HAL
open science

The hedgehog and the fox An argumentation-based decision support system

M. Morge, P. Mancarella

To cite this version:

M. Morge, P. Mancarella. The hedgehog and the fox An argumentation-based decision support system. 2007. hal-00193405

HAL Id: hal-00193405
https://hal.science/hal-00193405v1

Preprint submitted on 3 Dec 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The hedgehog and the fox
An argumentation-based decision support system

Maxime Morge[†] Paolo Mancarella[†]
morge@di.unipi.it paolo@di.unipi.it

[†]Dipartimento di Informatica
University of Pisa
Largo B. Pontecorvo, 3 I-56127 Pisa, Italy

Résumé :

Nous présentons dans cet article un Système d'Aide à la Décision (SDA). À cette intention, nous proposons un cadre d'argumentation pour le raisonnement pratique. Celui-ci s'appuie sur un langage logique qui sert de stucture de données concrète afin de représenter les connaisances, les buts et les décisions possibles. Différentes priorités y sont associées afin de de représenter la fiabilité des connaisances, les préférences de l'utilisateur, et l'utilité espérée des alternatives. Ces structures de données constitue l'épine dorsale des arguments. De part la nature abductive du raisonnement pratique, les arguments sont construits à partir des conclusions. De plus, nous les définissons comme des structures arborescentes. De cette manière, notre SDA suggère à l'utilisateur les meilleures solutions et propose une explication interactive et compréhensible de ce choix.

Mots-clés : Intelligence artificielle, Raisonnement pratique, Argumentation

Abstract:

We present here a Decision Support System (DSS). For this purpose, we propose an Argumentation Framework for practical reasoning. A logic language is used as a concrete data structure for holding the statements like knowledge, goals, and actions. Different priorities are attached to these items corresponding to the reliability of the knowledge, the preferences between goals, and the expected utilities of alternatives. These concrete data structures consist of information providing the backbone of arguments. Due to the abductive nature of practical reasoning, we build arguments by reasoning backwards. Moreover, arguments are defined as tree-like structures. In this way, our DSS suggests some solutions and provides an interactive and intelligible explanation of the choices.

Keywords: Artificial Intelligence, Practical reasoning, Argumentation

1 Introduction

Decision making is the cognitive process leading to the selection of a course of action among alternatives based on estimates of the values of those alternatives. Indeed, when a human identifies her needs and specifies them with high-level and abstract terms, there should be a way to select an existing solution. Decision Support Systems (DSS) are computer-based systems that support decision making activities including expert systems and Multi-Criteria Decision Analysis (MCDA). In this paper, we propose a DSS which suggests some solutions and provides an interactive and intelligible explanation of the choices.

In this paper, we present our Decision Support System (DSS). This computer system is built upon an Argumentation Framework (AF) for decision making. For this purpose, we consider practical reasoning as the vehicle of decision making, which is a knowledge-based, goal-oriented, and action-related reasoning. A logic language is used as a concrete data structure for holding the statements like knowledge, goals, and actions. Different priorities are attached to these items corresponding to the reliability of the knowledge, the preferences between goals, and the expected utilities of alternatives. These concrete data structures consist of information providing the backbone of arguments. Due to the abductive nature of practical reasoning, arguments are built by reasoning backwards. Moreover, arguments are defined as tree-like structures. In this way, our DSS

suggests some solutions, as other classical approaches, but also provides an interactive and intelligible explanation of this choice.

Section 2 presents the principle and the architecture of our DSS. Section 3 introduces the walk-through example. In order to present our Argumentation Framework (AF) for decision-making, we will browse the following fundamental notions. First, we define the *object language* (cf Section 4). Second, we will focus on the internal structure of *arguments* (cf Section 5). We present in Section 6 the *interactions* amongst them. These relations allow us to give a declarative model-theoretic *semantics* to this framework and we adopt a *dialectical proof procedure* to implement it (cf Section 7). Section 8 draws some conclusions and directions for future work.

