the issue below." and "Add information about the new alternative below." The form contains four input fields: "In Issue" with the text "Prolactinoma case of patient CD-5687-98; what's the appropriate", "Subject" with "Surgical operation", "URL" with "http://www.unil.ch/med/surgical/caseCD5687.html", and "Comments" with "Check the URL address to get a full description of my proposa". At the bottom of the window, there are three buttons: "Submit", "Clear", and "Cancel".

Fig. 5: Adding a new alternative

The applet window for adding a new alternative to an existing issue is shown in Figure 5. When an alternative alt_i is added to another alt_j (and not directly to the issue iss_j where alt_j belongs), a new issue iss_i is automatically created inside iss_j . Both alt_i and alt_j are now put inside the new issue and compared through $update(iss_i)$. $update(iss_i)$ will be called from $update(iss_i)$ and the recommended choice between alt_i and alt_j will be compared against the other alternatives of the external (initial) issue. Note that in Figure 5, users can give a *subject* (title) of the new alternative, but also provide more details about their entry through the URL pane. More specifically, they can fill in a web address that corresponds to a file with multimedia content. In such a way, they actually "attach" to their elements larger pieces of text, images, figures, sound, etc., thus providing the extra information needed.

Fig. 6: Adding a new position

The applet window for adding a new position is shown in Figure 6. The *father* element can be an alternative, another position, or a constraint. In addition to the “Add a new alternative” applet window, users have to specify here the *type of link* (*argues for* or *argues against*) and the *proof standard* they prefer (depending on the discussion context, this option may be inactivated; that is, the same proof standard is used for all positions). The purpose of the Consult Database button will be discussed in the next section.

Fig. 7: Adding a new constraint

Figure 7 illustrates the applet window for adding a new constraint to an issue. Depending on the argumentation element selected for its insertion, the *pair of items* pane provides users a menu with all valid pairs (combinations) of positions, preventing users from making errors in expressing a preference. The *relation type* menu includes the preference relations *more (less) important than* and *equally important to*. The pseudo-code for adding a constraint to an issue is as follows:

```

add_Constraint_to_Issue (constraint con, issue iss) {
    if (redundant(con)) return; /* ignore */
    refersTo(con)=iss;
    for all constraints conj that refer to iss

```

```

    if (conflicting( $con_j$ ,  $con$ ))
    {
        create new issue  $iss_n$ ;
        In( $con_j$ )=  $iss_n$ ;
        In( $con$ )=  $iss_n$ ;
        update( $iss_n$ );
        return; }
    if(contains_inactive( $con$ ))  $\neg$  active( $con$ );
    else {
        active( $con$ );
        if ( $\neg$  consistent_with_previous)( $con$ )
             $\neg$  consistent( $con$ );
        else {
            consistent( $con$ );
            update( $iss$ ); } }

```

Redundant and conflicting constraints were discussed in the previous section. Contains_inactive(con) checks whether one or both constituent positions are *inactive*, in which case con becomes *inactive*. The justification is that it does not make sense to argue about the relevant importance of positions that are rejected (con will be activated automatically when the constituent positions get activated — see **activate**(*position pos*)). If con does not contain_inactive, then it is checked for consistency. Only if it is found consistent, the position that it refers to is updated, since inconsistent constraints do not participate in the weighting scheme.

5.2 Internal acts

Internal acts are functions performed by the system in order to check consistency, update the discussion status and recommend solutions. These functions are automatically triggered by the agent acts (that is, whenever a new item is added in the discourse graph) and are hidden from the end user. For instance, consistent_with_previous(con), called by add_Constraint_to_Issue, constructs a graph similar to Figure 4 and applies path consistency. In the rest of this section, we present the pseudo-code and an analysis of the most representative internal acts:

```

boolean compute_activation(position pos) {
boolean  $\neg$  status_changed, old_activation=activation(pos);
switch Proof_Standard(pos) {
    case Scintilla of Evidence {
         $\neg$  activation(pos);
        for all positions  $pos_j$  that refer to pos
            if (active( $pos_j$ )  $\wedge$  in_favor( $pos_j$ , pos))
                {
                    activation(pos);
                    break ; } }
    case Beyond Reasonable Doubt {
        activation(pos);
        for all positions  $pos_j$  that refer to pos

