A Group Decision and Negotiation Support System for Argumentation Based Reasoning

Nikos Karacapilidis* Dimitris Papadias**

* INRIA Sophia Antipolis, Action AID
2004 Route des Lucioles BP 93
06902 Sophia Antipolis Cedex, France
e-mail: nikos@sophia.inria.fr

** Dept. of Computer Science
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
e-mail: dimitris@cs.ust.hk

Abstract. This paper describes a Group Decision and Negotiation Support System for cooperative or non-cooperative argumentative discourses. The system provides agents means of expressing and weighing their individual arguments and preferences, the aim being the selection of a certain choice. It also supports defeasible and qualitative reasoning in the presence of ill-structured information. Argumentation is performed through a set of discourse acts which call a variety of procedures for the propagation of information in the corresponding discussion graph.

1 Introduction

Numerous researchers have dealt with argumentative discourse from different standpoints such as *philosophical* (where argumentation is considered as a way of theory formation [23]), *linguistic* (inspired by speech act theory and discourse analysis [17]), *rhetorical* (where argumentation is studied primarily as a form of manipulation through language [38]), and *psychological* (where the relationships between arguments and cognitive processes are explored [16]). Most approaches follow two main methodologies, namely, *formal* and *informal* logic (for an extensive survey see [12]).

According to the formal perspective, arguments are de-contextualized sets of sentences or symbols viewed in terms of their syntactic or semantic relationships. On the other hand, informal logic views arguments as pragmatic, i.e., their meaning is a function of their purposive context[1]. Most AI researchers have focused on formal models of argumentation based on various logics. For instance, Brewka [7] reconstructed Rescher's [30] theory of formal disputation, Prakken [27] used Reiter's default logic [29], while Gordon's work [15] was based on Geffner and

[1] The term informal logic was primarily used to denote an approach to argument interpretation and evaluation that is distinct from traditional, formal logic.

Pearl's concepts of conditional entailment [14]. An extensive discussion on the use of alternative logics in argumentative discourse appears in [28]. Based on these methodologies, several models of dialectical systems have been proposed [15], [28]. They basically consist of a set of abstract components, that specify the underlying logic, the speech acts allowed, and the protocol of rights and responsibilities of the participants.

Instead of proposing another argumentation theory, or presenting a new formal model, this paper describes a Group Decision Support System for argumentation based reasoning. The system is implemented in Java and can be run on the World Wide Web using any standard browser. Our primary goal was the implementation of an argumentation system, by which groups of natural or artificial agents can express their claims and judgements, aiming at informing or convincing, depending on the kind of the interaction. The system can be used in any kind of group decision making processes, and is able to handle inconsistent, qualitative and incomplete information in cases where one has to weigh reasons for and against the selection of a certain course of action.

Many problems involving argumentative discourse are very complex, requiring large amounts of information and collaboration of numerous participants from different contexts (e.g., resource managers, politicians, citizen groups, etc.). Part of our goals was to exploit information technology in order to meet three major practical requirements:

- the system must provide relatively inexpensive access to a broad public;
- it must have an intuitive interface in order to be easily usable by inexperienced users, and
- it must be available on all prominent operating systems and hardware platforms.

The above requirements are ideally met by the World Wide Web, and as a consequence, some systems have already been implemented on this platform (for an on-line survey see [40]). Current systems, however, merely provide threaded, hierarchical discussion forums, where a message can be posted specifically as a response to another message (such a system is Open Meeting, a highly structured conferencing system that supports on-line meetings with numerous participants [18]). On the contrary, our system includes reasoning mechanisms that monitor the discussion, constantly updating the status of argumentation. The system focuses on distributed, asynchronous collaboration, allowing agents to surpass the requirements of being in the same place and working at the same time.

The next section presents the argumentation elements of the system. Section 3 describes the proof standards, while Section 4 illustrates the discourse acts involved. The latter relate to a variety of procedures for the propagation of the information, such as consistency checking, aggregation of preferences, and labelling of the argumentation elements according to different standards. Section 5 discusses related work, focusing on the extraction of the pros and cons of our approach, and proposes directions for future work. Finally, Section 6 concludes the paper.

2 Elements of Argumentation

Our approach maps a multi-agent decision making process to a discussion graph with a tree-like structure. For the design of the model described in this paper, we have been inspired by the informal *Issue-Based Information System* (IBIS) model of argumentation (first introduced in [22], [32], later developed at MCC [39], [10]. Throughout this paper, we refer to a discussion example about the selection of a site for the location of a new airport in Southern France (there exist three alternative sites). The agents involved in the discussion bring up the necessary argumentation in order to express interests and perspectives of the affected parties, such as communities, related regional and federal Planning Departments, environmental protection groups, and industry. Our argumentation model has as elements *issues, alternatives, positions, arguments,* and *constraints* representing *preference relations.* IBIS used the concepts of *topic* (serving as a crude organization principle for denoting the foci of concern), *issue, position* and *question of fact* (requesting information which is not assumed to be controversial).

