JOURNAL OF MULTI-CRITERIA DECISION ANALYSIS
J. Multi-Crit. Decis. Anal. **18**: 289–309 (2011)
Published online 20 November 2011 in Wiley Online Library
(wileyonlinelibrary.com) DOI: 10.1002/mcda.486

Towards Automating Decision Aiding Through Argumentation

WASSILA OUERDANE[a], YANNIS DIMOPOULOS[b*], KONSTANTINOS LIAPIS[c] and PAVLOS MORAITIS[d]
[a] *LGI, Ecole Centrale Paris, Paris, France*
[b] *Department of Computer Science, University of Cyprus, Nicosia, Cyprus*
[c] *Department of Economic and Regional Development, Panteion University, Athens, Greece*
[d] *LIPADE, Paris Descartes University, Paris, France*

ABSTRACT

Decision aiding can be abstractly described as the process of assisting a user/client/decision maker by recommending possible courses of his action. This process has to be able to cope with incomplete and/or inconsistent information and must adapt to the dynamics of the environment in which it is carried out. Indeed, on the one hand, complete information about the environment is almost impossible, and on the other hand, the information provided by the user is often affected by uncertainty; it may contains inconsistencies and may dynamically be revised because of various reasons. The aim of this paper is to present a model of the decision aiding process that is amenable to automation. The main features of the approach is that it models decision aiding as an iterative defeasible reasoning process, and it uses argumentation for capturing important aspects of the process. More specifically, argumentation is used for representing the relations between the cognitive artefacts that are involved in decision aiding, as well as for modelling the artefacts themselves. In modelling the cognitive artefacts, we make use of the notion of argument schemes and specify the related critical questions. More specifically, the work reported here aims at initiating a systematic study of the use of argumentation in future decision aiding tools. Our ambition is twofold: (i) enhance decision support capabilities of an analyst representing explicitly and accountably the reasons for which he recommend a solution for a decision maker and (ii) enhance decision support capabilities of an (semi) automatic device to handle (at least partially) the dialogue with the user. Copyright © 2011 John Wiley & Sons, Ltd.

KEY WORDS: argumentation; decision aiding; artificial intelligence

1. INTRODUCTION

Decision analysis (Belton and Stewart, 2002; Bouyssou *et al*., 2000; French, 1988) is concerned with the process of providing decision support to 'clients' who feel unable to handle alone a problem situation. We call such an activity 'decision aiding'. Decision aiding is a process characterized by the emergence of cognitive artefacts, resulting from the interaction between the 'client' and the 'analyst'. The decision analyst and the client are engaged in an iterative process, where the analyst attempts, through successive steps of interaction with the client, to obtain a better understanding of the problem the client is facing. To be able to cope with the complexity of both the real world and the needs of the client, the analyst needs to make assumptions and reason as if these assumptions were true in the world. The recommendations, which are the outcomes of the decision aiding process (DAP), are subject to the client validation. Rejection of the recommendations means that some of the assumptions made by the analyst are false and must be retracted.

On the contrary, systems that aim at assisting people in decision making help the user to shape a problem situation, formulate a problem and possibly try to establish a viable solution to it. Decision theory and Multiple-Criteria Decision Analysis have established the theoretical foundation upon which many decision support systems have blossomed. These approaches (and the formal tools coming along with them) have focused for a long time on how a 'solution' should be established. But it is clear that the process involves many other aspects that are handled more or less formally by the analyst. For instance,

- The problem of *accountability* of decisions is almost as important as the decision itself. The decision maker should then be convinced by a proper

*Correspondence to: Department of Computer Science, University of Cyprus, Nicosia, Cyprus.
E-mail: yannis@cs.ucy.ac.cy

Received 26 February 2010
Accepted 9 September 2011

explanation that the proposed solution is indeed the best (see Belton and Stewart, 2002; Bouyssou *et al*., 2000).
- It should be possible, for the client, to *refine*, or even *contradict*, a given recommendation. Decision–support processes often need to be constructive, in the sense that a user may revise the assumptions or other aspect of the problem description when the potential solutions and their implications become explicit.

Nowadays, decision-aiding situations are pervasive: they can occur in situations where the role of the analyst is taken by a non-expert, in some extreme cases even by an automatic tool. For instance, consider the following scenarios:

- Ann is not an experienced analyst, but she has good knowledge of some decision–support tools that she herself used quite often. She would like to help Bob to make a decision regarding some public policy investment. In this situation, Ann may find useful to have the support of a tool that would provide her with explicit explanations, justifications and possible replies that could occur in the course of an interaction with her 'client'. Similarly, such a system could be used for the non-expert analyst to practice and simulate some virtual interactions with a client.
- Bob is purchasing items on the Internet. He has to choose among a selection of 150 digital cameras on a commercial website (too many to be examined exhaustively). Bob first provides some preferential information to the system. On the basis of the responses of Bob to these questions, the *recommender system* selects a specific model. Bob, not fully satisfied or convinced by the recommendation, would like to interact with the system, at least to gain a better understanding of the reasons underlying this. Such needs has been identified by mainstream recommender systems (Chen and Pu, 2007) but is only very simply addressed. For instance, it is now possible to check *why* a given item has been recommended by Amazon and to contradict the relevance of a certain purchase act for forthcoming recommendations.

The aforementioned scenarios mean that several aspects usually delegated to the human analyst should (in these situations) ideally be handled by the decision–support system. The task is ambitious: in a 'human-to-human' interaction—even though the dialogue is possibly supported by standard protocols (as in the case of constructing a value or an utility function or assessing importance parameters) that fix some explicit formal rules on how such a process can be conducted—the dialogue is handled through typical human interaction. A tool should be able to *structure* the dialogue on a formal basis in order to be able to control and assess what the artefact concludes as far as the user preference models are concerned and what type of recommendations (if any) is going to reach. In short, we need on the one hand some formal theory about preferences (and this is basically provided by decision analysis), and on the other hand some formal language enabling to represent the dialogue, to explain it, to communicate its results, to convince the user/decision maker that what is happening is both theoretically sound and operationally reasonable.

Although, in the decision analysis literature, there was until recently very little attention to the use of decision theories and decision aiding methodology when the interaction occurs between a human (a user) and an automatic device (see Klein, 1994 for a noticeable exception), the recent surge of automatic decision aiding tools on the Internet (recommender systems) have motivated a great deal of research. For instance, there are studies on the impact of higher levels of interaction with the user or explanation capabilities on the efficiency of the recommendations (Pu and Chen, 2007). Because of the context however, only very simple interactions and models of preferences are envisaged (a typical consumer is not prepared to enter in a long preference elicitation process or to discuss endlessly the benefits of a given option as opposed to another one).

At the same time, there is a long tradition in artificial intelligence (AI), going back to the early work of Simon, to challenge some assumptions of decision theory models or to emphasize their limits in certain circumstances. Stimulated by the objective to design agents capable of autonomous decision-making abilities (think of a robot exploring planet Mars), AI researchers pointed out the need to deal with missing or incomplete information, to revise some objectives to adapt to the new contexts and so on. In particular, the *knowledge representation* trend of AI has greatly contributed to challenge and question the rather crude 'utility' models used in decision theory. Indeed, one of the key distinctive ingredient of many AI-based approaches is to represent decision making in terms of 'cognitive attitudes' (as exemplified in the famous Belief-Desire-Intention paradigm) (Dastani *et al*., 2005; Doyle and Thomason, 1999), instead of mere

DOI: 10.1002/mcda

utilities (as already elicited by the analyst). This change of perspective paved the way for more *flexible* decision-making models: goals may change with circumstances, and understanding these underlying goals offers, for instance, the opportunity to propose alternative actions. The approach is attractive as it offers a natural and powerful way to specify agents' preferences. This is because it naturally caters for partial specification of preferences and makes explicit many aspects that are usually somewhat hidden in decision models.

A second, very influential contribution of AI, related to the previous point, has been the development of techniques for reasoning in the presence of conflicting (possibly heterogeneous) information. As the DAP is based on retractable assumptions, the formal modelling language to be used must be a non-monotonic one. Among the several possibilities, it seems that argumentation is particularly well suited for this task. Indeed, recently, following some early works (Bonet and Geffner, 1996; Fox and Parsons, 1997), several models have been put forward in the AI community that makes use of argumentation techniques (Amgoud, 2009; Amgoud *et al*., 2005; Atkinson *et al*., 2006; Labreuche, 2006; Ouerdane *et al*., 2007, 2009) to tackle decision problems. Such approaches have identified a variety of argument structures allowing to highlight the benefits of argumentation for decision: expressiveness and explicit representation of reasoning steps. Thus, they have greatly extended our understanding of the construction of argument for action or decision.

Moreover, the use of argumentation in decision support systems has been ever increasing. Such systems aim at assisting people in decision making. The need to introduce arguments in such systems has emerged from the demand to justify and to explain the choices and the recommendations provided by them. Together with this, other needs have motivated the use of argumentation, such as dealing with incomplete information, qualitative information and uncertainty (Amgoud and Prade, 2006; Fox *et al*., 1993; Parsons and Greene, 1999). Such systems deal with different contexts and applications, which may involve very different types of decision makers, from experts (medical domains) (Atkinson *et al*., 2006; Shankar *et al*., 2006) to potential buyers (recommender systems) (Chesñevar and Maguitman, 2004; Chesñevar *et al*., 2004, 2006) or simple citizens (public debate) (Atkinson, 2006; Morge, 2004), and even largely autonomous pieces of software that should act on behalf of a user (multi-agent systems) (Kakas and Moraitis, 2003; Parsons and Jennings, 1998; Sillince, 1994).

It is important to note that many of the aspects of decision-support systems discussed previously touch upon issues that have been long identified in system design. Indeed, it was about four decades ago when researchers such as Rittel (Rittel and Webber, 1973) came to realize that in order to tackle ill-defined problems (as opposed to the well-defined problems of science), an 'argumentative approach' was needed. This initiated the *design rationale* movement that advocates the thesis that 'to understand why a system design is the way it is, we also need to understand how it could be different, and why the choices which where made are appropriate' (MacLean *et al*., 1989). The rationale for a system describes the decision that have taken the possible alternatives that were considered and the pros and cons of these alternatives. Not surprisingly, many design rationale systems, starting with the early IBIS (Issue-Based Information System) (Kunz and Rittel, 1970) system to the more recent SEURAT (Software Using RATionale system) (Burge and Brown, 2008), are based on some form of argumentation.

The work presented here is in accordance with previous studies that employ argumentation in decision making but from a different perspective. First, the approach described here is not based on cognitive attitudes and the underlying motivation of the decision maker, but it relies on information provided by the decision maker during a Decision Aiding Process (DAP). Second, decision aiding is understood as the process of constructing and revising cognitive artefacts, which gradually transform an abstract problem description to an invocation of a concrete decision support tool. Therefore, automating the DAP amounts to automating the construction of these cognitive artefacts taking also into account the defeasible character of the process. Argumentation is the language that captures the interdependencies between these artefacts and controls the overall process.