2 Principle and architecture

Basically, decision makers are categorized as either "hedgehogs", which know one big thing, or "foxes", which know many little things [1]. While most of the DSS are addressed to "hedgehogs", we want to provide one for both.

An "hedgehog" is an expert of a particular domain, who has intuitions and strong convictions. A "fox" is not an expert but she knows many different thinks in different domains. She decides by interacting with other and she is able to change her mind. Most of the DSS are addressed to "hedgehogs". These computer systems provide a way to express qualitative and/or quantitative judgements and synthesizes them to suggest an action. However the analytic skills needed for good judgments are those of foxes. We want to provide a DSS for the effective management of teams including both hedgehogs and foxes.

The current architecture of our DSS based upon an assistant agent. The mind of the agent relies upon an argumentative engine. The system only communicates with the users, i.e. the hedgehog and the fox, and the latter takes the final decision. On one side, the hedgehog informs the assistant agent in order to structure the decision making problem, to consider the different needs, to identify the alternatives, and to gather the required knowledge. On the other side, the fox can ask for a possible solution (*question*). The argumentative engine suggests some solutions (*assert*). The reasons supporting these admissible solutions can be interactively explored (*challenge/argue*).

3 Walk-through example

We consider here the decision making problem for selecting a suitable business location.

The assistant agent is responsible for suggesting some suitable locations, based on the explicit users'needs and on their knowledge. The main goal, that consists in selecting the location, is addressed by a decision, i.e. a choice between some alternatives (e.g. Pisa or London). The main goal (g_0) is split into sub-goals and sub-goals of these sub-goals, which are criteria for evaluating different alternatives. The location must offer a "good" regulation (g_1) and a "great" accessibility (g_2). These are abstract goals, revealing the user's needs. The knowledge about the location is expressed with predicates such as : Sea(x) (the location is accessible by sea transports), or Road(x) (the location is accessible by road transports).

Figure 1 provides a simple graphical representation of the decision problem called influence diagram [2]. The elements of the decision problem, i.e. *values* (represented by rectangles with rounded corners), *decisions* (represented by squares) and *knowledge* (represented by ovals), are connected by arcs where predecessors are inde-

pendent and affect successors. We consider here a multiattribute decision problem captured by a hierarchy of values where the abstract values (represented by rectangles with rounded corner and double line) aggregate the independent values in the lower levels. While the influence diagram displays the structure of the decision, the object language reveals the hidden details of the decision making.

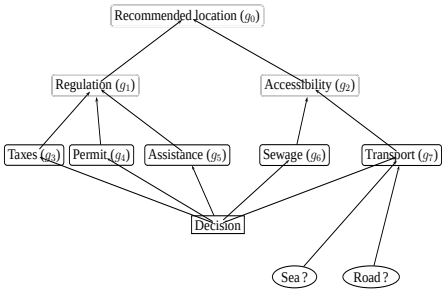

FIG. 1 – Influence diagram to structure the decision

4 The object language

Since we want to provide a computational argumentation model of practical reasoning and we want to instantiate it for our simple case study, we need to specify a particular logic.

The object language expresses rules and facts in logic-programming style. In order to address a decision making problem, We distinguish :

- a set of *goals*, i.e. some propositional symbols which represent the features that the decision must exhibit (denoted by $g_0, g_1, g_2, \ldots$) ;
- a *decision*, i.e. a predicate symbol which represents the action which must be performed (denoted by D) ;
- a set of *alternatives*, i.e. some constants symbols which represent the mutually exclusive solutions for the decision (e.g. pisa or london) ;
- a set of *beliefs*, i.e. some predicate symbols which represent epistemic statements (denoted by words such as Sea, or Road).