```

```

        if (active(posj) ∧ against(posj, pos))
        {
            ¬ activation(pos);
            break ; }
    case Preponderance of Evidence {
        score(pos)=0;
        calculate_weights(pos);
        for all positions posj that refer to pos
            if (active(posj) ∧ in_favor(posj, pos))
                score(pos) += weight(posj);
            else if (active(posj) ∧ against(posj, pos))
                score(pos) -= weight(posj);
        if (score(pos) ≥ 0) activation(pos);
        else ¬ activation(pos) }
    if (old_activation ≠ activation(pos)) status_changed;
    return status_changed; }

```

Compute_activation returns *status_changed*, which is *true* if the activation label of the position changed. Activation is calculated according to the proof standard used. Proof standards are a straightforward implementation of the definitions given in Section 4.1. In case of PoE, *calculate_weights(pos)* calls *topological sort* to compute the weights of the positions in favor and against *pos*.

```

update(position pos) {
    if (compute_activation(pos)) {
        if (active(pos)) activate(pos);
        else inactivate(pos);
        update(RefersTo(pos)); } } /* propagate upwards */

```

Update calls **compute_activation** to check whether activation label has changed. If this is the case, **activate** (see below) or **inactivate** are called and the change is propagated upwards. Note that **update** is automatically called and executed by the system each time a user adds a new item (i.e., alternative, position, constraint) in the discourse graph.

```

activate(position pos) {
    posj=refersTo(pos);
    for all constraints conj that refer to posj
        if (¬ active(conj) ∧ conj has a preference relation on pos)
            if (compute_activation(conj) {
                if (consistent_with_previous(conj)
                    consistent(conj);
                else ¬ consistent(conj); } }

```

Activation of position *pos* involves the retrieval of the constraints where *pos* appears (that is, the constraints whose one of the constituent parts is the position *pos*) and a check as to whether they can now become *active* and *consistent*. Inactivation of positions is similar, in the sense that when a position becomes inactive, the system searches for the constraints where the position appears and inactivates them as well. This may cause some other inconsistent constraints to become consistent, so consistency check is performed again for the related constraints.

In addition to the above, other internal acts have been implemented for the activation/inactivation of constraints and the update of issues (in which case there is only PoE). The procedures described in this section, although only a subset of the whole set of functions performed by the system, give an indication of its dynamic structure. A single insertion in the discussion graph may update a large portion of the tree. Every time there is a change, the status of the argumentation elements is recorded in the database that keeps track of the discourse.

6. SYSTEM ENHANCEMENTS

This section describes the integration of additional features to HERMES. Having as ultimate goal the development of an advanced system for real CDM applications, we aim at providing users efficient means to further assist them during argumentation-based decision making processes.

6.1. The Information Retrieval Tool

In order to reach understanding, negotiate and resolve conflicts, users may use various types of information to warrant their arguments towards the selection or rejection of a statement or action. Such information may be stored either in the system's proprietary database, or in external, remote ones consisting of legacy data. The Information Retrieval Tool (IRT) allows users to ask queries regarding an ongoing discussion. Each time users want to add a position in the discussion graph, they can use IRT to retrieve auxiliary information from a pre-specified set of databases (the tool provides easy addition or removal of a database from this set).