The specific approach to artefact construction that is taken in this work is not intended to provide a fully automated and general approach to decision aiding, as it is limited in several ways. Initially, it is restricted to the modelling of specific cognitive artefacts involved in the Decision Aiding Process, leaving out highly abstract cognitive tasks such as the representation of a problem situation. Furthermore, the construction of the artefacts is a process where *predefined* 'components' of a decision–support system are put together to make a meaningful whole. Argumentation maintains a *high-level control* of this synthetic process and enables, instead of a static 'composition' of the

DOI: 10.1002/mcda

elements of the system at design time, a more dynamic one that can be decided at run-time, through the interaction with the environment or the user. Therefore, instead of building a rigid decision aiding tool, argumentation offers the possibility of delivering different, context-dependent versions of such a tool.

Furthermore, we note that even within the limited scope of the method that is described in this work, it is not always easy or even possible to completely automate this process because on the one hand, it is not always easy to model the decision aiding methodology, and the expertise of the analyst in the decision problem is considered. On the other hand, it is also not always easy or obvious to identify all needs and information necessary to fully meet the expectations of the user. In such cases, the approach can be seen as a method of building tools that support the DAP, which may help both the analyst and the user to share a common representation of the problem and the proposed solutions.

The proposed approach aims at facilitating the development of automatic decision devices and the improvement of decision support and recommender systems. In fact, the diffusion of Web-based services pushed the development of online decision support and decision support and recommender systems for a large variety of applications (e-commerce, e-voting, e-services, semantic Web, etc.). Such tools have to combine traditional decision making methods with flexible reasoning procedures allowing to handle the large variety of tasks required, to be adaptable to the changing environment where they operate and to perform self-improvement. Therefore, automating the DAP, whenever this is possible, by using argumentation is a first step towards meeting these needs. Additionally, argumentation can be used to provide design rationale information to future users and developers.

In summary, this paper studies two different ways of employing argumentation in decision aiding. First, we show how the relation between different artefacts of the DAP can be modelled using the framework of Kakas and Moraitis (2003), which is dynamic in the sense that the arguments, and their strength depend on the particular context that the decision maker (or agent) finds himself, thus allowing the agent to adapt his decisions in a changing environment. The decision aiding theories can be easily implemented directly from their declarative specification in the Gorgias system (Gorgias, 2002) for this framework. We focus on the inferences that can be drawn by argumentation and the way these inferences can be retracted in the light of the new information, capturing in this manner the dynamics of the DAP.

Second, we investigate how argumentation can model crucial aspects of each artefact of the DAP. More specifically, we study how *argument schemes* can be developed and used in the DAP. Argument schemes are presented as general inference rules whereby given a set of premises, a conclusion can be drawn. However, such schemes are not deductively strict because of the defeasible nature of arguments (Norman *et al.*, 2003; Walton, 1996, 2005). The schemes allow for arguments to be represented within a particular context and take into account the fact that the underlying reasoning may be altered in the light of new evidence or exceptions to rules.

The paper is structured as follows. In Section 2, first we introduce the concept of the DAP and the cognitive artefacts it produces. Then, we explain, by means of an example, what is missing in such a process. In Section 3, we briefly introduce the argumentation framework we use and show how it can capture the relations between the cognitive artefacts. Section 4 presents how the notion of arguments schemes, and their related critical questions, can be used to represent the steps of a multicriteria evaluation process. Finally, Section 5 provides the conclusion.

2. THE DECISION AIDING PROCESS

Decision aiding is an activity occurring in the everyday life of almost everybody. In this paper, we are interested in that particular type of decision aiding where formal and abstract languages are used (different decision theories and approaches). A DAP is a particular type of decision process involving at least two actors: a client, who himself is involved in at least one decision process (the one generating the concern for which the aid is requested), and the analyst, who is expected to provide the decision support. The aim of this particular process is to establish a shared representation of the client's concern, using the analyst's methodological knowledge, a representation enabling to undertake an action towards the concern.

2.1. Cognitive artefacts

Although decision aiding is a distributed process of cognition, we will present this concept using an operational approach based on the identification of the cognitive artefacts of the process (the outcomes or deliverables) (for more details, the reader is referred to the studies of Bouyssou *et al.*, 2000, and Tsoukiàs, 2007, 2008). The outcomes of this process are as follows:

DOI: 10.1002/mcda

- a representation of the problem situation: $\mathcal{P}$;
- the establishment of a problem formulation: Γ;
- the construction of an evaluation model: T;
- the establishment of a final recommendation: Φ.

In this paper, we will focus on the establishment of Γ and the construction of T. Our interest in this part of the process is because both these artefacts represent the easier to formalize and more structured outcomes of the process. Therefore, they can be easily modelled using a formal language. Two points should be considered:

- although these two artefacts appear subsequent, they are constructed through continuous interactions;
- the way the DAP is conducted influences the process outcomes.

We can now go through more details as far as these two artefacts are concerned.

2.1.1. Problem formulation (Γ). For a given representation of the problem situation, the analyst might propose to the client one or more 'problem formulations'. This is a crucial point of the DAP. The representation of the problem situation has a descriptive (at the best explicative) purpose. The construction of the problem formulation introduces what we call a model of rationality. A problem formulation reduces the reality of the decision process, within which the client is involved, to a formal and abstract problem. The result is that one or more of the client's concerns are transformed to formal problems on which we can apply a method (already existing, adapted from an existing one or created ad hoc) of the type studied in decision theory. From a formal point of view, a problem formulation is a triplet $\Gamma =< \mathbb{A}, \mathbb{V}, \Pi >$ where

- $\mathbb{A}$ is the set of potential actions the client may undertake within the problem situation as represented in $\mathcal{P}$. It should be noted that these are not 'given' but have to be constructed. A typical situation is the refinement of abstract options to more precise actions. A does not necessarily have a formal structure.
- $\mathbb{V}$ is the set of points of view under which the potential actions are expected to be observed, analysed, evaluated, compared etc., including different scenarios for the future.
- Π is the problem statement, the type of application to perform on the set A, an anticipation of what the client expects.

A problem statement can be operational or not (such as describing or conceiving the elements of $\mathbb{A}$). Operational problem statements are partitioning operations to be applied on the set A within the evaluation model $\mathcal{M}$ (see below). As such they can partition the set A:

- in predefined categories (large-medium-small, illness (A)-illness(B)) or in categories to be inferred comparing the elements of A;
- in ordered categories (accepted-rejected, bad-medium-good) or unordered categories (greens-blacks, monkeys-elephants).

Therefore, Π can be a choice statement (ordered and not predefined categories), a ranking (ordered and not predefined categories), a classification (predefined and not ordered categories), a clustering (no predefined and not ordered categories) etc. (for details see Bana a Costa, 1996; Tsoukiàs, 2007).

2.1.2. Evaluation model ($\mathcal{M}$). The term evaluation model refers to the decision aiding models as they are conceived in operational research, decision theory or AI methods. Classic decision aiding approaches focus on the construction of this model and consider the problem formulation as given. An evaluation model is a tuple $\mathcal{M} =< A, D, H, U, R >$, where

- A is a precise set of alternatives or decision variables on which the model will apply; A has a precise structure: enumeration of actions, domain of real numbers, combinatorial structure etc.;
- D is a set of dimensions (attributes) under which the elements of A are observed, measured, described etc.; a scale is always associated to each element of D;
- H is a set of criteria (if any) under which each element of A is evaluated in order to take into account the client's preferences;
- U is an uncertainty structure;
- R is a set of operators (aggregation functions) such that it is possible to obtain a comprehensive relation and/or function on A, possibly allowing to infer a final recommendation.

We emphasize the different use of terms such as goals (or objectives), attributes and criteria. Goals (objectives) represent 'desired states of the world' and are implicitly or explicitly considered while constructing the set A (for instance, through the description of alternative plans enabling to achieve a certain task or through the composition of different investment options in order to satisfy a portfolio construction). Attributes and criteria instead represent the fact that achieving a goal or an objective is not only a feasibility problem but also a preferability one. When this is the case, it is necessary to describe the consequences of potential actions along different dimensions (establishing attributes) and to evaluate the client's preferences on some (possibly all) of such consequences (establishing criteria).

2.2. Conducting the process

The DAP is the result of a dialogue between an analyst and a decision maker. During this process, the four artefacts may evolve, change and undergo revisions. Moreover, because a DAP always refers to a decision process that has a time and space extension, it is natural that the outcomes of the DAP remain *defeasible cognitive artefacts* in the sense that new information, beliefs and values may invalidate them and require an update or a revision.

Going back to the model of DAP, we present example 1 that serves the purpose of illustrating possible changes, revisions or updates associated with the formulation problem and its corresponding evaluation model during a DAP.

Example 1. (Bouyssou *et al*., 2006)
A client looking for decision support within a problem situation described as 'the client's bus company is looking for a bus'. He presents a set of offers received from several suppliers, each offer concerning a precise type of bus. The analyst will establish a problem formulation in which

- $\mathbb{A}$ is the list of offers received;
- $\mathbb{V}$ is the list of point of view that are customary in such cases, let us say cost, quality and transportation capacity;
- Π is a choice problem statement (an offer has to be chosen).

It is possible to construct an evaluation model with such information in which

- A are the feasible offers;
- D are the dimensions on which the offers are analysed: price and management costs, technical features (for the quality point of view) etc.;
- H are the criteria that the client agrees to use in order to represent these preferences;
- there is no uncertainty;
- R can be a multi-attribute value function, assuming that the client is able to establish the marginal value function on each criterion.

When this model is presented to the client, his reaction could be 'in reality we can buy more than one bus and there is no reason that we should buy two identical buses, since these could be used for different purposes such as long range leisure travels or urban school transport'. With such information, it is now possible to establish a new evaluation model in which

- A are all pairs of feasible offers;
- D are the dimension under which the offers are analysed (price, management costs, technical features etc.) but now concerning pairs of offers plus a classification of the buses in categories (luxury liner, mass transit, etc.);
- H are the same as previously plus a criterion about 'fitting the demand' because two different types of buses may fit the demand better;
- now, uncertainty is associated to the different scenarios of bus use;
- R can be multi-attribute utility function, provided that the client is able to establish the marginal value function on each criterion.

The process may continue revising models and problem formulations until the client is satisfied.

Note that it is necessary to update the contents of different models as the DAP involves in time and space. When confronted with a result, the decision maker realized that the model is not exactly what he expected. Therefore, he makes changes or gives new information in order to adapt the model to his needs. The consequence of this update is that the two models should be revised, namely the problem formulation and the evaluation model.

Moreover, the outcome of decision aiding is a recommendation Φ that is submitted to the user. There are three possibilities for this:

- Φ_1 the recommendation is validated and implementable
- Φ_2 the recommendation is validated but fails to be implemented
- Φ_3 the recommendation is not validated

The way the recommendation is submitted to the user is out of the scope of this paper.

The user therefore receives a pair $\langle \Phi_j, T \rangle$ where

DOI: 10.1002/mcda

- Φ_j represents the state of the recommendation with the user;
- T represent the reasons for which the recommendation is in such a state.