Since we want to consider conflicting goals, mutual exclusive alternatives, and contradictory beliefs in this object language, we need some form of negation. For this purpose, we consider strong negation, also called explicit or classical negation. Since we restrict ourselves to logic programs, we cannot express in a compact way the mutual exclusion between alternatives. For this purpose, we define the incompatibility relation (denoted by $\mathcal{I}$) as a binary relation over atomic formulas which is symmetric. Obviously, $L \ \mathcal{I} \ \neg L$ for each atom L, and $D(a_1) \ \mathcal{I} \ D(a_2)$, a_1 and a_2 being different alternatives.

Definition 1 (Theory) *A* theory $\mathcal{T}$ *is an extended logic program, i.e a finite set of rules of the form* $R : L_0 \leftarrow L_1, \ldots, L_n$ *with* $n \geq 0$, *each* L_i *being a strong literal. The literal* L_0, *called the* head *of the rule, is denoted by head*(R). *The finite set* $\{L_1, \ldots, L_n\}$, *called the* body *of the rule, is denoted by body*(R). *The body of a rule can be empty. In this case, the rule, called a* fact, *is an unconditional statement.* R, *called the name of the rule, is an atomic formula.*

In the theory, we distinguish :

- *goal rules* of the form $R : g_0 \leftarrow g_1, \ldots, g_n$ with $n > 0$. Each g_i is a goal. According to this rule, the head goal is reached if the goals in the body are reached ;
- *epistemic rules* of the form $R : B_0 \leftarrow B_1, \ldots, B_n$ with $n \geq 0$. Each B_i is a belief literal ;
- *decision rules* of the form $R : g \leftarrow D(a), B_1, \ldots, B_n$ with $n \geq 0$. The head of this rule is a goal and the body include a decision literal ($D(a)$)

and a possible empty set of belief literals. According to this rule, the goal can be eventually reached by the decision $D(a)$, provided that conditions $B_1, \ldots, B_n$ are satisfied.

Considering statements in the theory is not sufficient to take a decision, since all relevant pieces of information should be taken into account, such as the reliability of knowledge, the preferences between goals, or the expected utilities of the different alternatives. We consider that the *priority* $\mathcal{P}$ is a (partial or total) preorder on $\mathcal{T}$. $R_1 \mathcal{P} R_2$ can be read "R_1 has priority over R_2". $R_1 \not\mathcal{P} R_2$ can be read "R_1 does not have priority over R_2", either since R_1 and R_2 are *ex æquo* (denoted $R_1 \sim R_2$), i.e. $R_1 \mathcal{P} R_2$ and $R_2 \mathcal{P} R_1$, or since R_1 and R_2 are not comparable, i.e. $\neg(R_1 \mathcal{P} R_2)$ and $\neg(R_2 \mathcal{P} R_1)$.

In this work, we consider that all rules are potentially defeasible and that the priorities are extra-logical and a domain-specific features. The priority of concurrent rules depends of the nature of rules. Rules are *concurrent* if their heads are the same or incompatible. We define three priority relations :

- the priority over *goal rules* comes from their levels of *preference*. Let us consider two goal rules R_1 and R_2 with the same head g_0. R_1 has priority over R_2 if the achievement of the goals in the body of R_1 are more "important" than the achievement of the goals in the body of R_2 as far as reaching g_0 is concerned ;
- the priority over *epistemic rules* comes from their levels of *certainty*. Let us consider, for instance, two concurrent facts F_1 and F_2. F_1 has priority over F_2 if the first is more likely to hold than the second one ;
- the priority over *decision rules* comes from the *expected utilities* of decisions. Let us consider two rules R_1 and R_2 with the same head. R_1 has priority over R_2 if the expected utility of the first conditional decision is greater than the second one.