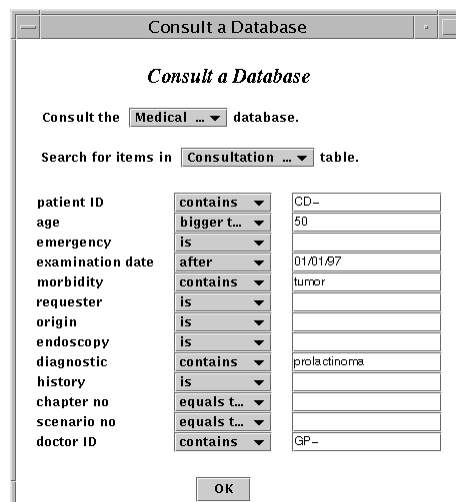


Fig. 8: Consulting a remote database

Figure 6 illustrates the applet window for adding a new position in favor of position-4: “Complete removal of tumor” (see again the example given in Section 4). This act can 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. Figure 8 illustrates such a query to a remote medical database. Note that this applet window is dynamically created, depending on the database and table selected by the user (this is achieved by exploiting *metadata* about the structure of each database). Also, depending on the contents of a database's table, the appropriate *selection menu buttons* for each entry will be placed (upon the data type of each field of the selected table). For instance, a selection menu with the options {before, after, is, is not} will be attached next to a date-type entry; a different one, with the options {contains, does not contain, is, is not}, next to a string-type entry, and so on. In the example shown in Figure 8, the user looks for patient cases where their examination date is after 01/01/97, morbidity contains the word “tumor”, diagnostic contains the word “prolactinoma”, the patient age (at the examination date) is larger than 50 years, and so on. Figure 9 depicts the applet window with the results of the query shown in Figure 8.

patient ID	age	emergency	exam. da...	morbidity	requester	origin	endoscopy	diagnostic	history	chapter No	scenario ...	doctor ID
CD-123...	55	high	23/04/98	tumor	Muller	Germany	NULL	prolactin...	NULL	3	87	GP-123
CD-765...	63	normal	12/01/97	tumor pi...	Martin	Switzerla...	PR-DA-23	prolactin...	NULL	2	65	GP-124
CD-443...	51	normal	25/03/98	tumor	Allaf	Lebanon	NULL	prolactin...	NULL	5	87	GP-123

Fig. 9: The query results

Communication with a set of 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 *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. For instance, in the example discussed in this paper, the users could consult a *MS Access* medical database, running on a PC platform, and an *Oracle* one, on Unix.

6.2. The Argument Builder Tool

Ongoing work is being performed towards the development of an Argument Builder Tool (ABT), which aims at assisting users to construct robust arguments. The tool basically deals with perusal of a discussion, in order to suggest the appropriate acts to be (individually) followed. It will extensively be interworking with IRT; having retrieved the required information, ABT will help decision makers to “wrap” these data with the appropriate discourse acts, to be then submitted to the corresponding discourse graph. The tool relies on a set of rules, concerning the *grammar* and *syntax* of CDM-related discussions. It comprises methods that, at any time, try to match *patterns* of an ongoing discussion with pre-specified ones (topological matching). For instance, if one's position is supported by only one piece of evidence, it will suggest him/her to take the appropriate actions to further

support it; in cases of conflicts between users (see previous section about conflicting constraints), it will advise them on actions for conflict resolution (such actions intend to achieve a deeper understanding of the issue to be solved). Note that, in any case, users are free to adopt or discard such suggestions. In addition, by exploiting Case-Based Reasoning (CBR) techniques, the tool will provoke the submission of the appropriate queries to the related databases and point out discourse acts that successfully reflect the interests and intentions of the parties involved (some preliminary work is presented in [Karacapilidis et al., 1997]). Such techniques have been particularly useful in the CDM domain, 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].

Related work has been recently performed in the context of the *Belvedere* system [Toth et al., 1997] to address students' tasks to find and use information about a scientific controversy (note that, in *Belvedere*, the required information exists in a prepared, rather modest, domain-relevant database of HTML documents; on the contrary, as described above, HERMES is able to access any-platform, remote databases). Another interesting approach, based on an integration of CBR and Multi-Attribute Utility Theory, is followed in PERSUADER [Sycara, 1987] to provide conflict resolution and negotiation support in labor-management disputes. In general, keeping track of previous cases (in HERMES's proprietary database) can also increase the efficiency of the system, by reusing previous instances of similar argumentation; such cases can be also attached as warrants to one's arguments.