In case the recommendation is in state Φ_1, then the reasons in T are the overall appreciation of the user. Possible reasons for a recommendation in state Φ_2 or Φ_3 can be (i) no feasible solution in A satisfies the user or the recommendation is no more feasible; (ii) the available measures of elements in A are considered irrelevant, erroneous or affected by too large uncertainty; (iii) the preference models applied on A are not reliable and the user feels not to be correctly represented; (iv) uncertainty is misrepresented; or (v) the aggregation procedure is revealed to be meaningless or irrelevant to the user.

In such situations, different questions can be asked: *how to construct such reasons to be meaningful in the decision context considered? How to identify the problem? Or how to provide the consequences to the decision maker of a modification or a changes? Is it possible to challenge the aggregation procedure and how to update it? etc*. Thus, decision aiding is more than simply solving a complex decision model more or less faithful to the decision maker's values and preferences. It involves understanding, interpreting, justifying, explaining, convincing, revising and updating the outcomes of what we call a DAP. Currently, the model of DAP provides a rich theoretical framework in terms of aggregation of preferences and constructing recommendation for various decision problems. However, from a practical point of view, it offers little about how such activities are formally represented. We might be interested to establish a formal representation of all such activities for at least two reasons:

- enhance decision support capabilities of the analyst representing explicitly and accountably the reasons for which he recommend (or not) a solution (if any);
- enhance decision support capabilities of an (semi) automatic device to handle (at least partially) the dialogue with the user.

This work addresses these needs by relying on the concepts and tools of argumentation theory.

3. ARGUMENTATION AND ARTEFACT DEPENDENCIES

We have seen in the previous example that different version of the cognitive artefacts can be established during the DAP. These different versions are because client does not know how to express clearly, at the beginning of the process, what is his or problem and what are his her preferences. So, as the model is constructed, the decision maker may revise and update his preferences and/or objectives. However, such different versions are strongly related to each other because they carry essentially the same information and only a small part of the model has to be revised (Bouyssou *et al*., 2006; Tsoukiàs, 2007). The problem that arises here is that this revision (or update) must be taken into account by the model. In other words, there is a need for a formal representation of how the evolution occurs between different versions.

In the following, we discuss an argumentation framework that captures the dependencies between the artefacts and illustrate its working by means of an example.

3.1. The argumentation framework

This section gives the basic concepts of the underlying argumentation framework in which an agent represents and reasons with its decision aiding theory. This framework was proposed in the study conducted by Kakas *et al*. (1994) and developed further in that of Kakas and Moraitis (2003), in order to accommodate a dynamic notion of priority over the rules (and hence the arguments) of a given theory.

In this framework, (the components of) an agent theory is layered in three levels. *Object-level decision rules* are defined at the first level. The next two levels describe priority rules on the decision rules of the first level and on themselves thus expressing a preference policy for the overall decision making of the agent. This policy is separated into two levels: level 2 to capture the *default* preference policy under normal circumstances, whereas level 3 is concerned with the *exceptional* part of the policy that applies under specific contexts. The argumentation-based decision making will then be sensitive to context changes.

In general, an argumentation theory is defined as follows.

Definition 3.1

A theory is a pair $(\mathcal{T}, \mathcal{P})$. The sentences in $\mathcal{T}$ are propositional formulae, in the background monotonic logic $(\mathcal{L}, \vdash)$ of the framework, defined as $L \leftarrow L_1, \ldots, L_n$, where $L, L_1, \ldots, L_n$ are positive or explicit negative ground literals. Rules in P are the same as in T apart from the fact that the head L of the rules has the general form $L = h\_p(\text{rule1}, \text{rule2})$, where rule1 and rule2 are ground functional terms that name any two rules in the theory. This higher priority relation given by h_p is required to be irreflexive. The derivability

DOI: 10.1002/mcda

relation, $\vdash$, of the background logic is given by the single inference rule of modus ponens.

For simplicity, it is assumed that the conditions of any rule in the theory do not refer to the predicate h_p thus avoiding self-reference problems. For any ground atom $h\_p(\text{rule1}, \text{rule2})$, its negation is denoted by $h\_p(\text{rule2}, \text{rule1})$ and vice versa.

An *argument* for a literal L in a theory $(\mathcal{T}, \mathcal{P})$ is any subset, T, of this theory that derives L, i.e. $T \vdash L$ under the background logic. The subset of rules in the argument T that belong to $\mathcal{T}$ is called the *object-level* argument. Note that in general, we can separate out a part of the theory $\mathcal{T}_0 \subset \mathcal{T}$ and consider this as a non-defeasible part from which any argument rule can draw information that it might need. We call $\mathcal{T}_0$ the background knowledge base.

The notion of attack between arguments in a theory is based on the possible conflicts between a literal L and its negation and on the priority relation of h_p in the theory.

Definition 3.2
Let $(\mathcal{T}, \mathcal{P})$ be a theory, $T, T' \subseteq T$ and $P, P' \subseteq P$. Then (T', P') attacks (T, P) iff there exists a literal L, $T_1 \subseteq T'$, $T_2 \subseteq T$, $P_1 \subseteq P'$ and $P_2 \subseteq P$ such that

(i) $T_1 \cup P_1 \vdash_{\min} L$ and $T_2 \cup P_2 \vdash_{\min} \neg L$
(ii) $(\exists\, r' \in T_1 \cup P_1, r \in T_2 \cup P_2$ such that $T \cup P \vdash h\_p(r, r')) \Rightarrow (\exists\, r' \in T_1 \cup P_1, r \in T_2 \cup P_2$ such that $T' \cup P' \vdash h\_p(r', r))$.

Here $S \vdash_{\min} L$ means that $S \vdash L$ and that no proper subset of S implies L. When L does not refer to h_p, $T \cup P \vdash_{\min} L$ means that $T \vdash_{\min} L$. This definition states that a 'composite' argument (T', P') is a counter-argument to another such argument when it derives a contrary conclusion, L, and $(T' \cup P')$ makes the rules of its counterproof at least 'as strong' as the rules for the proof by the argument that is under attack. Note that the attack can occur on a contrary conclusion $L = h\_p(r, r')$ that refers to the priority between rules.

Definition 3.3
Let $(\mathcal{T}, \mathcal{P})$ be a theory, $T \subseteq \mathcal{T}$ and $P \subseteq \mathcal{P}$. Then (T, P) is admissible iff $(T \cup P)$ is consistent and for any (T', P'), if (T', P') attacks (T, P), then (T, P) attacks (T', P'). Given a ground literal L, then L is a credulous (respectively sceptical) consequence of the theory iff L holds in a (respectively every) maximal (with respect to set inclusion) admissible subset of T.

Hence when we have dynamic priorities, for an object-level argument (from $\mathcal{T}$) to be admissible, it needs to take along with it priority arguments (from $\mathcal{P}$) to make itself at least 'as strong' as the opposing counter-arguments. This need for priority rules can repeat itself when the initially chosen ones can themselves be attacked by opposing priority rules, and again, we would need to make now the priority rules themselves at least as strong as their opposing ones.

An agent's argumentation theory will be defined as a theory $(\mathcal{T}, \mathcal{P})$, which is further layered in separating $\mathcal{P}$ into two parts as follows.

Definition 3.4
An agent's argumentative policy theory, T, is a theory $T = (\mathcal{T}, (\mathcal{P}_R, \mathcal{P}_C))$ where the rules in $\mathcal{T}$ do not refer to h_p, all the rules in $\mathcal{P}_R$ are priority rules with head $h\_p(r_1, r_2)$ such that $r_1, r_2 \in \mathcal{T}$ and all rules in $\mathcal{P}_C$ are priority rules with head $h\_p(R_1, R_2)$ such that $R_1, R_2 \in \mathcal{P}_R \cup \mathcal{P}_C$.

We therefore have three levels in an agent's theory. In the first level, we have the rules $\mathcal{T}$ that refer directly to the subject domain of the theory at hand. We call these the *object-level decision rules* of the agent. In the other two levels, we have rules that relate to the policy, under which the agent uses its object-level decision rules, associated to normal situations (related to a default context) and specific situations (related to specific or exceptional contexts). We call the rules in $\mathcal{P}_R$ (named R in the following) and $\mathcal{P}_C$ (named C), *default* or *normal context priorities* and *specific context priorities*, respectively.

3.2. Modelling of the decision aiding process

In a nutshell, an automated decision aiding system implements two mappings that correspond to the steps of the DAP. The first mapping, which corresponds to the problem formulation, is one of the form Γ : Problem $\rightarrow < \mathbb{A}, \mathbb{V}, \Pi >$. The second mapping corresponds to the evaluation model construction and is of the form $\mathcal{M} : < A, V, \Pi > \rightarrow < A, D, H, U, R >$.

The first mapping can be implemented by a set of logical rules that associate various parameters, such as the features of the input problem, the situation at hand, the profile of the user etc., with the parameters of the problem formulation. For instance the rule

$$\text{select}(A, A_i) \leftarrow \text{feature}(P, F_1), \ldots, \text{feature}(P, F_n)$$

states that if the problem at hand P has the features F_1, F_n, then the parameter $\mathbb{A}$ of the problem formulation of P is instantiated by the set $\mathbb{A}_i$. In the following, we use the notation $C_{\mathbb{A},\mathbb{A}_i}(P)$ as a shorthand for the set of conditions that need to be satisfied by problem P in order for the parameter $\mathbb{A}$ to be instantiated by the set $\mathbb{A}_i$ in the problem formulation of P. Therefore, the aforementioned rule can be represented more compactly by select$(\mathbb{A}, \mathbb{A}_i) \leftarrow C_{\mathbb{A},\mathbb{A}_i}(P)$.

Having incomplete information about the world, such a model needs to account for the *lack of information*. To cope with this, in the automated DAP *assumptions* are made about other conditions that may influence the selection of the parameters of the problem formulation. These assumptions, and the mode of reasoning associated with them, can be captured in the argumentation framework that is used by rules of the following form:

$$\begin{aligned}
&r_i^{\mathbb{A}} : \text{select}(\mathbb{A}, \mathbb{A}_i) \leftarrow C_{\mathbb{A},\mathbb{A}_i}(P)\\
&r_j^{\mathbb{A}} : \text{select}\big(\mathbb{A}, \mathbb{A}_j\big) \leftarrow C_{\mathbb{A},\mathbb{A}_j}(P)\\
&R_{i,j}^{\mathbb{A}} : h_p\left(r_i^{\mathbb{A}}, r_j^{\mathbb{A}}\right)\\
&R_{j,i}^{\mathbb{A}} : h_p\left(r_j^{\mathbb{A}}, r_i^{\mathbb{A}}\right) \leftarrow SC_{\mathbb{A},\{j,i\}}(P)\\
&C_{j,i}^{\mathbb{A}} : h_p\left(R_{j,i}^{\mathbb{A}}, R_{i,j}^{\mathbb{A}}\right)
\end{aligned}$$

The aforementioned set of rules says that under the conditions $C_{\mathbb{A},\mathbb{A}_i}$, $\mathbb{A}_i$ is the *default* parameter selection for $\mathbb{A}$ in the problem formulation. If in addition to $C_{\mathbb{A},\mathbb{A}_i}$ some special conditions $SC_{\mathbb{A},\{j,i\}}$ hold for the problem at hand, then $\mathbb{A}_j$ is selected instead. Similar rules can be written for the other parameters of the problem formulation, i.e. for $\mathbb{V}$ and Π. Each set of rules that corresponds to each of the parameters $\mathbb{A}$, $\mathbb{V}$ and Π of Γ is denoted by $T_{\mathbb{A}}$, $T_{\mathbb{V}}$ and T_{Π}, respectively.