TAB. 1 – The goal theory

$R_{012} : g_0 \leftarrow g_1, g_2$
$R_{1345} : g_1 \leftarrow g_3, g_4, g_5$
$R_{267} : g_2 \leftarrow g_6, g_7$
$R_{145} : g_1 \leftarrow g_4, g_5$
$R_{01} : g_0 \leftarrow g_1$
$R_{13} : g_1 \leftarrow g_3$
$R_{26} : g_2 \leftarrow g_6$
$R_{02} : g_0 \leftarrow g_2$
$R_{14} : g_1 \leftarrow g_4$
$R_{27} : g_2 \leftarrow g_7$
$R_{15} : g_1 \leftarrow g_5$

TAB. 2 – The epistemic theory

$F_1 : \text{Road}(pisa) \leftarrow$
$F_2 : \text{Sea}(pisa) \leftarrow$
$F_3 : \neg\text{Road}(pisa) \leftarrow$

In order to illustrate the notions introduced previously, let us go back to our example. The goal rules, the epistemic rules, and the decision rules are represented in Table 1, Table 2, and Table 3, respectively. A rule above another one has priority over it. To simplify the graphical representation of the theories, they are stratified in non-overlapping subsets, i.e. different levels. The *ex æquo* rules are grouped in the same level. Non-concurrent rules are arbitrarily assigned to a level.

According to the goal theory in Table 1, the achievement of both g_4 and g_5 is required to reach g_1, but this constraint can be relaxed and the achievement of g_4 is more important than the achievement of g_5 to reach g_1. According to the epistemic theory in Table 2, the assistant agent does not know if London is accessible by sea/road transports. Due to conflicting sources of information, the agent has conflicting beliefs about the road accessibility of Pisa. Since these sources of information are more or less reliable, $F_1 \mathcal{P} F_3$.

TAB. 3 – The decision theory

$R_{32} : g_3 \leftarrow D(\text{pisa})$
$R_{41} : g_4 \leftarrow D(\text{london})$
$R_{51} : g_5 \leftarrow D(\text{london})$
$R_{71}(x) : g_7 \leftarrow D(x), \text{Sea}(x)$
$R_{31} : g_3 \leftarrow D(\text{london})$
$R_{42} : g_4 \leftarrow D(\text{pisa})$
$R_{52} : g_5 \leftarrow D(\text{pisa})$
$R_{61} : g_6 \leftarrow D(\text{london})$
$R_{62} : g_6 \leftarrow D(\text{pisa})$
$R_{72}(x) : g_7 \leftarrow D(x), \text{Road}(x)$

According to the decision theory in Table 3, Pisa has a greater expected utility than London to reach g_3. The expected utilities of these alternatives with respect to g_7 depends on the knowledge : a location accessible by sea is preferred than a location accessible by road ($R_{71}\ \mathcal{P}\ R_{72}$). We will build now arguments in order to compare the alternatives.

5 Arguments

In this Section, we define and construct arguments by reasoning backwards due to the abductive nature of the practical reasoning. Since we adopt a tree-like structure of arguments, our framework not only suggests some solutions but also provides an intelligible explanation of them.

In order to consider the recursive nature of arguments, we adopt and extend the tree-like structure for arguments proposed in [5].

Definition 2 (Argument) *An argument has a conclusion, top rules, premises, suppositions, and sentences. These elements are abbreviated by the corresponding prefixes. An argument A is :*

1. *a* supposal argument *built upon an unconditional ground statement.*
   If L is a ground literal such that there is no rule R in $\mathcal{T}$ which can be instantiated in such a way that $L = \mathtt{head}(R)$, then the argument, which is built upon this ground literal is defined as follows :

   $$\mathtt{conc}(A) = L, \mathtt{top}(A) = \emptyset,$$
   $$\mathtt{premise}(A) = \emptyset, \mathtt{supp}(A) = \{L\},$$
   $$\mathtt{sent}(A) = \{L\}.$$

   or

2. *a* trivial argument *built upon an unconditional ground statement.*
   If F is a fact in $\mathcal{T}$, then the argument A, which is built upon the ground instance F^g of F, is defined as follows :