7. EVALUATION RESULTS

HERMES has been evaluated by different types of users, such as high school and university students, well-experienced computer science researchers, medical doctors, and civil and mechanical engineers. Evaluation has been conducted in three research labs (namely, FIT-KI at GMD-Sankt Augustin, Germany, AID at INRIA-Sophia Antipolis, France, and LITH at EPFL, Switzerland), a university department (Dept. of Mechanical Engineering, EPFL, with the participation of 3 professors and 46 undergraduate students) and a hospital (CHUV, Lausanne, Switzerland, with the participation of 12 medical doctors). The average number of users in each research lab was 9. Furthermore, 16 high school students were involved in the evaluation during two public demonstrations of the system in the second lab.

In all evaluation stages, users worked with individual computers and in different rooms. Even for inexperienced users, a system introduction of less than an hour was sufficient to get them acquainted with HERMES. A human moderator supervised the argumentation and assisted the users whenever needed. The role of the supervisor was similar to that of a system's administrator, i.e., to provide access rights, make sure that elements were inserted to the right position in the discussion graph, etc.

During the early evaluation phases, we asked for feedback concerning the usability of the model, the discourse structure, and the functionality of the user interface. Setting up various pilot discussions and by applying their individual way of argumentation, users contributed significantly in what the system looks like today. Features like nested issues or the grouping of conflicting constraints in a separate issue resulted from various suggestions at this stage.

In the sequel, the system was evaluated through two real application projects. The first concerned an instructional environment for Mechanical Engineering undergraduate students, which use the system to solve a problem given by their supervisor [Abou Khaled et al., 1998]. Students contemplate alternative solutions and argue on them linking their argumentation elements with existing electronic documents (supervisor’s notes, papers, etc.). The second project was in the context of medical decision making; more specifically, it concerned the appropriateness of certain medical and surgical procedures for patients with various tumor-related symptoms (the examples used in this paper have been extracted from this project). The objective was the achievement of consensus as to whether a medical treatment or an operation is appropriate. Doctors used the system from remote machines, mostly in an asynchronous way, to express and validate their proposals and argue on those of the others. This particular application motivated us to enhance HERMES with IRT (due to the practices followed in real life and the numerous medical databases available).

Each discussion in the first real application project involved 3-6 users, depending on the problem under consideration (12 different discussions were constructed). A supervisor and some students were using the system for a period of one academic semester. In some problems, argumentation was quite lengthy, with more than 35 argumentation items submitted. As far as the second application is concerned, discussions lasted from 2 weeks to 5 months, with the participation of 3-5 medical doctors (the system was tested in 7 different discussions). In one case, the related Discussion Forum consisted of more than 80 argumentation items (5 users, about 4 months long), while most discussions comprised 20-40 items. In all cases, the system was properly performing (negligible execution time), indicating that HERMES can efficiently handle long argumentation processes.