The next step in automating the DAP is to provide rules creating the mapping between the selected problem formulation and the possible evaluation models. This can also be done along the lines described above. Consider for instance the description of the relation between $\mathbb{A}$ in the problem formulation and the parameter A of an evaluation model. The rules that describe this mapping are of the form:

$$\begin{aligned}
&r_j^{A} : \text{select}\big(A, A_j\big) \leftarrow \text{select}(\mathbb{A}, \mathbb{A}_i), C_{A,A_j}(P)\\
&r_k^{A} : \text{select}(A, A_k) \leftarrow \text{select}(\mathbb{A}, \mathbb{A}_i), C_{A,A_k}(P)\\
&R_{j,k}^{A} : h_p\left(r_j^{A}, r_k^{A}\right)\\
&R_{k,j}^{A} : h_p\left(r_k^{A}, r_j^{A}\right) \leftarrow SC_{A,\{k,j\}}(P)\\
&C_{k,j}^{A} : h_p\left(R_{k,j}^{A}, R_{j,k}^{A}\right)
\end{aligned}$$

Similar rules are added for the other parameters of the evaluation model. The rules for the parameters D, H, U and R that correspond to the rule r_j^A are, respectively,

$$\begin{aligned}
&r_j^{D} : \text{select}\big(D, D_j\big) \leftarrow \text{select}(\mathbb{V}, \mathbb{V}_i), C_{D,D_j}(P)\\
&r_j^{H} : \text{select}\big(H, H_j\big) \leftarrow \text{select}(D, D_i), C_{H,H_j}(P)\\
&r_j^{U} : \text{select}\big(U, U_j\big) \leftarrow \text{select}(H, H_i), C_{U,U_j}(P)\\
&r_j^{R} : \text{select}\big(R, R_j\big) \leftarrow \text{select}(\Pi, \Pi_i), C_{R,R_j}(P)
\end{aligned}$$

Additional rules (of the type R and C), similar to those that have been described for A, are added to the argumentation theory and enforce different selections for the evaluation model parameters, wherever special conditions hold. The set of rules that are associated with the choice of the parameters of M are denoted by T_A, T_D, T_H, T_U and T_R, respectively.

Therefore, the resulting argumentation theory T is the union of the above subtheories, i.e. $T = T_{\mathbb{A}} \cup T_{\mathbb{V}} \cup T_{\Pi} \cup T_A \cup T_D \cup T_H \cup T_U \cup T_R$. At each cycle of the DAP that terminates with a rejection of the recommendations, the reasons for this rejection J are added to T, a new theory $T' = T \cup J$ is constructed, and a new reasoning phase starts, this time with the theory T'. In the following, we present an example that illustrates the method.

3.3. An illustrative example

An agent wishes to plan a dinner for this evening. He has four options. He could dine with his girlfriend, with his best friend, alone or stay at home and order a delivery. The agent prefers dining in a restaurant than staying home and dining with company than dining alone. In the first two cases, the venue is not as important as the company. When dining alone, the standard of the venue is very important. It must be in fact excellent in order to compensate for the lack of company. For dining very late at night, the agent prefers to dine alone, either out or order his favourite pizza for delivery. However, his decision criterion now becomes the time required for service.

The set of actions A relevant to the evening dinner can be represented as follows:

a_1 `dine_with_girlfriend`
a_2 `dine_with_best_friend`
a_3 `dine_alone` (X), $X \in \{r_1, \cdots r_n\}$
a_4 `order_pizza`

The `dine_alone` (X) action stands for a set of actions obtained by instantiating variable X with a specific restaurant.

The dining problem can be captured within the decision aiding model described earlier as follows. The agent can choose between two alternative problem formulations. The first is the formulation $\Gamma_1 = \langle \mathbb{A}_1, \mathbb{V}_1, \Pi_1 \rangle$, where $\mathbb{A}_1$ is the set of actions a_1, a_2, a_3, $\mathbb{V}_1$ is pleasure and venue standard and Π_1 is a choice (of the best thing to do this evening) or a

DOI: 10.1002/mcda

classification problem statement. A second problem formulation is $\Gamma_2 = \langle \mathbb{A}_2, \mathbb{V}_2, \Pi_2 \rangle$, where $\mathbb{A}_2$ is the same as $\mathbb{A}_1$, $\mathbb{V}_2$ is the time required for service and Π_2 is always a choice statement. Note that Γ_2 is applicable only in the case where dining takes place late at night. The construction of the two alternative problem formulations can be described in argumentation as follows, where vs denotes the venue standard. For simplicity, we replace the conditions in the right hand of the rules associated with a specific problem with the problem name.

$$
\begin{aligned}
&r_1^{\mathbb{A}} : \text{select}(\mathbb{A}, \{a_1, a_2, a_3\}) \leftarrow \text{dinner}\\
&r_2^{\mathbb{A}} : \text{select}(\mathbb{A}, \{a_3, a_4\}) \leftarrow \text{dinner}\\
&R_{1,2}^{\mathbb{A}} : h\_p\left(r_1^{a}, r_2^{a}\right)\\
&R_{2,1}^{\mathbb{A}} : h\_p\left(r_2^{\mathbb{A}}, r_1^{\mathbb{A}}\right) \leftarrow \text{late\_dinner}\\
&C_{2,1}^{\mathbb{A}} : h\_p\left(R_{2,1}^{\mathbb{A}}, R_{1,2}^{\mathbb{A}}\right)
\end{aligned}
$$

$$
\begin{aligned}
&r_1^{\mathbb{V}} : \text{select}(\mathbb{V}, \{\text{pleasure}, vs\}) \leftarrow \text{dinner}\\
&r_2^{\mathbb{V}} : \text{select}(\mathbb{V}, \{\text{service\_time}\}) \leftarrow \text{dinner}\\
&R_{1,2}^{\mathbb{V}} : h\_p\left(r_1^{\mathbb{V}}, r_2^{\mathbb{V}}\right)\\
&R_{2,1}^{\mathbb{V}} : h\_p\left(r_2^{\mathbb{V}}, r_1^{\mathbb{V}}\right) \leftarrow \text{late\_dinner}\\
&C_{2,1}^{V} : h\_p\left(R_{2,1}^{\mathbb{V}}, R_{1,2}^{\mathbb{V}}\right)
\end{aligned}
$$

$$
\begin{aligned}
&r_1^{\Pi} : \text{select}(\Pi, \{\text{choice}, \text{classif}\}) \leftarrow \text{dinner}\\
&r_2^{\Pi} : \text{select}(\Pi, \{\text{choice}\}) \leftarrow \text{dinner}\\
&R_{1,2}^{\Pi} : h\_p\left(r_1^{\Pi}, r_2^{\Pi}\right)\\
&R_{2,1}^{\Pi} : h\_p\left(r_2^{\Pi}, r_1^{\Pi}\right) \leftarrow \text{late\_dinner}\\
&C_{2,1}^{\Pi} : h\_p\left(R_{2,1}^{\Pi}, R_{1,2}^{\Pi}\right)
\end{aligned}
$$

After describing how the problem formulation alternatives are generated, the mapping between these formulations and the possible evaluation models needs to be specified. Take for instance the first formulation, Γ_1. The two alternative evaluation models that can be generated from it are $\mathcal{M}_1$ and $\mathcal{M}_2$ as specified below:

- $$
\begin{aligned}
\mathcal{M}_1 : \langle A &= \{a_1, a_2\},\\
D &= \text{pleasure},\\
H &= \{\text{pleasure} : a_1 >_p a_2\},\\
U &= \varnothing,\\
\mathcal{R} &= (\text{choice procedure})\rangle
\end{aligned}
$$
- $$
\begin{aligned}
\mathcal{M}_2 : \langle A &= \{a_3(X) : X \in \{r_1, \cdots r_n\}\},\\
D &= \{\text{venue standard, the associate scale}\\
&\quad\ \text{being Excellent (E), Acceptable (A)}\}\\
H &= \{\text{venue standard} : E > A\},\\
U &= \varnothing,\\
\mathcal{R} &= (\text{classification procedure})\rangle
\end{aligned}
$$

The construction of these evaluation models is also modelled in the argumentation theory. The rules that correspond to the construction of $\mathcal{M}_1$ and $\mathcal{M}_2$ are described below, where nagf and nabf stand for 'not available girlfriend' and 'not available best friend', respectively.

$$
\begin{aligned}
&r_1^{A} : \text{select}(A, \{a_1, a_2\}) \leftarrow \text{select}(A, A_1), \text{dinner}\\
&r_2^{A} : \text{select}(A, \{a_3\}) \leftarrow \text{select}(A, A_1), \text{dinner}\\
&R_{1,2}^{A} : h\_p\left(r_1^{A}, r_2^{A}\right)\\
&R_{2,1}^{A} : h\_p\left(r_2^{A}, r_1^{A}\right) \leftarrow \text{nagf}, \text{nabf}\\
&C_{2,1}^{A} : h\_p\left(R_{2,1}^{A}, R_{1,2}^{A}\right)
\end{aligned}
$$

$$
\begin{aligned}
&r_1^{D} : \text{select}(D, \{\text{pleasure}\}) \leftarrow \text{select}(\mathcal{V}, \mathcal{V}_1), \text{dinner}\\
&r_2^{D} : \text{select}(D, \{vs\}) \leftarrow \text{select}(\mathcal{V}, \mathcal{V}_1), \text{dinner}\\
&R_{1,2}^{D} : h\_p\left(r_1^{D}, r_2^{D}\right)\\
&R_{2,1}^{D} : h\_p\left(r_2^{D}, r_1^{D}\right) \leftarrow \text{nagf}, \text{nabf}\\
&C_{2,1}^{D} : h\_p\left(R_{2,1}^{D}, R_{1,2}^{D}\right)
\end{aligned}
$$

$$
\begin{aligned}
&r_1^{H} : \text{select}\left(H, \{a_1 >_p a_2\}\right) \leftarrow \text{select}(D, \{\text{pleasure}\}),\\
&\quad \text{dinner}\\
&r_2^{H} : \text{select}(H, \{E > A\}) \leftarrow \text{select}(D, \{vs\}), \text{dinner}\\
&R_{1,2}^{H} : h\_p\left(r_1^{H}, r_2^{H}\right)\\
&R_{2,1}^{H} : h\_p\left(r_2^{H}, r_1^{H}\right) \leftarrow \text{nagf}, \text{nabf}\\
&C_{2,1}^{H} : h\_p\left(R_{2,1}^{H}, R_{1,2}^{H}\right)
\end{aligned}
$$

$$
\begin{aligned}
&r_1^{R} : \text{select}(R, \{\text{choice}\}) \leftarrow \text{select}(\Pi, \{\text{choice}\}), \text{dinner}\\
&r_2^{R} : \text{select}(R, \{\text{classif}\}) \leftarrow \text{select}(\Pi, \{\text{classif}\}), \text{dinner}\\
&R_{1,2}^{R} : h\_p\left(r_1^{R}, r_2^{R}\right)\\
&R_{2,1}^{R} : h\_p\left(r_2^{R}, r_1^{R}\right) \leftarrow \text{nagf}, \text{nabf}\\
&C_{2,1}^{R} : h\_p\left(R_{2,1}^{R}, R_{1,2}^{R}\right)
\end{aligned}
$$