   $$\mathtt{conc}(A) = \mathtt{head}(F^g),$$
   $$\mathtt{top}(A) = F^g,$$
   $$\mathtt{premise}(A) = \{\mathtt{head}(F^g)\},$$
   $$\mathtt{supp}(A) = \emptyset,$$
   $$\mathtt{sent}(A) = \{\mathtt{head}(F^g)\}.$$

   or

3. *a* tree argument *built upon an instantiated rule such that all the literals in the body are the conclusion of subarguments. If R is a rule in $\mathcal{T}$, we define the argument A built upon a ground instance R^g of R as follows. Let $\mathtt{body}(R^g) = \{L_1, \ldots, L_n\}$ and $\mathtt{sbarg}(A) = \{A_1, \ldots, A_n\}$ be a collection of arguments such that, for each $L_i \in \mathtt{body}(R^g)$, $\mathtt{conc}(A_i) = L_i$ (each A_i is called a subargument of A). Then :*

   $$\mathtt{conc}(A) = \mathtt{head}(R^g),$$
   $$\mathtt{top}(A) = R^g,$$
   $$\mathtt{premise}(A) = \mathtt{body}(R^g),$$
   $$\mathtt{supp}(A) = \cup_{A' \in \mathtt{sbarg}(A)} \mathtt{supp}(A'),$$
   $$\mathtt{sent}(A) = \cup_{A' \in \mathtt{sbarg}(A)} \mathtt{sent}(A') \cup \mathtt{body}(R^g).$$

The set of arguments built upon $\mathcal{T}$ is denoted $\mathcal{A}(\mathcal{T})$.

As in [5], we consider *atomic* arguments (2) and *composite* arguments (3). Moreover, we distinguish *supposal* arguments (1)

and *built* arguments (2/3). Due to the abductive nature of practical reasoning, we define and construct arguments by reasoning backwards. Therefore, arguments do not include irrelevant information such as sentences not used to prove the conclusion.

Contrary to the other definitions of arguments (pair of premises - conclusion, sequence of rules), our definition considers that the different premises can be challenged and can be supported by composite arguments. In this way, arguments are intelligible explanations. Triples of conclusions - premises - suppositions are simple representations of arguments. Let us consider the previous decision making example. Some of the arguments concluding g_7 are the following :

–

$$B^1 = \langle g_7, (D(\text{pisa}), \text{Sea}(\text{pisa})), ((D(\text{pisa}))\rangle;$$

–

$$B^2 = \langle g_7, (D(\text{pisa}), \text{Road}(\text{pisa})), ((D(\text{pisa}))\rangle;$$

–

$$A^1 = \langle g_7, (D(\text{london}), \text{Sea}(\text{london})), (D(\text{london}), \text{Sea}(\text{london}))\rangle;$$

–

$$A^2 = \langle g_7, (D(\text{london}), \text{Road}(\text{london})), (D(\text{london}), \text{Road}(\text{london}))\rangle.$$

The tree argument B^1 contains two subarguments : one supposal argument ($\langle D(\text{pisa}), \emptyset, (D(\text{pisa}))\rangle$) and one trivial argument ($\langle \text{Sea}(\text{pisa}), (\text{Sea}(\text{pisa})), \emptyset\rangle$). Due to their structure and their nature, arguments interact with one another.

6 Interactions amongst arguments

The interactions between arguments may come from their nature, from the incompatibility of their sentences, and from the priority relation between the top rules of built arguments. We examine in turn these different sources of interaction.

Since sentences are conflicting, arguments interact with one another. For this purpose, we define the attack relation. An argument attacks another argument if the conclusion of the first one is incompatible with one sentence of the second one.

Definition 3 (Attack relation) *Let A and B be two arguments. A* attacks *B (denoted by attacks (A, B)) iff $\texttt{conc}(A)\, \mathcal{I}\, \texttt{sent}(B)$.*

This attack relation, often called *undermining* attack, is indirect, i.e. directed to a "subconclusion". The attack relation is useful to build an argument which is an homogeneous explanation.

Due to the nature of argument, arguments are more or less hypothetical.