2.1 Issues, alternatives, positions and arguments

Issues correspond to decisions to be made, or goals to be achieved (e.g., issue I_1: "choose the most appropriate site for the new airport"). They are brought up by agents and are open to dispute. There is always an issue at the top (root) of the discussion tree. Issues consist of a set of *alternatives* that correspond to potential choices (e.g., alternative A_1 to I_1: "Select the Biot site", alternative A_2 to I_1: "Select the Grasse site"). Issues can be inside other issues in cases where some alternatives need to be grouped together. For instance, assume that there are two potential sites at the greater Biot area, namely, "Select the northern Grasse site" and "Select the southern Grasse site"). In this case, the last two alternatives should be grouped in an internal issue ("select the most appropriate site in Biot") and, after the related discourse, the selected (best) one will be compared to the alternatives A_2 and A_3 (alternative A_3 to I_1: "Select the Cagnes site"). Only one alternative of an issue is finally selected.

Positions are asserted in order to advocate the selection of a specific course of action (alternative), or avert the agents' interest from it. For instance, position $P_{1.1}$: "alternative A_1 is a public land parcel" is asserted to support A_1, while position $P_{1.2}$: "alternative A_1 is an environmentally sensitive zone" expresses one's objection to A_1. Positions may also refer to some other position in order to provide additional information about it (e.g., positions $P_{1.1.1}$: "no purchase cost (if a site is a public land parcel)", and $P_{2.2.1}$: "local authority report (providing details about the owners of the land for the Grasse site)"). A position always *refers to* a single other position or alternative, while an alternative or issue is always *in* a single issue. This results to a tree-like argumentation structure. Figure 1 illustrates an instance of the discussion, where *refers to* is indicated by lines, alternatives and positions by ellipses, and issues by rectangles (actual graphs may have many more elements organized in multiple levels).

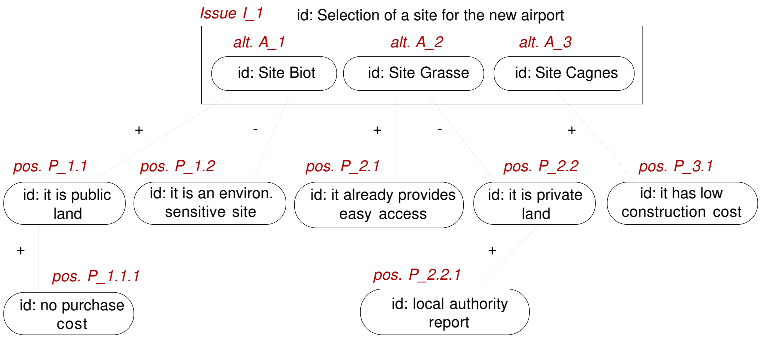

Fig. 1. An instance of the discussion graph for the airport example.

Numerous meanings and formalisms of the term *argument* have been proposed in the literature [28], in order to address various application areas [2]. For instance, in [9], an argument consists of a *support base*, that may contain formulas which speak for or against a certain position, and a *conclusion*. A similar notion of argument has been given in [3][2]. An argument in our system is a tuple of either the form $(position, link, position)$ or $(position, link, alternative)$. In other words, it links together a position with an alternative or another position belonging to a different issue. Agents put forward arguments to convince their opponents or to settle an issue via a formal decision procedure. We distinguish between *supporting arguments* and *counterarguments*. As in real life applications, two conflicting arguments can be simultaneously applied (see also [5], [6]).

In implementation terms, the elements of our system are illustrated in Figure 2 in the form of an Entity-Relationship diagram [1] (according to the E-R model, entities are represented as rectangles, attributes as ellipses, and relations as diamonds). The *Basic Element* consists of two attributes: an *Id* (primary key) and a *Url* that contains relevant information (sender of the argumentation element, submission date, comments, etc.). It is an abstract class that acts as a superclass for all the others. An *issue* is a basic element that contains other issues and alternatives. Each alternative has a *weight*, which is determined by its supporting and counter-arguments, an *activation label* indicating its status (it can be *active* or *inactive*), and a *proof standard* (discussed in the next section). In addition, each position has a boolean *link* that denotes whether it participates in a supporting (+), or a counter-argument (-).

Active alternatives correspond to "recommended" choices, i.e., choices that are the strongest among the alternatives in their issue. Active positions are considered "accepted" due to the discussion underneath (e.g., strong supporting arguments, no counter-arguments), while inactive positions are considered "rejected". The mechanisms for the calculation of activation labels are described in Section 3.

[2] Both are extensions of the one proposed in [34].

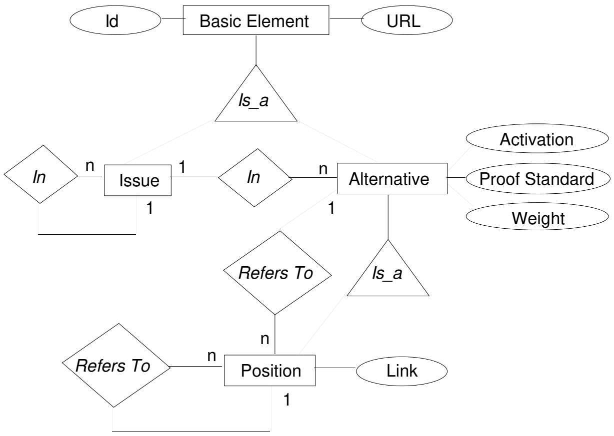

Fig. 2. Elements of the model.

2.2 Constraints

In many cases, we have to compare arguments in favor of the same conclusion, as well as arguments leading to contradictory conclusions. The subjects of priority relationships and preference orders between arguments have been mostly handled through quantitative approaches. For example, [25] and [33] have used the concepts of *penalty logic* (cost of not taking a premise into account) and *confidence factors*, respectively. Unfortunately, well defined utility and probability functions regarding properties or attributes of alternatives (used, for instance, in traditional OR approaches), as well as complete ordering of these properties are usually absent. On the other hand, non-monotonic formalisms have defined preference relations on subtheories in various ways (see for example the concepts of *preferred subtheories* [5], *conditional entailment* [14], *prioritized syntax-based entailment* [4]), and *Qualitative Value Logic* [8], [19].