The following scenario illustrates how the agent could use the above argumentation theory T. When faced with a dining decision, theory T derives Γ_1 as the preferred formulation of the problem. Given Γ_1, the theory generates the evaluation model $\mathcal{M}_1$. The best choice according to $\mathcal{M}_1$ is dining with the girlfriend but the agent discovers that she is not available. The second choice is to dine with the best friend but he is also not available. This means that the condition nabf and nagf are true, and therefore, the dining decision has to be taken according to the model $\mathcal{M}_2$. By applying $\mathcal{M}_2$, he attempts to find an excellent

DOI: 10.1002/mcda

restaurant, but meanwhile, he discovers that it is already late evening. This means that late_dinner special condition is now true. The theory now becomes $T \cup \{\text{nabf, nagf, late\_dinner}\}$. The new theory derives that Γ_1 should be abandoned altogether and selects the problem formulation Γ_2, which leads to the evaluation model

$$\mathcal{M}_3 : \langle A = \{a_3, a_4\},$$
$$D = \{\text{min of estimates of service time for each action}\}$$
$$: \text{min}_\text{t}\text{ime}(a_3(X) :$$
$$X \in \{r_1, \cdots r_n\}) = 50 \ \text{min}, \text{min\_}time(a_4) = 10 \ \text{min}\}$$
$$H = \{\text{time} : x >_t y \ \ iff \ t(x) < t(y)\}$$
$$U = \varnothing, \mathcal{R} = (\text{choice procedure})\rangle$$

and the choice will be action a_4, which finally is implemented. The evaluation model $\mathcal{M}_3$ is generated by rules similar to those for $\mathcal{M}_1$ and $\mathcal{M}_2$.

In this section, we presented a first attempt to use argumentation within a Decision Aiding Process. The proposed approach is based on a set of logical rules that allow, among others, to make explicit the relation that exists between a problem formulation and its corresponding evaluation model. The aim is, on the one hand, to design autonomous agents able to undertake decision aiding tasks and, on the other hand, to show why such a theory could be useful for automatic decision purposes in autonomous agents. The idea is to allow agent to control the different artefacts of the DAP, in order to detect the parameters that should be updated and therefore to adapt the final decision. However, with this approach, some features remain not clear or unsatisfactory for a practical DAP. For instance, such a process is an interaction between two agents, and with this framework, it is not very clear how we can model this interaction, even through an automatic system. Moreover, it is desirable to generate justifications that explain the result to the final decision maker. In other words, there is a need to identify the elements that should be used in the construction of a convincing justification to the decision maker. Consider for instance the following example that represents a possible dialogue between a user and a system, which intents to assist the former to resolve his decision problem.

User: [specifies his evaluation model: two actions a and b, five criteria h_1, h_2, h_3, h_4, h_5, among which h_1 is considered to be of primary importance, followed by h_2, and so on. Acceptability is initially intrinsic: no action should have a score on a criterion lower than 1. The performance table is as follows:]

	h_1	h_2	h_3	h_4	h_5
a	7.5	6	2	3	5
b	7	4	8	4	7

1. **System:** given your information, a is at least as good as b.
2. **User:** why?
3. **System:** the most important criteria according to you defend this claim, so by comparing actions on the basis of criteria of decreasing importance, a should be preferred to b (although the majority of arguments defends the opposite claim).
4. **User:** I still think that h_1 is the most important criteria so the majority makes little sense. However, now I see that the difference between a and b is not significant enough...
5. **System:** My recommendation remains the same because a is preferred to b on the second most important criterion.
6. **User:** I agree but still a cannot be acceptable.
7. **System:** why?
8. **User:** b is way better than a on h_3, I could not justify this choice...
9. **System:** ok. But beware that applying this condition may have the following consequences...

This example dialogue illustrates how different types of explanations can be asked (and provided) and how the available information may change and be corrected (not only because the decision maker really changes his mind but also because the system necessarily make some assumptions that hold only by default). This is especially true when the decision maker is confronted with explicit justifications, as they enable him to identify relevant questions, and possible critics.

The full specification of the dialogue game is outside the scope of this paper (for more details, the reader can refer to Ouerdane (2009)). Our interest is directed towards constructing the different justifications. The aim of this paper is not to produce such natural language explanation but to provide the theoretical underpinnings upon which such explanations can later be generated. If we take the previous approach, then it is certainly useful to manage the revision and the update during the dialogue, but it does not give any information on how to provide such explanations (see turn 3, 5, ...). Thus, in what follows, we take a different approach that enables such a construction. We will base our model on what it is called argument schemes and their associated critical questions.

4. ARGUMENT SCHEMES FOR ARTEFACT MODELLING

In this section, we discuss how the steps involved in a multiple criteria evaluation process, which is a part of a Decision Aiding Process, can be captured in argumentative terms. To achieve this, we make use of the popular notion of argument schemes. A hierarchical structure of argument schemes allows to decompose the process into several distinct steps—and for each of them, the underlying premises are made explicit, which allows in turn to identify how these steps can be dialectically defeated.

One of the problems is that the field of multiple criteria analysis is extensive and diverse. It embraces a wide range of methodologies and approaches, commonly classified as value function approaches, outranking approaches, multi-objective mathematical programming methods etc. (see for instance Stewart (1992) for an overview of the field). Therefore, it is impossible to address all the existing methodologies and approaches in the context of this work. What we offer instead is a first contribution towards using formal argumentation in the construction of explanations in a DAP. More specifically, we attempt to identify the element that should be included in an argument so that it is meaningful in a decision aiding context. Our contribution is limited in several ways: (i) we focus on a multiple criteria evaluation model and the final recommendation (the two last artefacts of the DAP); (ii) we consider only the choice problem statement (the choice problem leads to a relative or pairwise evaluation among a set of actions); and (iii) we use only very simple numerical representation of the type $h_i(a)$ (called performance). In other words, the information available at the beginning of the process is represented mainly by a performance table. Furthermore, we use a very simple preference model of the type *aPb*, representing a preference between two actions, and *aIb*, representing the indifference between the two actions.

The hierarchical structure presented in the following is constructed on the basis of the above restrictions, and there are already different ways or means to model a problem situation (linear programming for an optimisation problem, for instance). Moreover, there exist several other types of problem statements (e.g. sorting, clustering). Thus, our approach should be adapted to meet different problem situations, and this is the subject of ongoing work.

4.1. Argument schemes

Argumentation is not only useful in representing the relations between the artefacts of the DAP but also for modelling the artefacts themselves. To this end, we will employ *argument schemes*, forms of arguments that capture stereotypical patterns of humans reasoning, especially defeasible ones (Norman *et al.*, 2003; Walton, 2005). Specifically, the arguments are presented as general inference rules whereby given a set of premises, a conclusion can be drawn (Walton, 1996). However, such schemes are not deductively strict because of the defeasible nature of arguments. The schemes allow for arguments to be represented within a particular context and take into account that the reasoning presented may be altered in the light of new evidence or exception to rules.

It is now well established that argument schemes can play two roles: (i) when constructing arguments, they provide a repertory of forms of argument to be considered, and a template prompting for the pieces that are needed; (ii) when attacking, arguments provide a set of critical question that can identify potential weaknesses in the opponents' case. Then, as Walton puts it, 'we have two devices, *schemes* and *critical questions*, which work together. The first device is used to identify the premises and conclusion. The second one is used to evaluate the argument by probing into its potentially weak points' (Walton and Reed, 2002). The set of critical questions have to be answered, when assessing whether their application in a specific case is warranted. Prakken and Bench-Capon (Bench-Capon and Prakken, 2005) specify that argument schemes are not classified according to their logical form but according to their content. Some argument schemes express epistemological principles or principles of practical reasoning: different domains may have different sets of such principles. Our aim in this paper is to identify those schemes that are involved in multicriteria evaluation process, one of the steps of a DAP.

Our goal is to devise a modelling framework that satisfies two desiderata. First, it should provide a formal justification or explanation of the final recommendation. Second, it must cope with the problem of the different versions of the evaluation model used to compute the final decision. The approach based on argument scheme that we sketch in what follows is particularly well suited to tackle these aspects: (i) by presenting the reasoning steps under the form of argument schemes, it makes justification possible, and offers the possibility to handle default reasoning with incomplete models, and (ii) by defining the set of attached critical questions, it establishes how the revision procedure can be handled.

DOI: 10.1002/mcda

4.2. A hierarchy of schemes for the evaluation process

In order to construct the whole evaluation model, we need different classes of argument schemes, which can be very broadly distinguished depending on (i) whether they are concerned with a single criterion or with the aggregation of several criteria, (unicriteria versus multicriteria); (ii) whether they follow a pairwise evaluation or whether they use an intrinsic evaluation, the action being compared with a neutral point (pairwise versus intrinsic); (iii) whether they are concerned with the evaluation of the action or its mere acceptability (evaluation versus acceptability); and (iv) whether they are concerned with a positive reason or a negative reason (positive versus negative).

The distinction between positive and negative reflects the fact that during the process, we can have two types of information: positive information and negative information. In the decision making context, such situations occur frequently (see Oztürk *et al*., 2005; Tsoukias *et al*., 2002): there always exist positive reasons supporting a certain decision and negative reasons against it. A characteristic example is the decision process of the Security Council of the United Nations, which is composed of 15 members (10 elected and 5 permanent). The rule for adopting a resolution requires that at least 9 of the 15 members agree and that no permanent member raises a veto. According to this decision rule, there exist agents who have a negative power, which is not compensated by the positive power of other agents when forming the majority. It acts independently and only negatively. These two powers cannot be combined, although they both influence the final decision. Therefore, the majority (positive power) is supported by a set of positive arguments pro the decision, and the veto (negative power) is advanced by a negative argument against that decision. Thus, it is possible to build arguments expressing either a 'positive' reason supporting an action or a 'negative' reasons against that action.

We turn now our attention to argument schemes. In fact, as it may have become clear from the aforementioned discussion, there is an underlying hierarchical structure that ties the different argument schemes. In short, we can distinguish three levels of argument schemes that will be embedded. At the highest level is the multicriteria pairwise evaluation (MC-PW-EV), which is based on the aggregation of positive and negative reasons, which in turn is based on unicriteria evaluation of actions versus other actions.

4.2.1. Argument schemes for unicriteria action evaluation. The first way to perform an action evaluation is to compare two actions from the point of view of the chosen criterion: this is modelled by the scheme for Unicriteria Pairwise Evaluation (UC-PW-EV) (see Table I). This argument scheme is the basic piece of reasoning that is required in our decision aiding context. It concludes that an action a is at least as good as an action b from the point of view of a given criterion h_i, based on some preference relation $\succeq_i$ (Oztürk *et al*., 2005).