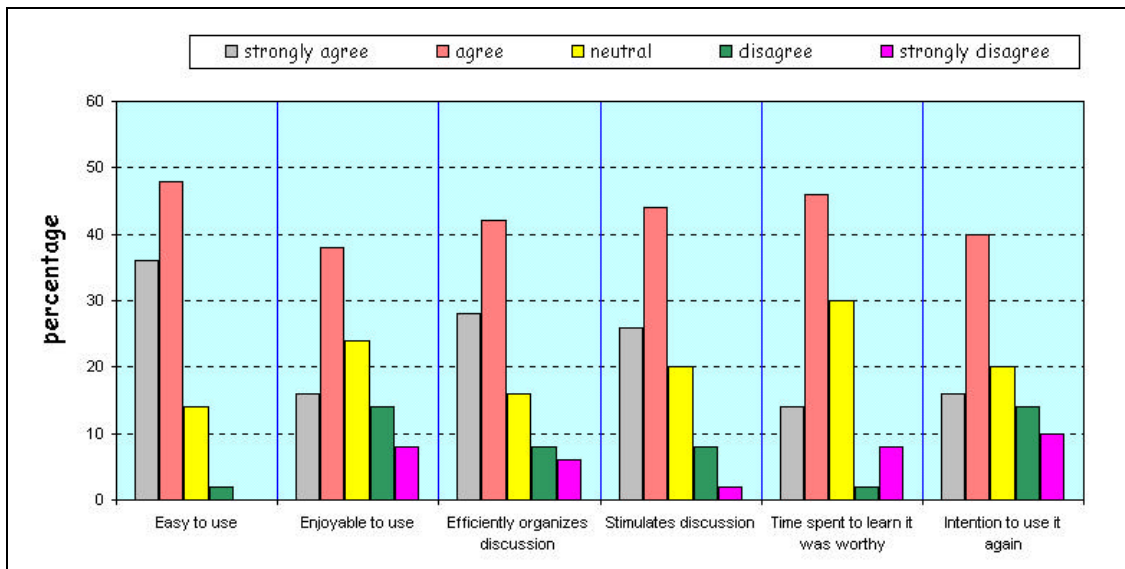


Fig. 10: Evaluation data – Questionnaire I

Two series of questionnaires were distributed to all users involved in these two projects (61 in total: 3 professors, 46 undergraduate students, 12 medical doctors). We received feedback from 50 of them (3 professors, 35 undergraduate students, 12 medical doctors). The objective of the first questionnaire was to evaluate the users’ overall opinion about the system. Users were asked whether HERMES is easy and enjoyable to use, provides

efficient support to organize an electronic discourse, and stimulates discussion. Also, whether the time spent working with it was worthy and they intend to use it again in the future. To fill the questionnaire in, users had to give an answer from the [strongly agree, agree, neutral, disagree, strongly disagree] scale (results are given in Figure 10). The objective of the second questionnaire was to get users' assessments about the effectiveness of the proposed CDM environment. More specifically, users had to evaluate the levels of interaction (answering whether what they had to do and had achieved using the system was clear, the content level was right and challenging, and the presentation was sufficiently informative), interfaces (answering whether windows are easy to read, terms and procedures for navigation are consistent, and functions, menus and icons provided are easy to understand) and navigation offered (answering whether it was easy to find out what they had already completed and what is still to be done, and what options were available). For all these issues, users had the [agree, neutral, disagree] choices to select from (results are given in Figure 11).