Definition 4 (Supposition size) *Let A be an arguments. The* size of suppositions *for A, denoted $\texttt{suppsize}(A)$, is defined such that :*

1. *if A is a supposal argument, then $\texttt{suppsize}(A) = 1$;*
2. *if A is a trivial argument, then $\texttt{suppsize}(A) = 0$*
3. *if A is a tree argument and $\texttt{sbarg}(A) = \{A_1, \ldots, A_n\}$ is the collection of subarguments of A, then $\texttt{suppsize}(A) = \sum_{A' \in \texttt{sbarg}(A)} \texttt{suppsize}(A')$.*

Since arguments have different natures (supposal or built) and the top rules of built arguments are more or less strong, they interact with one another. For this purpose, we define the strength relation.

Definition 5 (Strength relation) *Let A_1 be a supposal argument, and A_2, A_3 be two built arguments.*

1. *A_2 is stronger than A_1 (denoted $A_2 \ \mathcal{P}^{\mathcal{A}} \ A_1$);*
2. *If $(\texttt{top}(A_2) \ \mathcal{P} \ \texttt{top}(A_3)) \wedge \neg(\texttt{top}(A_3) \ \mathcal{P} \ \texttt{top}(A_2))$, then $A_2 \ \mathcal{P}^{\mathcal{A}} \ A_3$;*
3. *If $(\texttt{top}(A_2) \not\mathcal{P} \texttt{top}(A_3)) \wedge (\texttt{suppsize}(A_2) \leq \texttt{suppsize}(A_3))$, then $A_2 \ \mathcal{P}^{\mathcal{A}} \ A_3$;*

Since $\mathcal{P}$ is a preorder on $\mathcal{T}$, $\mathcal{P}^{\mathcal{A}}$ is a preorder on $\mathcal{A}(\mathcal{T})$. The strength relation is useful to choose (when it is possible) between homogeneous concurrent explanations, i.e. non conflicting arguments with the same conclusions.

The two previous relations can be combined to choose (if possible) between non-homogeneous concurrent explanations, i.e. conflicting arguments with the same conclusion or with conflicting conclusions.

Definition 6 (Defeats) *Let A and B be two arguments. A* defeats *B (written defeats (A, B)) iff :*

1. *attacks (A, B);*
2. *$\neg(B \ \mathcal{P}^{\mathcal{A}} \ A)$.*

Similarly, we say that a set S of arguments defeats an argument A if A is defeated by one argument in S.

By definition, two equally relevant arguments both defeat each other.

Let us consider our previous example. The arguments in favor of London (A^1 and A^2) and the arguments in favor of Pisa (B^1 and B^2) attack each other. Since the top rule of A^1 and B^1 (i.e. R_{71}) has priority over the top rule of A^2 and B^2 (i.e. R_{72}), and $\texttt{suppsize}(B^1) = \texttt{suppsize}(B^2) = 1$ and $\texttt{suppsize}(A^1) = \texttt{suppsize}(A^1) = 2$, B^1 (resp. A^1) defeats A^2 (resp. B^2) and B^1 is stronger than A^1. If we only consider these four arguments, the assistant suggest Pisa and justify it with the availability of sea transports. In this section, we have defined the interactions between arguments in order to give them a status. Determining whether a solution is ultimately suggested requires a complete analysis of all arguments and subarguments.

7 Semantics and procedures

We can consider our AF abstracting away from the logical structures of arguments and equip it with various semantics, which can be computed by dialectical proof procedures.