When an action needs to be intrinsically evaluated, there is a need to define the categories and *separation profiles*. Such a separation profile defines on each criterion a sort of neutral point: this is not necessarily an existing action, but it allows to define to which category to affect the action. A particular case is when we only consider 'pro' and 'con' categories. The scheme for Unicriteria Intrinsic Action Evaluation, as given in Table II, details this scheme.

Now, at the same level (elementary level) but from the negative side, we propose argument schemes that reflect the concept of 'acceptability' of an action. By acceptable, we mean that it is not possible to find any evidence that express a strong negative reason against the action and therefore against the conclusion 'a is at least as good as b'. We distinguish two types of acceptability: intrinsic and relative.

Let us start with the intrinsic one. The idea is that an action is said to be acceptable if its evaluation does not exceed a certain threshold called a *veto* (noted μ). In the contrary case, we have negative reasons against such action and therefore the possibility to reject the proposition 'a is at least as good as b' (disregarding

Table I. Scheme for unicriteria pairwise evaluation

Premises	A criterion	h_i
	An action	a
	Whose performance is	$h_i(a)$
	An action	b
	Whose performance is	$h_i(b)$
	A preference relation	$\succeq_i$
Conclusion	a is at least as good as b	$a \succeq_i b$

Table II. Scheme for unicriteria intrinsic evaluation

Premises	A criterion	h_i
	An action	a
	Whose performance is	$h_i(a)$
	A separation profile	p_i
	Whose performance is	$h_i(p_i)$
	A preference relation	$\succeq_i$
Conclusion	a is assigned to a category	$a \succeq_i p_i$

DOI: 10.1002/mcda

the performance of the action a on the other criteria). The scheme of Table III illustrates such idea.

A different kind of acceptability relies instead on the relative comparison of actions: it may be the case that an action is considered to be unacceptable because the difference in performance is so huge with another action. In this case, we talk about an *Argument Scheme for Pairwise or Relative Acceptability* (UC-PW-AC). We believe this is self-explanatory, given the examples provided so far, and shall not give any further detail here.

4.2.2. Aggregation level. The problem of multicriteria aggregation is to synthesize information reflecting different aspects or point of views, sometimes conflicting, about the same set of actions. It is a significant issue in many evaluation procedures and comparison in decision aiding methodology. The aggregation level in our hierarchy is divided into two parts: positive side, called the *supporting reasons*, and negative side, called the *opposing reasons*. We note that the procedures that belong to the first category will be denoted by: SR-AG (name of the procedure) (Supportive Reasons-Aggregation).

4.2.2.1. Argument schemes for aggregating supporting reasons. The aim at the supporting aggregation level is to construct a set of supporting reasons that allow to support the claim 'a is at least as good as b', at the top of the hierarchy. Indeed, when we reach this level, we are confronted with a set of arguments for and against the claim. Obviously, these arguments reflect the position of each criterion regarding this claim. What conclusion to draw depends entirely on the procedure or the rule that is used to aggregate the arguments that are both in favour and against that conclusion. Different procedures necessarily yield different results.

Perhaps the most obvious such a scheme, at least one that is ubiquitous in multicriteria making, is the *principle of majority*. It only says that a is at least as good as b when there is a majority of criteria supporting this claim. Table IV gives the detail of the corresponding argument scheme.

Note that this scheme makes explicit that criteria are considered to be of equal importance. This is not necessarily the case, and more generally, many other aggregation techniques may be used to instantiate *SR* (see Table V). These other schemes will potentially require additional information, which justifies that we have many different schemes and not a single generic one. For instance, a possible scheme would conclude that a is at least as good as b when it is at least as good on (some of) the most important criteria (*argument from sufficient coalition of criteria*).

Another example is the one of the lexicographical method (Table VI). Unlike the majority principle that does not suppose an importance degree among criteria, in order to use the lexicographical method, it is necessary to establish a linear order among the criteria. This order expresses the fact that each criterion is totally or infinitely more important than all other criteria lower in this order, and no compensation is possible.

The last example that we present in this paper is the one of the weighted majority method. Table VII

Table III. Scheme for unicriteria intrinsic acceptability

Premises	A criterion	h_i
	An action	a
	Whose performance is	$h_i(a)$
	A veto threshold	μ_i
Conclusion	a is unacceptable according to h_i	$h_i(a) < \mu_i$

Table IV. Scheme for argument from the majority principle (SR-AG (maj))

Premises	A set of criteria considered to be of equal importance	$\{h_1, \ldots, h_n\}$
	A set of pairwise evaluation of actions a and b	
	The majority support the claim	
Conclusion	There are good reasons to support a is at least as good as b	$a \succeq b$

Table V. Scheme for multicriteria pairwise evaluation

Premises	An action	a
	An action	b
	A set of criteria	$\{h_1, \ldots, h_n\}$
	There are enough supporting reasons	*SR*
	There are no sufficiently strong opposing reasons	*OR*
Conclusion	a is at least as good as b	$a \succeq b$

Table VI. Scheme for argument from the lexicographical method (SR-AG (lex))

Premises	A set of criteria	$\{h_1, \ldots, h_n\}$
	A linear order on the set of criteria	$h_1 > \ldots > h_n$
	A set of pairwise evaluation of actions a and b	
	a is strictly better than b on h_i	$a \succ_i b$
	a is indifferent to b on h_j for any $j < i$	$a \simeq_j b (j < i)$
Conclusion	There are good reasons to support a is at least as good as b	$a \succeq b$

DOI: 10.1002/mcda

Table VII. Scheme for argument for the weighted majority method (SR-AG (W-maj))

Premises	A set of criteria	$\{h_1, \ldots, h_n\}$
	A set of importance coefficients	$\{w_1, w_2, \ldots, w_n\}$
	A set of pairwise evaluation of actions a and b	
	$W_{ab} = \sum_{i:aSb} w_i \geq W = \sum_{i:bSa} w_i$	$a \succeq b$
Conclusion	There are good reasons to support a is at least as good as b	$a \succeq b$

makes explicit the different assumptions underlying the use of this procedure.

Note that the basic input information that needs to be provided to these schemes is that of a pairwise comparison on a single criterion dimension (the output of UC-PW-EV). Indeed, this will be in most case the basic building block upon which the recommendation can be built. There is however a different type of scheme that would aggregate instead intrinsic valuations of both actions: that would be the case of argument-based aggregation procedures that take as input sets of arguments 'pro' and 'con'. Clearly, the basic argument scheme required will be different here, for it needs to provide an intrinsic evaluation of the action.

4.2.2.2. Argument schemes for aggregating opposing reasons. The last step before concluding that 'a is at least as good as b' is to check that there are no arguments against this conclusion. Indeed, we have explained at the beginning of this section that the establishment of the binary relation between two alternatives is based on the presence of positive information that supports this relation and the absence of negative information against it. By negative information, we refer to any information that will contradict the conclusion established on the positive side. Such information cannot be compensated by the positive reasons and acts independently.

In general, as in the positive side, we can imagine the use of any aggregation procedure to construct the negative reasons, either a multicriteria procedure or one of those used in an argumentation framework. However, in our work, we have restricted the opposing reasons for a particular type, called veto (intrinsic or absolute). Thus, the opposing aggregation aims at synthesizing all veto built at the elementary level.

An example for the existence of a negative aggregation side is the one presented through the second part of ELECTRE I (ELimination Et Choix Traduisant la REalité (ELimination and Choice Expressing REality)) method. The idea in ELECTRE I is to build a binary relation on the set A, called *outranking relation*. An alternative a outranks b if and only if the coalition of criteria such that a is better than b is sufficiently large and if b is not much better than a on a given dimension (Maystre *et al.*, 1994; Roy, 1968, 1971).

The first condition in ELECTRE I can be mainly represented by a weighted majority, where an *importance coefficient* w_i is associated with each criterion, and the large coalitions are those for which the sum of the importance coefficients is larger than a threshold c, called *concordance threshold*. The second condition expresses that a is much better than b on a given dimension because the difference between their performances exceeds a threshold that the decision maker considers as very large. This is what we call a veto. Table VIII gives the details of the scheme representing such a rule.

4.2.3. Multicriteria level. The argument scheme that lies at the top of our hierarchy is inspired by outranking multicriteria techniques (Bouyssou *et al.*, 2006), and its argumentative flavour is obvious. The claim holds when enough supportive reasons can be provided and when no exceptionally strong negative reason is known (see Table V).

This distinction between positive and negative already suggests that there will be (at least) two ways to attack this argument: either on the basis of a lack of positive support or on the basis of the presence of strong negative reasons (for instance, a 'veto'). Typically, supportive reasons are provided by action evaluation, and negative reasons are provided by action (lack of) acceptability.

We note that that in some cases, for instance, if the aggregation procedure corresponds to ELECTRE I, then we need a further premise in the multicriteria pairwise evaluation scheme (MC-PW-EV) to conclude that effectively 'a is at least as good as b' or more precisely that a outranks b. In fact, we recall that for the

Table VIII. Scheme for opposing reasons

Premises	A set of criteria	$\{h_1, \ldots, h_n\}$
	A relative acceptability on h_i	$g_i(a) < g_i(b) + \delta_i$
Conclusion	There are strong negative reasons against the claim	$a \not\succeq b$

DOI: 10.1002/mcda

outranking methods, the test to accept such an assertion is implemented by the existence of enough supporting reasons (concordance principle) and the absence of strong negative reasons (a non-discordance principle), which are already included in the MC-PW-EV scheme. The first principle, as we said before, can be represented by a weighted majority (see Table VII) and the second principle is designed through the existence of a veto (see Table VIII). Now, a further step in the model building phase is to combine these two measures to produce a measure of the degree of outranking, i.e. *a credibility value* that assesses the strength of the assertion that '*a* is at least as good as *b*'. The idea is that if the strength of the concordance exceeds that of the discordance, then the concordance value should not be modified. In other terms, if the discordance is verified for any $(a, b) \in A$ and any criterion j (which means that there is a veto), then we have no confidence that '*a* outranks *b*'. So, before concluding that '*a* is at least as good as *b*', we need to take into account such an information in the scheme. To account for that, we need a further premise that includes both the computation of such degree and the decision rule when the relation is accepted according to such a degree (in general, only measures that are sufficiently close to such degree are considered).

Finally, the construction of the hierarchy is the result of a dialogue between the system and its user. The system builds up the global relation '*a* is at least as good as *b*' by embedding argument schemes of the three levels. The argument schemes are built on the basis of the information provided by the user and, in some cases, by using default instantiation (when the scheme allows for it). Thus, the levels are nested as boxes so that the largest (the highest) contains the smallest (the lowest) one. Now, if the global relation is challenged by the user, then the system provides the different steps of reasoning by revealing parsimoniously the lower level schemes that compose the conclusion at the highest level. However, note that the argument schemes presented previously will allow building justification for a global preference relation between only two alternatives, which is not the final recommendation of a decision problem. In fact, to construct such a recommendation in the case, for instance, of a choice problem, we need to compare each action of the set A with all other actions and make a choice among the set of actions on the basis of each comparison. The result is that we obtain for each pair of comparisons its own hierarchy of schemes, containing the necessary arguments to justify the result of such a comparison. The final recommendation will be the aggregation of all the obtained results.