Preference relations provide a qualitative way to weigh reasons for and against the selection of a certain course of action. In our system, a *constraint* is a tuple $(Position, Preference\_Relation, Position)$, where the preference relation can be "$>$", "$=$", "$<$" ("$>$" is interpreted as *more important than* while "$=$" as *of equal importance as*)[3]. Constraints may attach various levels of importance to positions or alternatives. They may *refer to* another constraint, a position, or an issue. A constraint, like the other elements, is subject to discussion; therefore, it may be "linked" with positions supporting or challenging it, as well as other constraints that refer to it. Activation labels of constraints are calculated by the same proof standards used for positions. Figure 3 illustrates the E-R diagram for constraints.

[3] See also [20] introducing the following preference relations: *SMP (strongly more preferable), SLP (strongly less preferable), EQP (equally preferable), WMP (weakly more preferable)* and *WLP (weakly less preferable).*

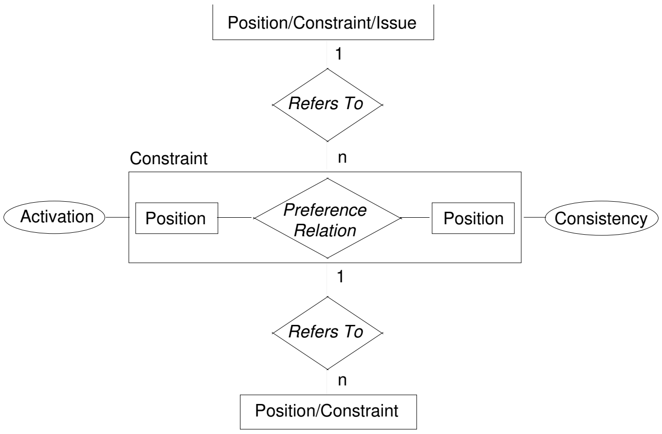

Fig. 3. E-R diagram for constraints.

Two types of preferences can be expressed by the system:

- *Local* preferences: when a constraint refers to a position, or another constraint, the positions that constitute the parts of the constraint must refer to the same element. Assume that in the example shown in Figure 1, there is also a position $P_{2.2.2}$:"Federal Government Map". Then, a preference of this type can be expressed by the constraint "$P_{2.2.1}$ (local report) is more important (accurate) than $P_{2.2.2}$ (federal map)". A constraint of the form "$P_{1.2}$ is more important than $P_{1.1.1}$" is not permitted.
- *Non-local* preferences: when a constraint refers to an issue, its parts must refer to alternatives (not necessarily the same one) of this issue (e.g., in Figure 1, "$P_{2.2}$ (the fact that Grasse site is a private land parcel) is less important than $P_{1.2}$ (Biot being an environmentally sensitive zone)"). Issues provide a way of grouping comparable elements together, since they permit the expression of preferences between positions attached to different alternatives.

These rules about the formation of constraints are preserved by the interface. When an agent wants to assert a constraint that refers to a position P_i, the related interface permits only a preference relation between the positions linked with P_i.

3 Proof Standards

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 [13]. 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 our system (the names have the same interpretation as in the legal domain). We do not claim that the list is exhaustive; other standards, that match specific application needs, can be easily incorporated to the system.

Scintilla of Evidence (SoE): according to this proof standard, a position is active if at least one active position is supporting it (linked with "+"):
$active(P_i) \Leftarrow \exists P_j \ni (active(P_j) \wedge link(P_j) \wedge Refers\_To(P_j, P_i))$.

Beyond Reasonable Doubt (BRD): according to *BRD*, a position is active if there are not any active positions that speak against it (linked with "-"):
$active(P_i) \Leftarrow \neg\exists P_j \ni (active(P_j) \wedge \neg link(P_j) \wedge Refers\_To(P_j, P_i))$.

In most cases the same proof standard is used for all positions in the argumentation graph in order to achieve fairness. However, when there is a reason, the system allows the discussion moderator to change the proof standard for individual positions.

Activation in our system is a recursive procedure; a change of the activation label of an element is propagated upwards, until the root of the discussion tree. Consider the two instances of the discussion illustrated in Figure 4, assuming that the proof standard is *BRD* for all elements (as discussed in the sequel, alternatives have another proof standard, namely Preponderance of Evidence; we assume here that A_2 has also the BRD standard to give a more illustrative example of the propagation performed). At the left hand side instance, $P_{2.2}$ is *active* (there is only a position supporting it), while A_2 is *inactive* (since $P_{2.2}$ is active). As shown in the right hand side instance, when $P_{2.2.2}$ is inserted, $P_{2.2}$ becomes *inactive*. This change is propagated upwards, making A_2 *active*. Note here that the leaf nodes of the tree (elements with no argumentation underneath) are always considered to be *active*[4].

When an alternative is affected during the discussion, the issue it belongs to should be updated because a new choice may be made. This brings us to the last proof standard, *Preponderance of Evidence (PoE)*, used for the selection of the best alternative (however, it can also be used for activation/inactivation of positions).