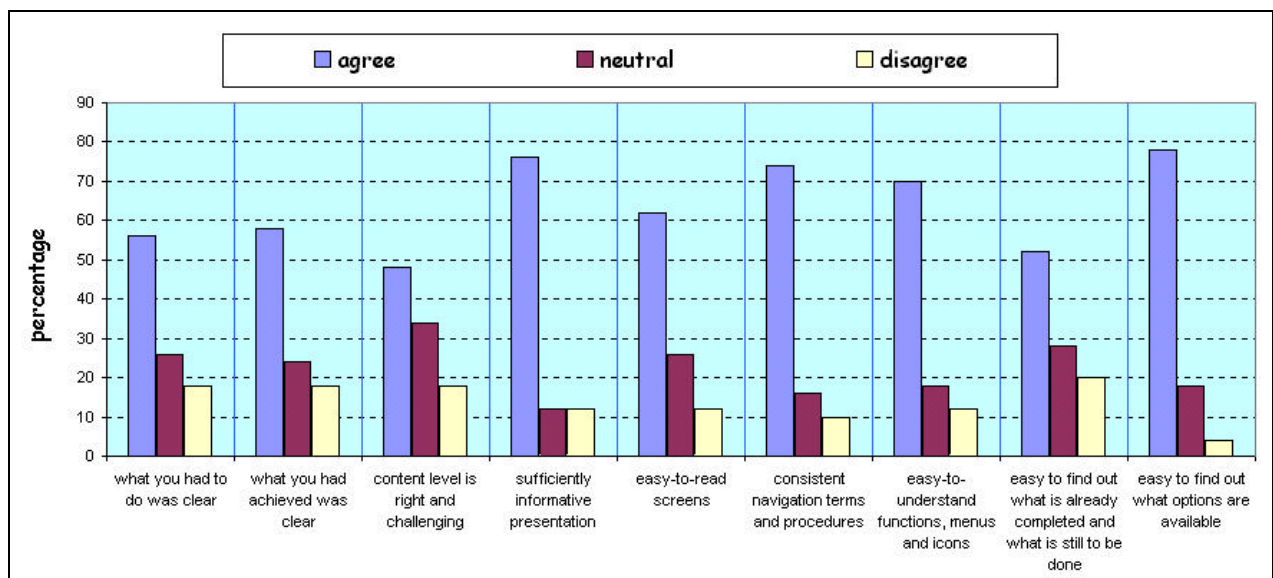


Fig. 11: Evaluation data – Questionnaire II

The feedback we received was certainly encouraging. In all issues raised in the first questionnaire, the percentage of the positive answers (sum of the strongly agree and agree percentages) was over 54% (note that it was 84% for the “easy to use” issue). Very positive results came also out of the second questionnaire; the percentage of the positive answers (agree) was in the range 48-78%, the users being more skeptical on the “right and challenging content level” issue. Having further elaborated their answers to this very issue, we concluded that this was due to the change of the way they were accustomed to work. However, the majority of the users was convinced that the appropriate integration of web-based tools in a collaborative decision making environment has a high potential to enhance its effectiveness.

8. CONCLUSION

We have presented HERMES, a system that aids decision makers reach a decision, not only by efficiently structuring the discussion rationale, but also by providing reasoning mechanisms that constantly update the discourse status in order to recommend the most backed-up alternative. Our primary goal was to develop an *active* system, which not

only “captures the informal organizational memory” embodied in decision making settings, but also helps the users during the decision making process *per se*. We view human intervention as necessary to determine certain instances of the argumentation items being asserted and the relations holding between them. HERMES acts as an *assistant* and *advisor*, by facilitating communication and recommending solutions, but leaving the final enforcement of decisions and actions to the decision makers involved. Exploiting the system’s features, users can better understand the arguments of their fellow decision makers, and advance more pointed argumentation of their own.

It is our belief that the integration of the system 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 CDM 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 framework, appears to be feasible.

Acknowledgements: The authors thank Gerhard Brewka, Tom Gordon, Henry Prakken and Hans Voss for helpful discussions and comments during the development of the system described in this paper.

REFERENCES

- A. Aamodt and E. Plaza. Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches. *AI Communications* 7 (1), 39-59 (1994).
- O. Abou Khaled, N. Karacapilidis, Y. Rekik and C. Vanoirbeek. MEDIT: A Web-based environment for advanced pedagogical issues. In C. Alvegard (ed.) *Proceedings of the 4th International Conference on Computer-Aided Learning and Instruction in Science and Engineering (CALISCE'98)*, Göteborg, Sweden, Chalmers University of Technology, 325-333 (1998).
- G. Brewka. A Reconstruction of Rescher's Theory of Formal Disputation Based on Default Logic. In *Working Notes of the AAAI'94 Workshop on Computational Dialectics*, July 31 - August 4, Seattle, WA, 15-27 (1994).
- E. J. Conklin. Designing Organizational Memory: Preserving Intellectual Assets in a Knowledge Economy. GDSS Working Paper (1996). Available at: <http://www.gdss.com/wp/DOM.htm>
- E. J. Conklin. Capturing Organizational Memory. In D. Coleman (ed.) *Proceedings of the GroupWare'92 Conference*, Morgan Kaufmann, San Mateo, CA, 133-137 (1992).
- E. J. Conklin and M. L. Begeman. gIBIS: A Hypertext Tool for Team Design Deliberation. In *Proceedings of the Hypertext'89 Conference*, ACM Press, New York, 247-252 (1987).
- G. De Michelis and M.A. Grasso. Situating conversations within the language/action perspective: the Milan Conversation Model. In *Proceedings of the CSCW'94 Conference*, ACM Press, New York, 89-100 (1994).
- B. Di Eugenio, P.W. Jordan, R. H. Thomason and J. D. Moore. Reconstructed Intentions in Collaborative Problem Solving Dialogues. In *Proceedings of the AAAI-97 Fall Symposium on Communicative Action in Humans and Machines*, Cambridge, MA, AAAI Press (1997).