However, computing such a recommendation can be done directly or indirectly because the result delivered to the decision maker is not always a straightforward consequence of such different outputs. Therefore, we will consider an additional level in the hierarchy that is dedicated to the computation of the final recommendation of a decision problem. At this level, we suggest building arguments that justify the final recommendation or decision. Such justifications are constructed by taking into account both the results of the lower levels (MC-PW-EV) and the *choice procedure*, the Condorcet rule,[1] for instance, used to provide the final output. Table IX illustrates this idea.

4.3. Critical questions

It is necessary to provide the decision maker means to communicate with the system and to express his doubts on the conclusions and arguments presented during the process. Thus, the decision maker is involved in developing the recommendation, by pointing out those elements that appear missing or wrong in the reasoning steps taken by the system. The intervention of the decision maker can be realized by *critical questions*.

We recall that critical questions represent attacks, challenges or criticisms that, if not answered adequately, falsify the argument fitting the scheme. Moreover, Verheij (2003) distinguishes different roles for critical questions. For instance, they can point either to exceptional situations in which a scheme should not be used or to other argument that might be used to attack the scheme.

Table IX. Scheme for recommendation

Premises	A set of actions	A
	A set of criteria	$H = \{h_1, \ldots, h_n\}$
	Conclusion of multicriteria pairwise evaluation for pairwise comparison of a and each action $x \in A$	$a \succ x$
Conclusion	a is recommended *according to* a choice procedure	

[1]Other procedures can be used such as the kernel, the covering relation, ... (see Henriet, 1985; Laslier, 1997; Roy, 1991).

DOI: 10.1002/mcda

For instance, a possible move in the previous example dialogue (see section 3.3) would be that the user attacks the acceptability of the action on a given criterion by using the argument scheme for unicriteria intrinsic or relative acceptability (see turn 8). The critical question used, in this case, is '*is the action acceptable*?'. Another example is that the user may suggest an option that is different from the one proposed by the system as the solution to his problem. Of course, the user should justify his proposition by providing an argument supporting that option. We find this type of attack mainly at the highest level of the hierarchy, more specifically, when the system presents its recommendation for the user. Indeed, it is possible that the decision maker rejects the solution of the system on the basis that he favours, for some reasons, one of the eliminated actions as better than the proposed choice.

We now list some of the questions that can be attached to the different premises of our argument schemes (a more extensive list of questions can be found in the study conducted by Ouerdane, 2009). For instance, for the *Argument Scheme for Multicriteria Pairwise Evaluation* the different type of questions is clear: (i) *actions*: is the action possible? (ii) *list of criteria*: is this criteria relevant? Should we introduce a new criteria? Are these two criteria in fact the same?; (iii) *supporting reasons*: are there enough supporting reasons to support the claim?; and (iv) *opposing reasons*: are there not enough reasons to block the claim?

We note that the burden of proof lies on the proponent when he must provide supportive evidence (positive reasons) for the main claim. On the contrary, the opponent should be the one providing negative reasons to block the conclusion. Moreover, note also that although the use of a specific aggregation technique may be challenged at this level ('why are we using a majority principle here?'), the actual exchange of argument regarding this aspect will involve the subargument scheme concerned with this aggregation.

Along with the *Scheme for Argument from the Majority Principle*, come two obvious questions: (i) *list of criteria*: are the criteria of equal importance? and (ii) *majority aggregation*: is the simple majority threshold relevant to the current decision problem?

As for the *Argument Scheme for Unicriteria Pairwise Action Evaluation*, we propose this tentative set of questions: (i) *actions*: is the action possible?; (ii) *criterion*: is the criterion relevant?; (iii) *action's performance*: is the performance correct?; and (iv) *preference relation*: is the preference relation appropriate?

A negative answer to some of these questions leads to a conflict whose resolution requires sometimes the transition to a different stage of the DAP. For instance, when the action feasibility is challenged, this essentially concerns the problem formulation (cf. Section 2), where the set of alternatives is defined. This can be handled by our framework, as it also models the dependencies between the artefacts. However, a detailed discussion of this issue is outside the scope of this paper. We just notice that through the different critical questions, we have the opportunity to review and correct not only the evaluation model but also other stages of the process. Indeed, as it was showed in the study conducted by Verheij (2003), the critical questions have different roles: (i) they can be used to question wether a premises of scheme holds; (ii) they can point to exceptional situations in which a scheme should not be used; (iii) they can set condition for the proper use of a scheme; and (iv) they can point to other argument that might be used to attack the scheme.

Moreover, the evaluation of an argument involves shifts of burden of proof in a *dialogue*. When the respondent asks one of the critical questions, then the burden of proof shifts back to the proponent's side, attacking the argument temporarily until the critical question has been answered successfully (Gordon *et al*., 2007; Reed and Walton, 2007).

4.4. An illustrative example

In this section, we go back to the dialogue example presented in Section 3.3 and show how it exploits the argument schemes and critical questions that we have put forward so far. We reiterate that the full specification of the dialogue game is not the subject of this paper. The process initiates with the client specifying the basic elements of the evaluation model[2]: it specifies a set of actions (for sake of clarity of the paper we limit ourselves to two actions though), a set of criteria and the aggregation operators that shall be used. Contrary to classical decision tools, these sets should be seen only as the *current* evaluation model, which can be revised throughout the process. Now, as we see it, an argumentation-based decision aiding process should.

(i) *present a recommendation that can be explicitly justified*. Crucially, by presenting its justifications in the form of arguments, the system will make it

[2]Of course a more ambitious dialectical system would have to consider the previous steps as well. This is beyond the scope of this paper.

DOI: 10.1002/mcda

possible for the user to pinpoint those steps that pose problems. The system builds up the current recommendation by embedding argument schemes of the three levels. The argument schemes are built on the basis of the information provided by the user and, in some cases, by using default instantiation (when the scheme allows for it). If challenged by the user, then the system provides the different steps of reasoning by revealing parsimoniously the lower level schemes that compose the recommendation. Each time a scheme is presented, the entire set of critical questions is at the disposal of the user to challenge the current conclusion. There are very different reasons to revise in such a process: in certain case, the user may simply want to correct/refine one of its previous statements or introduce new information. In other cases, it will contradict one of the system's assumptions.

(ii) *to revise any piece of reasoning involved in this process and be informed of the consequences of such moves*. In many cases, the user would not foresee the various consequences of a seemingly local modification: in these cases, the system helps the user by making explicit the hidden and critical consequences of its move.

Let us briefly analyse such a dialogue. Turn 1 provides the recommendation, which is challenged by the user on turn 2. Not being more explicit, the challenge can be assumed to ask the system to provide more explicit information regarding positive reasons supporting the claim. The system, in turn 3, explains that the claim is based on the use of the SR-AG(lex) scheme (see Table VI). Note that it also generates a possible counter-argument by relaxing some of the information provided by the user (here the fact that criteria have different importance). Observe that this is an indirect way for the system to use a critical question. The user rejects this counter-argument in turn 4 (by re-affirming the fact that criteria have unequal importance) but attacks the basic UC-PW-EV argument upon which the recommendation is based. The critical question used here is that of the relevance of the preference relation. The system accepts the move (and modifies the user's information by specifying that actions should exhibit at least half a point of difference; otherwise, they should be considered as indifferent). But the system restates that the recommendation remains unchanged: this is because of the fact on the second most important criterion, *a* is again better than *b*. The user accepts this but now attacks on the ground of negative reasons and explains that *a* cannot be accepted on the basis of pairwise acceptability (UC-PW-AC). Finally, the system revises its recommendation but may at the same time make explicit the consequences of the proposed change.

5. CONCLUSIONS

In this paper, we studied the problem of integrating argumentation in decision aiding in order to either help the expert or to replace him in some decision tasks. We used a highly expressive argumentation framework, which allowed us to capture the defeasible character of the cognitive artefacts generation process. Indeed, we showed that it can model the interdependencies between the artefacts constructed during decision aiding and can therefore adapt to the dynamics of the process.

Moreover, we provided a first account of the modelling of the steps of a multicriteria DAP by means of argument schemes and critical questions. We focused on the evaluation model and considered mainly the restricted but fundamental case of the comparison of two actions. To represent the decision evaluation process, we identified a hierarchical structure of argument schemes. Each level refers to one step in the classical multicriteria evaluation. The highest level represents the pairwise evaluation, which is based on the aggregation level, which is in turn based on unicriteria evaluation (pairwise or intrinsic). To these schemes, we associated a set of critical questions.

Our proposal is an attempt to provide a formal unifying account to decision aiding for descriptive but also prescriptive purposes. The specific automated decision aiding model we proposed gives the flexibility to choose dynamically evaluation models and thus to take the most appropriate type of decisions (best choice, ranking or classification of the possible alternatives, etc.) according to the circumstances at hand.

The subject of this paper naturally raises the question of whether decision aiding is a process that is amenable to automation. We reiterate that the work described here is far from a fully automated general method, given that the construction of the artefacts is based on predefined 'components'. Therefore, it should be seen as a step towards achieving more flexibility in dynamic environments. As noted in the introduction, this is an important issue, as there seems to be an increasing need towards automating some forms of decision aiding, which for the moment are not very complex (e.g. aiding a user in shopping on the Internet).

DOI: 10.1002/mcda

Certainly such a need is not specific to decision aiding only. For instance, in fields such as Service Computing, Software Engineering and Systems Management, the quest for automation takes the form of Service Composition (Rao and Su, 2004), Self-Adaptive Software (Oreizy *et al*., 1999) and Autonomic Computing (Ganek and Corbi, 2003), respectively. The research efforts there are not restrained by the possibility of the impossibility of the ultimate goal of fully automating the procedures involved in these fields or even the difficulty of building highly adaptive systems. Even if full automation is a goal that will be never reached, intermediate results may lead to systems and tools that feature high flexibility and adaptivity to the benefit of their users.

There are several directions for future research. This work gives only the very basic ingredients of the dialectical system currently under development. One of our ongoing research efforts focuses on specifying the dialogue game involved in decision aiding. We also work on extending the model to take into account, on one hand, a large set of alternatives, and, on other hand, to handle different decision problems (ranking, sorting, . . .). Our ultimate goal is to build a dialectical system that covers the whole DAP.

Moreover, this work studied the problem of decision aiding of individual entities, or 'agents' in the AI parlance, where argumentation is used as a self-deliberation mechanism. However, the same argumentation framework can be used in multi-agent context in order to model agent interaction (Kakas *et al*., 2004). In this case, the revision of the produced artefacts of the DAP can be triggered by the defeat, during an inter-agent argumentation-based interaction, of the arguments involved in the construction of these artefacts. Finally, it seems that integrating these techniques with ideas from design rationale is another promising avenue of future research.

REFERENCES

Amgoud L. 2009. Argumentation for decision making. In *Argumentation in Artificial Intelligence*, Rahwan I, Simari G (eds). Springer: US; 301–320.

Amgoud L, Bonnefon J-F, Prade H. 2005. An argumentation-based approach to multiple criteria decision. *Proceedings of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*. LNCS; 269–280.