In the case of *PoE*, each position has a $weight = (max\_weight + min\_weight)/2$ while an alternative has $weight = min\_weight$. Max_weight and min_weight are initialized to some pre-defined values (for the following discussion we assume that initially $min\_weight = 0$ and $max\_weight = 10$; this may be changed by preference constraints). The *score* of an element E_i (position or alternative) is used to compute its *activation label* (score is calculated

[4] Similar ideas have been proposed in [11], viewing argumentation as a special form of logic programming. In this model, an argumentation model consists of two parts: an argument generation unit and an argument processing unit. The second one is a logic program consisting of the clauses $\texttt{acc(x)} \leftarrow \neg\texttt{defeat(x)}$ (an argument is acceptable if it is not defeated) and $\texttt{defeat(x)} \leftarrow \texttt{attack(y,x)}, \texttt{acc(y)}$ (an argument is defeated if it is attacked by an acceptable argument).

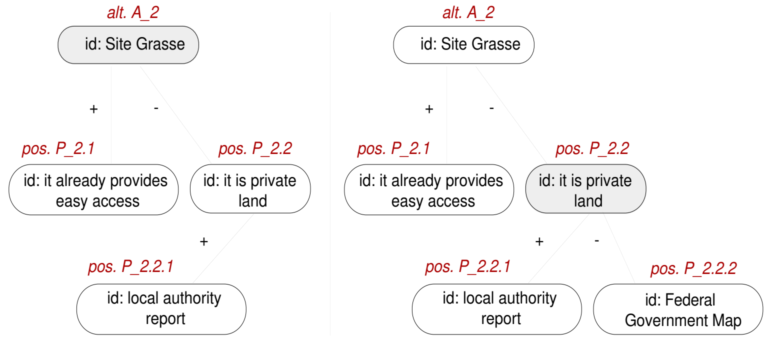

Fig. 4. Two instances of the discussion tree.

on-the-fly and is not stored in the database; therefore, it does not appear in the E-R diagram of Figure 2). 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 the element:

$$score(E_i) \;=\; \sum_{\substack{Refers\_To(P_j,P_i)\wedge\\ link(P_j)\wedge active(P_j)}} weight(P_j) \;-\; \sum_{\substack{Refers\_To(P_k,P_i)\wedge\\ \neg link(P_k)\wedge active(P_k)}} weight(P_k) \qquad (1)$$

Preponderance of Evidence: According to this standard, a position is *active* when the active positions that support it outweigh those that speak against it: $active(P_i) \Leftarrow score(P_i) \geq 0$.
In case of alternatives, *PoE* will produce positive activation label (meaning that the alternative is a "recommended" choice) when there are no alternatives with larger score:
$active(A_i) \Leftarrow \forall A_j \; inissue(A_i),\; (score(A_j) \leq score(A_i))$.

The score of an element is used only for the calculation of its own activation label and does not propagate upwards. Before the insertion of $P_{2.2.2}$ in Figure 4, the scores are: $score(P_{2.1}) = score(P_{2.2.1}) = 5$, $score(P_{2.2}) = 5$, and $score(A_2) = 0$ (a counter-argument cancels a supporting one). When $P_{2.2.2}$ is inserted, the scores become: $score(P_{2.2}) = 0$, and $score(A_2) = 5$. In general, large positive scores indicate strong acceptance and large negative ones strong rebuttal of a claim (e.g., when another position that speaks against $P_{2.2}$ is inserted, it will be $score(P_{2.2}) = -5$).

Every time a constraint is inserted in the discussion graph, the system checks for *redundancy*. It actually checks if both positions of the new constraint exist in another, previously inserted, constraint. If this is the case, the new constraint is redundant (independently of its preference relation) and ignored. If not, its consistency is checked against previous active and consistent constraints referring to the same element and its consistency label is determined. This process is based on a polynomial ($O(N^3)$) *path consistency* algorithm [24]. Although path

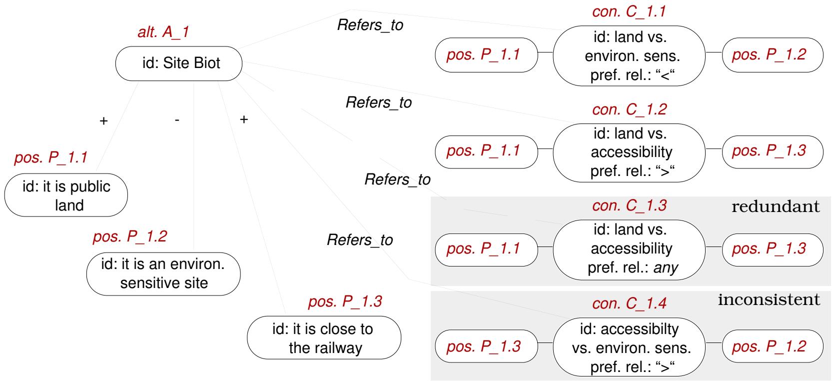

Fig. 5. Redundancy and inconsistency among constraints.

consistency, as most discourse acts described in the sequel, interacts with the database that stores the discussion graph, the algorithm is very efficient for local preferences, because in most cases N (the number of positions that refer to a single position, or constraint) is small. Even for non-local preferences involving issues with numerous alternatives and arguments, execution time is negligible compared to communication delay. In Figure 5, assume that constraints $C_{1.1}$ and $C_{1.2}$ have already been asserted. The new-coming constraint $C_{1.3}$ will be ignored as redundant, while $C_{1.4}$ will be labeled *inconsistent*.