- F. H. van Eemeren, R. Grootendorst and F. Snoeck Henkemans. *Fundamentals of Argumentation Theory*. Erlbaum, Mahwah, NJ (1996).
- A. M. Farley and K. Freeman. Burden of Proof in Legal Argumentation. In *Proceedings of the 5th Int. Conference on AI and Law*, ACM Press, New York, 156-164 (1995).
- G. Fischer, R. McCall and A. Morch. JANUS: Integrating Hypertext with a Knowledge-based Design Environment. In *Proceedings of the Hypertext '89 Conference*, ACM Press, New York, 105-117 (1989).
- J. Fox and S. Parsons. On Using Arguments for Reasoning about Actions and Values. In Proceedings of the AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning, Stanford, CA (1997)
- H. Geffner and J. Pearl. Conditional Entailment: Bridging two Approaches to Default Reasoning. *Artificial Intelligence* 53(2-3), 209-244 (1992).
- M. Goldszmidt and J. Pearl. Qualitative Probabilities for Default Reasoning, Belief Revision, and Causal Modeling. *Artificial Intelligence* 84, 57-112 (1996).
- T. Gordon. *The Pleadings Game: An Artificial Intelligence Model of Procedural Justice*. Kluwer Academic Publishers, Dordrecht, The Netherlands (1995).
- T. Gordon and N. Karacapilidis. The Zeno Argumentation Framework. In *Proceedings of the ICAIL '97 Conference*, ACM Press, New York, 10-18 (1997).
- D. W. Green. Arguments and Mental Models: A Position Paper. In D. M. Gabbay and H. J. Ohlbach (eds.): *Practical Reasoning, Proceedings of the International Conference on Formal and Applied Practical Reasoning (FAPR '96)*, Bonn, Germany, June 3-7, Lecture Notes in Computer Science, Vol. 1085, Springer, 697-704 (1996)
- C. H. Heidrich. Montague-grammars for Argumentative Dialogues. In Barth and Martens (eds.) *Argumentation: Approaches to Theory Formation*. John Benjamins, Amsterdam, The Netherlands (1982).
- R. Hurwitz and J. C. Mallery. The Open Meeting: A Web-Based System for Conferencing and Collaboration. In *Proceedings of the 4th International World Wide Web Conference*, December 11-14, Boston, MA (1995). Available at: <http://www.ai.mit.edu/projects/iiip/doc/open-meeting/paper.html>
- N. Karacapilidis and D. Papadias. HERMES: Supporting Argumentative Discourse in Multi-Agent Decision Making, in *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98)*, Madison, WI, AAAI/MIT Press, 827-832 (1998a).
- N. Karacapilidis and D. Papadias. A Computational Approach for Argumentative Discourse in Multi-Agent Decision Making Environments, *AI Communications Journal* 11(1), 21-33 (1998b).
- N. Karacapilidis, B. Trousse and D. Papadias. Using Case-Based Reasoning for Argumentation with Multiple Viewpoints. In D. Leake and E. Plaza (eds.) *Case-Based Reasoning: Research and Development*, Lecture Notes in AI, Vol. 1266, Springer-Verlag, Berlin, 541-552 (1997).
- D. E. Knuth. *Fundamental Algorithms*. Art of Computer Programming, Vol. 1. Addison-Wesley, 2nd edition (1973).
- K. L. Kremer and J. L. King. Computer-based systems for cooperative work and group decision making. *ACM Computing surveys* 20, 2, 115-146 (1988).