Amgoud L, Prade H. 2006. Explaining qualitative decision under uncertainty by argumentation. In *Proceedings of the 21st Conference on Artificial Intelligence (AAAI'06)*, Yolanda G, Moo R (eds). AAAI Press: Boston, Massachusetts; 219–224.

Atkinson K. 2006. Value-based argumentation for democratic support. In *Proceedings of the 1st International Conference on Computational Models of Natural Argument(COMMA'06)* vol. **144**, Dunne PE, Bench-Capon TJM (eds). IOS Press: Amsterdam, The Netherlands; 47–58.

Atkinson K, Bench-Capon TJM, Modgil S. 2006. Argumentation for decision support. In *Proceedings of the 17th International Conference on Database and Expert Systems Applications*, Bressan S, Küng J, Wagner R (eds). Springer: Kraków, Poland; 822–831.

Bana a Costa C. 1996. Les problématiques de l'aide à la décision: vers l'enrichissement de la trilogie choix-tri-rangement. *Recherche Operationnelle* **30**(2): 191–216.

Belton V, Stewart T. 2002. *Multiple Criteria Decision Analysis: An Integrated Approach*. Kluwer Academic: Dordrecht.

Bench-Capon TJM, Prakken H. 2005. Argumentation. In *Information Technology & Lawyers: Advanced Technology in the Legal Domain from Challenges to Daily Routine*, Lodder AR, Oskamp A (eds). Springer-Verlag: Berlin; 61–80.

Bonet B, Geffner H. 1996. Arguing for decisions: a qualitative model of decision making. *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann: Portland, Oregon, USA; 98–105.

Bouyssou D, Marchant T, Pirlot M, Perny P, Tsoukiàs A, Vincke PH. 2000. *Evaluation and Decision Models: A Critical Perspective*. Kluwer Academic: Dordrecht.

Bouyssou D, Marchant T, Pirlot M, Tsoukiàs A, Vincke P. 2006. *Evaluation and Decision Models with Multiple Criteria: Stepping Stones for the Analyst*. International Series in Operations Research and Management Science vol. **86**. Springer: Boston, MA.

Burge JE, Brown DC. 2008. SEURAT: integrated rationale management. *Proceedings of the 30th international conference on Software engineering*, ACM, ICSE '08, New York, NY, USA; 835–838.

Chen L, Pu P. 2007. Preference-based organization interfaces: Aiding user critiques in recommender systems. In *Proceedings of 11th International Conference on User Modeling (UM'07)*, Conati C, McCoy KF, Paliouras G (eds); 77–86.

Chesñevar CI, Maguitman AG. 2004. ArgueNet: an argument-based recommender system for solving web search queries. *Proceedings the 2nd. International IEEE Conference on Intelligent Systems*. IEEE Press: Varna, Bulgaria; 282–287.

Chesñevar CI, Maguitman AG, Simari GR. 2004. A first approach to argument-based recommender systems based on defeasible logic programing. In *Proceedings of the 10th International Workshop on Non Monotonic Reasoning (NMR'04)*, Delgrande JP, Schaub T (eds). Whistler, Canada; 109–117.

Chesñevar CI, Maguitman AG, Simari GR. 2006. Argument-based critics and recommenders: a qualitative

DOI: 10.1002/mcda

perspective on user support systems. *Journal of Data and Knowledge Engineering* **59**(2): 293–319.

Dastani M, Hulstijn J, van der Torre L. 2005. How to decide what to do? *European Journal of Operations Research* **160**(3): 762–784.

Doyle J, Thomason R. 1999. Background to qualitative decision theory. *AI Magazine* **20**(2): 55–68.

Fox J, Krause P, Elvang-Gøransson M. 1993. Argumentation as a general framework for uncertain reasoning. In *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence (UAI'93)*, Heckerman D, Mamdani A (eds); 428–434.

Fox J, Parsons S. 1997. On using arguments for reasoning about actions and values. *Proceedings of the AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning*, AAAI Press: Standford; 55–63.

French S. 1988. *Decision Theory—An Introduction to the Mathematics of Rationality*. Ellis Horwood: Chichester.

Ganek AG, Corbi TA. 2003. The dawning of the autonomic computing era. *IBM Systems Journal* **42**(1): 5–18.

Gordon T, Prakken H, Walton D. 2007. The Carneades model of argument and burden of proof. *Artificial Intelligence* **171**: 875–896.

Gorgias. 2002. A system for argumentation and abduction. Available at http://www.cs.ucy.ac.cy/~nkd/gorgias [accessed 10/20/2011].

Henriet D. 1985. The Copeland choice function: an axiomatic characterization. *Social Choice and Welfare* **2**: 49–64.

Kakas AC, Mancarella P, Dung PM. 1994. The acceptability semantics for logic programs. *Proceedings of the 11th International Conference on Logic Programming*. MIT Press: Cambridge, MA; 504–519.

Kakas AC, Maudet N, Moraitis P. 2004. Layered strategies and protocols for argumentation-based agent interaction. *Proceedings in the 1st International Workshop on Argumentation in Multi-Agent Systems*, *LNCS* vol. **3366**. Springer: New York; 64–79.

Kakas A, Moraitis P. 2003. Argumentation based decision making for autonomous agents. *Proceedings of the 2nd international joint conference on Autonomous agents and multiagent systems*, ACM: Melbourne; 883–890.

Klein DA. 1994. *Decision-Analytic Intelligent Systems: Automated Explanation and Knowledge Acquisition*. L. Erlbaum Associates Inc.: Hillsdale, NJ, USA.

Kunz W, Rittel H. 1970. Issues as elements of information systems. Technical Report 131, Center for Planning and Development Research, University of California at Berkeley.

Labreuche C. 2006. Argumentation of the decision made by several aggregation operators based on weights. *Proceedings of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'06)*, Paris, France; 683–690.

Laslier J-F. 1997. *Tournament Solutions and Majority Voting*. Springer-Verlag: Berlin.

MacLean A, Young RM, Moran TP. 1989. Design rationale: the argument behind the artifact. *SIGCHI Bulletin* **20**: 247–252.

Maystre LY, Pictet J, Simos J. 1994. *Méthodes Multicritères ELECTRE. Description, Conseils Pratiques et cas D'application à la Gestion Environnementale*. Presses Polytechniques et Universitaires Romandes: Lausanne.

Morge M. 2004. Computer supported collaborative argumentation. *Proceedings of 4th workshop on Computational Models of Natural Argument (CNMA04)*, Valencia, Spain; 1–4.

Norman TJ, Carbogim DV, Krabb EC, Walton D. 2003. Argument and multi-agent systems. In *Argumentation Machines: New Frontiers in Argument and Computation* (Chapter 2), Reed C, Norman TJ (eds). Kluwer Academic Publishers: Dordrecht, the netherlands; 15–54.

Oreizy P, Gorlick MM, Taylor RN, Heimbigner D, Johnson G, Medvidovic N, Quilici A, Rosenblum DS, Wolf AL. 1999. An architecture-based approach to self-adaptive software. *IEEE Intelligent Systems* **14**: 54–62.

Ouerdane W. 2009. Multiple criteria decision aiding: a dialectical perspective. PhD thesis, University Paris-Dauphine, Paris, France.

Ouerdane W, Maudet N, Tsoukiàs A. 2007. Arguing over actions that involve multiple criteria: a critical review. *Proceedings of the 9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, Hammamet, Tunisia; 308–319.

Ouerdane W, Maudet N, Tsoukiàs A. 2009. Argumentation theory and decision aiding. In *New Trends in MCDA*, Figueira J, Greco S, Ehrgott M (eds). International Series in Operations Research and Management Sciences vol. **142**. Springer-Verlag: Berlin; 177–208.

Oztürk M, Tsoukiàs A, Vincke Ph. 2005. Preference modelling. In *Multiple Criteria Decision Analysis: State of the Art Surveys*, Figueira J, Greco S, Ehrgott M (eds). Springer-Verlag, Boston, MA; Dordrecht; 27–72.

Parsons S, Greene S. 1999. Argumentation and qualitative decision making. In *Proceedings of the 3rd European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'95)*, Froidevaux C, Kohlas J (eds). Springer: Fribourg, Switzerland; 328–340.

Parsons S, Jennings N. 1998. Argumentation and multiagent decision making. *Proceedings of the AAAI Spring Symposium on Interactive and Mixed-Initiative Decision Making*, Stanford; 89–91.

Pu P, Chen L. 2007. Trust-inspiring explanation interfaces for recommender systems. *Knowledge-Based Systems* **20**(6): 542–556.

Rao J, Su X. 2004. A survey of automated web service composition methods. *Semantic Web Services and Web Process Composition (SWSWPC)*; 43–54.

Reed C, Walton DN. 2007. Argument schemes in dialogue. In *Dissensus and the Search for Common Ground*, Hansen HV, Tindale CW, Johnson RH, Blair JA (eds). Ontario Society for the Study of Argumentation (OSSA), University of Windsor: Windsor, Canada.

Rittel HWJ, Webber MM. 1973. Dilemmas in a general theory of planning. *Policy Sciences* **4**: 155–169.
Roy B. 1968. Classement et choix en présence de points de vue multiples: la méthode ELECTRE. *Revue Francaise d'Informatique et de Recherche Opérationnelle* **8**: 57–75.
Roy B. 1971. Problems and methods with multiple objective functions. *Mathematical Programming* **1**(2): 239–266.
Roy B. 1991. The outranking approach and the foundations of ELECTRE methods. *Theory and Decision* **31**: 49–73.
Shankar RD, Tu SW, Musen MA. 2006. Medical arguments in an automated health care system. *Proceedings of the AAAI Spring Symposium*, AAAI Press: Stanford University; 55–63.
Sillince J. 1994. Multi-agent conflict resolution: a computational framework for an intelligent argumentation program. *Knowledge-Based Systems* **7**: 75–90.
Stewart TJ.1992. A critical survey on the status of multiple criteria decision making theory and practice. *Omega* **20**(5–6): 569–586.
Tsoukiàs A. 2007. On the concept of decision aiding process. *Annals of Operations Research* **154**(1): 3–27.
Tsoukiàs A. 2008. From decision theory to decision aiding methodology. *European Journal of Operational Research* **187**(1): 138–161.
Tsoukias A, Perny P, Vincke P. 2002. From concordance/discordance to the modelling of positive and negative reasons in decision aiding. In Bouyssou D, Jacquet-Lagreze E, Perny P, Slowinski R, Vanderpooten D (eds). *Aiding Decisions with Multiple Criteria: Essays in Honor of Bernard Roy*. Kluwer Academic: Dordrecht; 147–174.
Verheij B. 2003. Dialectical argumentation with argumentation schemes: an approach to legal logic. *Artificial Intelligence and Law* **11**(2–3): 167–195.
Walton DN. 1996. *Argumentation Schemes for Presumptive Reasoning*. Erlbaum: Mahwah, NJ.
Walton DN. 2005. Justification of argument schemes. *Australasian Journal of Logic* **3**: 1–13.
Walton DN, Reed CA. 2002. Argumentation schemes and defeasible inferences. In *Workshop on Computational Models of Natural Argument*, Carenini G, Grasso F, Reed C (eds).