The activation label of constraints can change due to two factors: the discussion underneath and the activation label of their consistuent positions. As shown in Figure 6 (which illustrates the discussion instance after that of Figure 5), the insertion of position $PC_{1.2.1}$ inactivates constraint $C_{1.2}$ (assume *BRD* as proof standard for all positions and constraints). In addition, position $P_{1.1.2}$ inactivates position $P_{1.1}$, which in turn inactivates constraint $C_{1.1}$ (we assume that $C_{1.2}$ is already *inactive*). $C_{1.4}$ remains active because it does not involve $P_{1.1}$. Notice that $C_{1.4}$ now becomes consistent (path consistency runs whenever there is a change to the activation status of a constraint), and is the only constraint taken into account by the scoring mechanism.

Active and consistent constraints participate in the weighting scheme. In order to demonstrate how the algorithm for altering weights works, we use the example of Figure 7. There exist five positions and four constraints that relate them as illustrated in Figure 7a. The arrowed lines correspond to "$>$" (e.g., $P_1 > P_2$) and the dotted line to "$=$" (e.g., $P_3 = P_4$). First path consistency explicates all "$>$" relations (Figure 7b). Then *topological sort* [21] is applied twice to compute the possible maximum and minimum weights for each position (Figure 7c). The *weight* is the average of the new *max_weight* and *min_weight*: $weight(P_1) = 6, weight(P_2) = 4.5, weight(P_3) = 5, weight(P_4) = 5$ and $weight(P_5) = 4$.

The basic idea behind the above scheme is that the weight of a position is increased every time the position is more important than another one (and

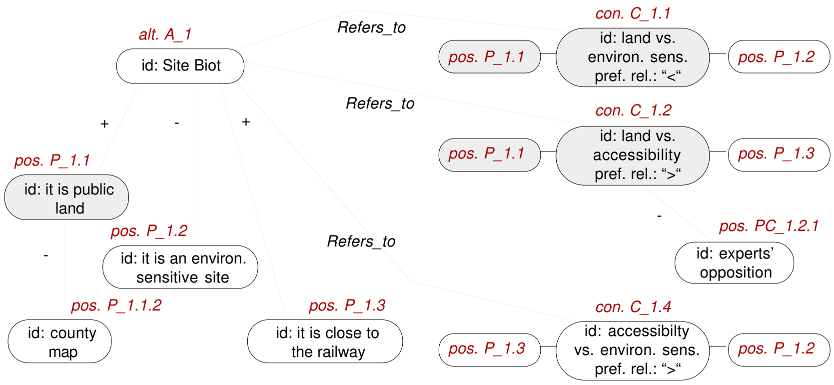

Fig. 6. Inactivation of constraints.

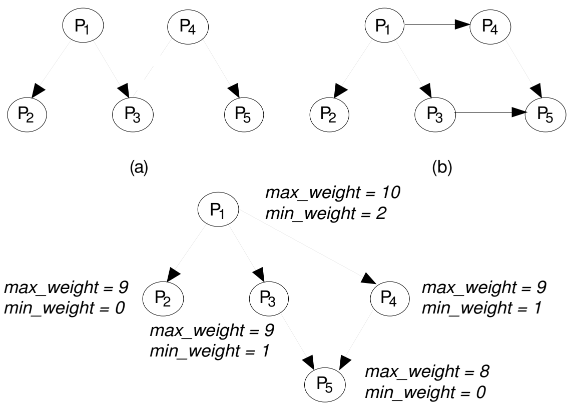

Fig. 7. The weighting scheme.

decreased when 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, can be developed.

4 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 three major categories: *agent acts, internal (system) acts* and *moderator acts*. Agent acts refer to user actions and correspond to functions directly supported by the user interface. Such functions include opening an issue, adding an alternative, etc. Below we present the pseudo-code for a few representative agent acts:

```
add_Alternative_to_Issue(alternative A, issue I) {
    In(A)=I;          /* A is made a part of I */
    update(I); }      /* I is updated */

add_Alternative_to_Alternative(alternative A_i, alternative A_j) {
    I_j=In(A_j);
    create new issue I_i;
    In(I_i)=I_j;
    In(A_j)=In(A_i)=I_i;
    update(I_i); }
```

When an alternative A_i is added to another A_j (and not to the issue I_j where A_j belongs) a new issue I_i is created inside I_j. Both A_i and A_j are now put inside the new issue and compared through update(I_i). update(I_j) will be called from update(I_i) and the recommended choice between A_i and A_j will be compared against the other alternatives of the external (original) issue.

```
add_Constraint_to_Position(constraint C, position P) {
    RefersTo(C)=P;
    if (redundant(C)) exit;
    if (contains_Inactive(C)) ¬active(C);
        else {
        active(C);
        if (¬consistent_with_Previous(C))
            ¬consistent(C);
        else
        {   consistent(C);
            update(P); }}}
```

Contains_Inactive(C) checks whether one or both positions related by C are *inactive*, in which case C becomes *inactive*. The justification is that it does not make sense to argue about the relevant importance of positions that are rejected (C will be activated automatically when the compared positions get activated - see activate(*position* P)). If C 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.

Other agent acts involve addition of constraints to issues, insertion of supporting and counter-arguments to positions, constraints and alternatives, etc. There is also a second type of agents acts, which is not immediately connected to the discussion forum, but allows agents to ask the 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*".