- W. Kunz and H. Rittel. Issues as Elements of Information Systems. Technical Report, 0131. Universität Stuttgart, Institut für Grundlagen der Planung (1970).
- J. Lee. SIBYL: A Tool for Managing Group Decision Rationale. In *Proceedings of the CSCW'90 Conference*, ACM Press, New York, 79-92 (1990).
- G. L. Lueken. Incommensurability, Rules of Argumentation and Anticipation. In F.H. Van Eemeren, R. Grootendorst, J.A. Blair and Ch. A. Willard (eds.) *Proceedings of the 2nd International Conference on Argumentation*, June 19-22, Amsterdam, The Netherlands, 244-252 (1991).
- A. Mackworth and E. Freuder. The Complexity of some Polynomial Network Consistency Algorithms for Constraint Satisfaction Problems. *Artificial Intelligence* 25, 65-74 (1985).
- A. MacLean, R.M. Young, V. Bellotti and T. Moran. Questions, Options and Criteria: Elements of Design Space Analysis. *HCI* 6 (3-4), 210-250 (1991).
- J. Mylopoulos, A. Borgida, M. Jarke and M. Koubarakis. Telos: Representing knowledge about information systems. *ACM Transactions on Information Systems* 8, 325-362 (1990).
- G. Pinkas. Propositional Non-Monotonic Reasoning and Inconsistency in Symmetric Neural Networks. In *Proceedings of the IJCAI'91 Conference*, Morgan Kaufman Publishers, San Mateo, CA, 525-530 (1991).
- H. Prakken. From Logic to Dialectics in Legal Argument. In *Proceedings of the 5th Int. Conference on AI and Law*, ACM Press, New York, 165-174 (1995).
- B. Ramesh and V. Dhar. Supporting Systems Development by Capturing Deliberations During Requirements Engineering. *IEEE Transactions on Software Engineering* 18 (6), 498-510 (1992).
- N. Rescher. *Dialectics: A Controversy-Oriented Approach to the Theory of Knowledge*. State University of New York Press, Albany (1977).
- H. Rittel and M. Webber. Dilemmas in a General Theory of Planning. *Policy Sciences*, 155-169 (1973).
- W. N. Robinson and S. Fickas. Supporting Multi-Perspective Requirements Engineering. In *Proceedings of the First International Conference on Requirements Engineering*, IEEE Computer Society Press, April 18-22, 206-215 (1994).
- W. N. Robinson and S. Volkov. A Meta-Model for Restructuring Stakeholder Requirements. In *Proceedings of the 19th International Conference on Software Engineering*, May 17-24, Boston, IEEE Computer Society Press, 140-149 (1997).
- T. Saaty. *The analytic hierarchy process*. Mc-Graw Hill, New York (1980).
- S. Shum, A. MacLean, J. Forder and N. Hammond. Summarising the Evolution of Design Concepts within a Design Rationale Framework. In *Adjunct Proceedings of InterCHI'93: ACM/IFIP Conference on Human Factors in Computing Systems*, April 24-29, Amsterdam, 43-44 (1993).
- G. R. Simari and R. P. Loui. A Mathematical Treatment of Defeasible Reasoning and its Implementation. *Artificial Intelligence* 53, 2-3, 125-157 (1992).

- P. Smolensky, B. Fox, R. King and C. Lewis. Computer-aided Reasoned Discourse, or How to Argue with a Computer. In Guindon, R. (ed.) *Cognitive Science and its Applications for Human-Computer Interaction*, Erlbaum, Hillsdale, NJ, 109-162 (1987).
- N. Streitz, J. Hannemann and M. Thüring. From ideas and arguments to hyperdocuments: Travelling through activity spaces. In *Proceedings of the Hypertext '89 Conference*, ACM Press, New York, 343-364 (1989).
- D. Suthers, A. Weiner, J. Connelly and M. Paolucci. Belvedere: Engaging Students in Critical Discussion of Science and Public Policy Issues. In *Proceedings of the 7th World Conference on Artificial Intelligence in Education (AI-ED '95)*, August 16-19, Washington DC, 266-273 (1995).
- K. Sycara. Resolving Adversarial Conflicts: An Approach Integrating Case-Based and Analytic Methods. Ph.D. dissertation, School of Information and Computer Science, Georgia Institute of Technology (1987).
- S. W. Tan and J. Pearl. Qualitative Decision Theory. In *Proceedings of the AAAI'94 Conference*, AAAI/MIT Press, 928-933 (1994).
- J. A. Toth, D. Suthers and A. Weiner. Providing Expert Advice in the Domain of Collaborative Scientific Inquiry. In *Proceedings of the 8th World Conference on Artificial Intelligence in Education (AI-ED'97)*, August 20-22, 1997, Kobe, Japan (1997).
- P. Valesio. *Novantiqua: Rhetorics as a Contemporary Theory*. Indiana University Press (1980).