Given an AF, [3] defines the following notions of "acceptable" sets of arguments :

Definition 7 (Semantics) *An AF is a pair $\langle \mathcal{A}, \text{ defeats } \rangle$ where $\mathcal{A}$ is a set of arguments and defeats $\subseteq \mathcal{A} \times \mathcal{A}$ is the defeat relationship*[1] *for AF. For $A \in \mathcal{A}$ an argument and $S \subseteq \mathcal{A}$ a set of arguments, we say that :*

- *A is* acceptable *with respect to S (denoted $A \in \mathcal{S}^S_{\mathcal{A}}$) iff $\forall B \in \mathcal{A}$, defeats (B, A) $\exists C \in S$ such that defeats (C, B);*
- *S is* conflict-free *iff $\forall A, B \in S \neg$ defeats (A, B);*
- admissible *iff S is conflict-free and $\forall A \in S, \ A \in \mathcal{S}^S_{\mathcal{A}}$;*

The admissible semantics sanctions a set of arguments as acceptable if it can successfully dispute every arguments against it, without disputing itself. However, there might be several conflicting admissible sets. Since a DSS involves an ultimate choice of the user between various admissible set of alternatives, we adopt this semantics. The decision $D(a_1)$ is *suggested* iff $D(a_1)$ is a supposition of one argument

[1] Actually, in [3] the defeat relation is called attack.

in an admissible set. Let us focus on the goal g_6 in the previous example, i.e. on the following theory $\mathcal{T} = \{R_{62}, R_{61}\}$. Since $\{A^3 = \langle g_6, (D(\text{london})), (D(\text{london}))\rangle\}$ and $\{B^3 = \langle g_6, (D(\text{pisa})), (D(\text{pisa}))\rangle\}$ are both admissible, Pisa and London must be suggested as different alternatives to reach g_6.

Since our practical application requires to specify the internal structure of arguments, we adopt the procedure proposed in [4] to compute admissible arguments. If the procedure succeeds, we know that the argument is contained in a preferred set.

We have implemented our AF, called MARGO[2] (Multiattribute ARGumentation framework for Opinion explanation). For this purpose, we have translated our AF in an assumption-based AF (ABF for short). CaSAPI[3] computes the admissible semantics in the ABF by implementing the procedure proposed in [4]. Moreover, we have developed a CaSAPI meta-interpreter to relax constraints on the goals achievements and to make suppositions in order to compute the admissible semantics in our concrete AF. In this section, we have shown how arguments in the framework can be categorized in order to suggest some solutions.

8 Conclusions

In this paper we have presented a DSS which suggests some solutions and provides an interactive and intelligible explanation of these choices. For this purpose, we have proposed and implemented a concrete AF for some applications of practical reasoning. A logic language is used as a concrete data structure for holding the statements like knowledge, goals, and actions. Different priorities are attached to these items corresponding to the reliability of the knowledge, the preferences between goals, and the expected utilities of alternatives. These concrete data structures consist of information providing the backbone of arguments. Due to the abductive nature of practical reasoning, arguments are built by reasoning backwards. To be intelligible, arguments are defined as tree-like structures. Due to their nature, the incompatibility of their sentences, and the priority relation between the top rules of built arguments, the arguments interact with one another. Since a DSS involves an ultimate choice of the user between various admissible set of alternatives, we have adopted an admissible semantics. Future investigations must explore how this proposal scales to drive argumentation-based negotiations.

[2] https ://margo.sourceforge.net/

[3] http ://www.doc.ic.ac.uk/∼dg00/casapi.html

9 Acknowledgements

This work is supported by the Sixth Framework IST programme of the EC, under the 035200 ARGUGRID project. We would like to thank the anonymous reviewers for their detailed comments on this paper.

Références

[1] Sir Isaiah Berlin. *The Hedgehog and the Fox*. Simon & Schuster, 1953.

[2] Robert Taylor Clemen. *Making Hard Decisions*. Duxbury. Press, 1996.

[3] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2) :321–357, 1995.

[4] Phan Minh Dung, Paolo Mancarella, and Francesca Toni. A dialectic procedure for sceptical, assumption-based argumentation. In *Proc. of COMMA*. IOS Press, 2006.

[5] Gerard Vreeswijk. Abstract argumentation systems. *Art. Intel.*, 90(1-2) :225–279, 1997.