Internal acts are functions performed by the system in order to check consistency, update the discussion status and recommend solutions. These functions are called mainly by the agent or moderator acts and are hidden from the end user. For instance, consistent_with_Previous(C), called by add_Constraint_to_Position, constructs a graph (similar to Figure 7) from the arguments of the position where C refers to and applies path consistency. Some other representative internal acts are:

```
boolean compute_activation(position P) {
boolean ¬ status_changed, old_activation=activation(P);
switch Proof_Standard(P) {
        case Scintilla of Evidence {
                ¬activation(P);
                for all positions Pj that refer to P
                        if (active(Pj) ∧ link(Pj))
                        {       activation(P);
                                break; }}
        case Beyond Reasonable Doubt {
                activation(P);
                for all positions Pj that refer to P
                        if (active(Pj) ∧ ¬link(Pj))
                        {       ¬activation(P);
                                break; }}
        case Preponderance of Evidence {
                score(P)=0;
                calculate_weights(P);
                for all positions Pj that refer to P
                        if (active(Pj) ∧ link(Pj))
                                score(P) += weight(Pj);
                        else if (active(Pj)∧ ¬link(Pj))
                                score(P) -= weight(Pj);
                if (score(P) ≥ 0) activation(P);
                        else ¬activation(P) }}
if (¬ old_activation==activation(P)) 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 in Section 3. In case of PoE, calculate_weights(P) calls topological sort to compute the weights of the supporting and counter-arguments of P.

```
update(position P) {
if (compute_activation(P)) {
        if (active(P)) activate(P);
              else inactivate(P);
        update(RefersTo(P)); }}          /* 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.

```
activate(position P) {
Pj=RefersTo(P);
for all constraints Cj that refer to Pj
        if (¬active(Cj) ∧ Cj has a preference relation on P)
                if (compute_Activation(Cj)) {
                        if (consistent_with_Previous(Cj)
                                consistent(Cj);
                        else ¬consistent(Cj); }}
```

Activation of positions involves finding the potential constraints where the position appears (these constraints must refer to the same position as P) and checking 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 in a preference relation 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. A number of additional internal acts were implemented for the activation-inactivation of constraints, update of issues (in which case there is only PoE), etc.

Moderator acts are invoked by the discussion mediator and may involve the assignment of *access rights* (or *importance weights*) to agents, change of the proof standard, etc. 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.

5 Related and future work

In this section we focus on previous, well-tried concepts and theories, attempting to account for how humans combine various types of reasoning during argumentative discourse. We aim at extracting the pros and cons of our approach, and motivation for further work.

Toulmin's argumentation theory [37] identified six components in the construction of an argument: *datum* (i.e., a fact or data), *warrant* (i.e., inference rule), *backing* (i.e., support of the warrant), *claim* (i.e., conclusion of the argument), *rebuttal* (i.e., objection or exception) and *qualifier* (similar to *score* in our system). Figure 8 illustrates how Toulmin's structure of arguments can be mapped onto ours. We view any component as a position (claims may also be alternatives) with the appropriate role in the discussion graph. Main weaknesses of Toulmin's theory are the lack of the appropriate formalism for addressing multiple issues and ordering competing arguments, and its orientation towards only cooperative-type argumentation. Instead, our framework is applicable to adversarial or cooperative group decision making, and provides a means of ordering positions and alternatives in an issue.

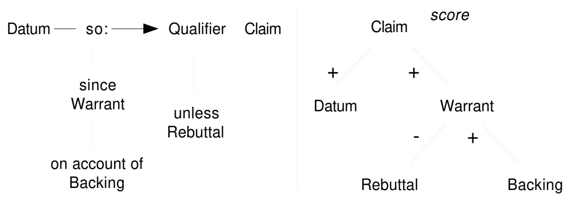

Fig. 8. Structure of an argument according to Toulmin's (left) and the proposed (right) model.

Pollock's OSCAR model of defeasible reasoning [26] was one of the first attempts to base defeasible reasoning on arguments, influencing later work (see for example [34], [14]). His model does not deal with resolving disputes, but with prescribing the set of beliefs a single rational agent should hold, under certain simplifying assumptions. According to him, the initial *epistemic basis* of an agent consists of a set of positions, either supported by perception or recalled from memory, a set of defeasible and a set of nondefeasible *inference rules*. Belief on positions stands until defeated by new reasons, disregarding the original ones. Pollock also distinguishes between *warranted* and *justified belief*; we deal with these issues using proof standards. A weakness of OSCAR, addressed in our framework, is that there are no reasoning schemata provided for the validity of inference rules, or their relative weight and priority.

Rescher, in his theory of *formal disputation* [30], considers disputation to be a three-party game, taking place with a *proponent*, asserting a certain position, an *opponent*, able to challenge the proponent's position, and a *determiner* which decides whether the proponent's position was successfully defended or not. Our model plays the last role, mediating the discussion among multiple agents. A more formal reconstruction of Rescher's theory has been presented by Brewka [7], based on Reiter's Default Logic [29]. His work clarifies Rescher's concepts and goes ahead defining *elementary* and *legal moves* during a dispute, as well as *winning situations*. Nevertheless, both approaches are limited in that the "players" have no chance to disagree about defaults.

Approaches to reasoning in the presence of inconsistency can be divided into those based on the notion of *maximal consistent subbase*, where coherence is restored by selecting consistent subbases, and those based on *argumentation principles*, where inconsistency is "accepted" by providing arguments for each conclusion [9]. Non-monotonic inference relations are basically used in the first category, defined by methods of generating *preferred belief subbases* and classical inference [5]. The second category, inspired from work done on AI and Law, views the validity and priority of belief bases as subject to debate. Throughout this paper, we have taken into account belief bases that have been explicitly stated by an agent.

In the kind of discussions our system addresses, domain theories are usually weak, i.e., rules and concepts may have not been expressed and, if there are, they often suffer from definitional flaws. To remedy this, future work directions include incorporation of an agent's *viewpoint* in the system, and integration of *Case-Based Reasoning (CBR)* methods. Viewpoints will model each agent's agenda, aiming at making clear (from the beginning) his interpretation of the issue under discussion. They will include desired attributes that each alternative should possess in order to "win" the issue, and the preference relations between them. Viewpoints will explicitly map the interests of each participant, and may evolve during the discourse. Integration of our system with CBR methods will increase its efficiency, by reusing previous instances (*cases*) of similar discussions [36], [35]. One will contrast precedent cases with the current one, reason about the similarities and differences with it, and finally argue why a previous interpretation can (or cannot) be applied. Issues involved are *case indexing*, *retrieval of relevant cases* from the case base, *selection of the most promising cases* to reason with, *adaptation of case* (construction of a solution or interpretation for the new case), and *update* of case base adjusting relevant indices (see also [31]).

Moreover, we need to implement norms or protocols about the rights and duties of the different types of agents that participate in a discussion. Ideas from similar structures in formalized public activities should be exploited together with approaches from AI and Law (i.e., Deontic Logic, Argumentation Theory) and Distributed AI. Protocols should take into account the roles of participants and the evolution of the decision making procedure.

6 Conclusion

The argumentation system is a part of a larger project that attempts to link argumentation with *collaboration* services in the context of Group Decision Making Applications. Collaboration services include a *shared workspace*, where agents can put any type of files (e.g. text, images) related to the argumentation, *navigation mechanisms* and appropriate *visualization tools*. In addition the system includes *information services* which facilitate information retrieval from internal (e.g., shared workspace) and external (e.g., the World Wide Web) resources. Several interesting problems (beyond the scope of this paper) related to the project, include *authentication* and *access rights, concurrency control* (in some cases, users may need to simultaneously modify different parts of the same object) and *data conversion* and *integration* issues.

Use of information technology may assist multi-agent decision making in various ways. An important goal is to provide easy access to the current knowledge, at any time. This would be greatly facilitated if all relevant data and documents are made available and maintained in electronic form, in a well-structured and organized way. Our system's primary task is to provide direct computer support for the argumentation, negotiation and mediation process in these environments. A computer network can be used as a medium for such a process, in which documents are indexed according to their role and function, using the presented model of argumentation. The system presented in this paper combines concepts from various well-established areas such as Decision Theory, Non-Monotonic Reasoning, Constraint Satisfaction and Cognitive Modeling. We intend it to act as an assistant and advisor, by recommending solutions and leaving the final enforcement of decisions and actions to agents.

Acknowledgements: The authors thank Gerhard Brewka, Tom Gordon, Brigitte Trousse and Hans Voss for helpful discussions on various topics of this paper. Nikos Karacapilidis is financed by the Commission of the European Communities through the ERCIM Fellowship Programme. Dimitris Papadias is supported by a DAG grant from Hong Kong RGC.

References

1. Batini, C., Ceri, S., Navathe, S.B.: *Conceptual Database Design: An Entity-Relationship Approach.* Benjamin/Cummings Publishing, 1992.
2. Bench-Capon, Tr.: Argument in Artificial Intelligence and Law. In J.C. Hage, T.J.M. Bench-Capon, M.J. Cohen and H.J. van den Herik (eds.), *Legal knowledge based systems - Telecommunication and AI & Law*, Koninklijke Vermande BV, Lelystad, 1995, pp. 5-14.
3. Benferhat, S., Dubois, D., Prade, H.: How to infer from inconsistent beliefs without revising? In *Proceedings of the 14th IJCAI*, Montreal, 1995, pp. 1449-1455.
4. Benferhat, S., Cayrol, C., Dubois, D., Lang, J., Prade, H.: Inconsistency Management and Prioritized Syntax-Based Entailment. In *Proceedings of the 13th IJCAI*, Chambery, 1993, pp. 640-645.
5. Brewka, G.: Preferred Subtheories: An extended logical framework for default reasoning. In *Proceedings of the 11th IJCAI*, Detroit, 1989, pp. 1043-1048.
6. Brewka, G.: Reasoning about Priorities in Default Logic. In *Proceedings of the 12th AAAI*, Seattle, 1994, pp. 940-945.
7. Brewka, G.: A Reconstruction of Rescher's Theory of Formal Disputation Based on Default Logic. In *Working Notes of the 12th AAAI Workshop on Computational Dialectics*, Seattle, 1994, pp. 15-27.
8. Brewka, G., Gordon, T.: How to Buy a Porsche: An Approach to Defeasible Decision Making. In *Working Notes of the 12th AAAI Workshop on Computational Dialectics*, Seattle, 1994, pp. 28-38.

9. Cayrol, C.: On the Relation between Argumentation and Non-monotonic Coherence-Based Entailment. In *Proceedings of the 14th IJCAI*, Montreal, 1995, pp. 1443-1448.
10. Conklin, E.J.: Capturing Organizational Memory. In D. Coleman (ed.) *Groupware '92*, 1992, pp. 133-137.
11. Dung, P.M.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning and logic programming. In *Proceedings of the 13th IJCAI*, Chambery, 1993, pp. 852-857.
12. Eemeren, F.H. van, Grootendorst, R., Snoeck Henkemans, F.: *Fundamentals of Argumentation Theory: A Handbook of Historical Backgrounds and Contemporary Developments.* Lawrence Erblaum Associates, Mahwah, New Jersey, 1996.
13. Farley, A.M., Freeman, K.: Burden of Proof in Legal Argumentation. In *Proceedings of the 5th Int. Conference on AI and Law*, 1995, pp. 156-164.
14. Geffner, H., Pearl, J.: Conditional Entailment: Bridging two Approaches to Default Reasoning. *Artificial Intelligence* 53 (2-3), 1992, pp. 209-244.
15. Gordon, T.: *The Pleadings Game: An Artificial Intelligence Model of Procedural Justice.* Kluwer, 1995.
16. Green, D.W.: Arguments and Mental Models. In *Proceedings of the Int. Conference on Formal and Applied Practical Reasoning*, 1996.
17. Heidrich. C.H.: Montague-grammars for Argumentative Dialogues. In Barth & Martens (Eds.) *Argumentation: Approaches to Theory Formation.* John Benjamins, Amsterdam, 1982.
18. Hurwitz, R., Mallery, J.C.: The Open Meeting: A Web-Based System for Conferencing and Collaboration. In *Proceedings of the 4th Int. WWW Conference*, 1995.
19. Karacapilidis, N.I.: Planning under Uncertainty: A Qualitative Approach. In C. Pinto-Ferreira and N.J. Mamede (eds.), *Progress in Artificial Intelligence*, LNAI 990, Springer-Verlag, 1995, pp. 285-296.
20. Karacapilidis, N.I., Papadias, D., Gordon, T.: An Argumentation Based Framework for Defeasible and Qualitative Reasoning. *Advances in Arificial Intelligence*, LNAI 1159, Springer-Verlag, Berlin, 1996, pp. 1-10.
21. Knuth, D. E.: *Fundamental Algorithms*, Art of Computer Programming, Vol.1, Addison-Wesley, Second Edition, 1973.
22. Kunz, W., Rittel, H.W.J.: Issues as Elements of Information Systems. Working Paper 131, Universität Stuttgart, Institut für Grundlagen der Plannung, 1970.
23. Lueken, G-L.: Incommensurability, Rules of Argumentation and Anticipation. In *Proceedings of the 2nd Int. Conference of Argumentation*, 1991.
24. Mackworth, A., Freuder, E.: The Complexity of some Polynomial Network Consistency Algorithms for Constraint Satisfaction Problems, *Artificial Intelligence* 25, 1985, pp. 65-74.
25. Pinkas, G.: Propositional Non-Monotonic Reasoning and Inconsistency in Symmetric Neural Networks. In *Proceedings of the 12th IJCAI*, Sydney, 1991, pp. 525-530.
26. Pollock, J.: Defeasible Reasoning. *Cognitive Science* 11, 1988, pp. 481-518.
27. Prakken, H.: Logical tools for modelling legal argument. Ph.D. Dissertation, Free University of Amsterdam, 1993.
28. Prakken, H.: From Logic to Dialectics in Legal Argument. In *Proceedings of the 5th International Conference on AI and Law*, ACM Press, 1995, pp. 165-174.
29. Reiter, R.: A Logic for Default Reasoning. *Artificial Intelligence* 13, 1980, pp. 81-132.
30. Rescher, N.: *Dialectics: A Controversy-Oriented Approach to the Theory of Knowledge.* State University of New York Press, Albany, 1977.

31. Rissland, E.L., Skalak, D.B.: CABARET: rule interpretation in a hybrid architecture. *Int. Journal of Man-Machine Studies* 34, 1991, pp. 839-88.
32. Rittel, H.W.J., Webber, M.M.: Dilemmas in a General Theory of Planning. *Policy Sciences* 4, 1973, pp. 155-169.
33. Sian, S.S.: Adaptation based on cooperative learning in multi-agent systems. In Y. Demazeau and J.P. Müller (eds.), *Decentralized AI 2*, Elsevier Science Publishers B.V., 1991, pp. 257-272.
34. Simari, G.R., Loui, R.P.: A Mathematical Treatment of Defeasible Reasoning and its Implementation. *Artificial Intelligence* 53 (2-3), 1992, pp. 125-157.
35. Sycara, K.: Resolving Goal Conflicts via Negotiation. In *Proceedings of the 7th AAAI*, Saint Paul, Minnesota, 1988, pp. 245-250.
36. Sycara, K.: Resolving Adversarial Conflicts: An Approach Integrating Case-Based and Analytic Methods. Ph.D. diss., School of Information and Computer Science, Georgia Institute of Technology, 1987.
37. Toulmin, S.E.: *The Uses of Argument.* Cambridge University Press, 1958.
38. Valesio, P.: *Novantiqua: Rhetorics as a Contemporary Theory.* Indiana University Press, 1980.
39. Yakemovic, K.C.B., Conklin, E.J.: Report on a Development Project Use of an Issue-Based Information System. In F. Halasz (ed.), *Proceedings of CSCW 90*, LA, 1990, pp. 105-118.
40. http://union.ncsa.uiuc.edu/HyperNews/get/www/collab/conferencing.html.