



HAL
open science

Multiple criteria decision aiding: a dialectical perspective

Wassila Ouerdane

► **To cite this version:**

Wassila Ouerdane. Multiple criteria decision aiding: a dialectical perspective. Artificial Intelligence [cs.AI]. Université Paris Dauphine - Paris IX, 2009. English. NNT: . tel-01792308

HAL Id: tel-01792308

<https://hal.science/tel-01792308v1>

Submitted on 15 May 2018

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

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

UNIVERSITÉ PARIS DAUPHINE

N° attribué par la bibliothèque

--	--	--	--	--	--	--	--	--	--



THÈSE

Pour l'obtention du titre de
DOCTEUR EN INFORMATIQUE
(Arrêté du 7 août 2006)

Spécialité: Aide à la Décision, Intelligence Artificielle

Multiple Criteria Decision Aiding: a Dialectical Perspective

Candidate: Wassila OUERDANE

JURY

- Directeur de thèse : Alexis Tsoukiàs
Directeur de recherche (CNRS), Université Paris-Dauphine
- Rapporteurs : Simon Parsons
Professeur, Department of Computer
and Information Science Brooklyn College New York
- Patrice Perny
Professeur, Université Pierre et Marie Curie
- Suffrageants Leila Amgoud
Chargé de Recherche (CNRS), Université Paul Sabatier Toulouse
- Sylvie Coste-Marquis
Maître de conférences, Université d'Artois
- Christophe Labreuche
Ingénieur Chercheur, Thales (*invité*)
- Thierry Marchant
Professeur, Ghent University Belgium
- Nicolas Maudet
Maître de conférences, Université Paris-Dauphine

Présentée et soutenue publiquement le 1 décembre 2009.

L'université n'entend donner aucune approbation ni improbation aux opinions émises dans les thèses: ces opinions doivent être considérées comme propres à leurs auteurs.

Contents

INTRODUCTION	5
I Literature Review	15
1 Multi-criteria Decision Aiding (MCDA)	16
1.1 Introduction	16
1.2 A model of Decision Aiding Process (DAP)	17
1.2.1 The problem situation	18
1.2.2 Problem formulation	19
1.2.3 Evaluation Model	20
1.2.4 Final recommendation	20
1.3 Basic concepts of MCDA	21
1.3.1 Set of actions	21
1.3.2 Criterion and family of criteria	21
1.3.3 The choice problem statement	23
1.4 On the notion of preferences	25
1.4.1 Preference modeling	26
1.4.1.1 Preference structures	28
1.4.2 Preference aggregation	30
1.4.2.1 Aggregation procedures	30
1.4.2.2 How to choose a procedure?	33
1.4.3 Exploiting preferences	35

1.5	What is missing in the DAP model?	37
1.6	Conclusion	41
2	Argumentation Theory	42
2.1	Introduction	42
2.2	Argumentation theory: an overview	44
2.2.1	Structures of arguments	47
2.2.1.1	Argument for beliefs vs for decisions	50
2.2.2	How arguments interact?	52
2.2.3	Arguments in dialogue	58
2.2.3.1	Dialogue system	58
2.2.3.2	Argument schemes in dialogue	59
2.2.4	Computing acceptability	60
2.2.5	Constructing “natural” arguments	64
2.2.6	Diagramming of arguments	69
2.3	Conclusion	72
3	Arguing for decision: a <i>critical review</i>	73
3.1	Introduction	73
3.2	Argument schemes for action: some existing approaches	74
3.2.1	Scheme for actions and values	75
3.2.2	Scheme for practical reasoning	77
3.2.3	Argument and multi-criteria decision making	78
3.3	Discussion	80
3.4	Conclusion	82
II	MCDA: a dialectical perspective	83
4	Different levels for different reasons	84
4.1	Introduction	84
4.2	Arguing over actions	85

4.2.1	Criterion vs. Argument	86
4.2.2	Structure of an argument	87
4.2.3	Aggregation procedures vs. Proof standards	91
4.3	A hierarchy of schemes	92
4.3.1	Elementary (or Unicriteria) level	94
4.3.2	Aggregation Level	99
4.3.2.1	Argument schemes for Aggregating Supporting reasons	99
4.3.2.2	Set of “properties” to support the choice of a proof	102
4.3.2.3	Argument schemes for Aggregating opposing reasons	105
4.3.3	Multicriteria Level	106
4.4	Conclusion	108
5	Towards an argumentation-based decision aiding tool	110
5.1	Introduction	110
5.2	Choosing a proof standard	113
5.2.1	Formal Concept Analysis (FCA)	116
5.2.2	FCA for choosing a proof standard	118
5.3	Generating arguments	123
5.3.1	Computing minimal explanation for Condorcet-winner	124
5.3.2	Computing explanation for a non Condorcet winner	135
5.4	Generating counter-arguments	136
5.5	Acceptability of arguments	141
5.6	Managing the dialogue	147
5.6.1	Basic structures	148
5.6.2	Locution rules	148
5.6.3	Commitment rules	152
5.6.4	Dialogue rules	153
5.7	Properties of the dialogue game	155
5.8	Conclusion	156
6	ArgDec: a prototype for an Argumentation based Decision aiding tool	157

6.1	Introduction	157
6.2	ArgDec: an overview	158
6.3	Illustrative example	160
CONCLUSION		175
Bibliography		182
III Appendix		195
A	Binary relations	196
B	Toulmin’s scheme with the Carneades Software	197
C	Argumentative Dialogue	200
D	Minimal explanations w.r.t Generic statements	202
E	An example of dialogue	207

List of Figures

1.1	Relative evaluation	24
1.2	How to aggregate preferences?	30
2.1	Toulmin's Scheme	49
2.2	Toulmin's Scheme	49
2.3	Construction of Tweety's scheme in Carneades	71
2.4	Reconstruction of Toulmin Diagrams in Carneades	71
4.1	Example of an argument for pairwise comparison	90
4.2	General structure of an argument	90
4.3	Different classes of argument schemes	93
4.4	A hierarchy of argument schemes	95
4.5	Explicit representation of the proof standard	100
4.6	Explicit representation of the proof standard	103
4.7	Abstraction levels of argumentation	107
5.1	System's functionalities	113
5.2	An example of a concept lattice for objects consisting of integer numbers and attributes	118
5.3	A concept lattice for objects consisting of proof standard and attributes consisting of properties.	120
5.4	Concept lattice to guide the choice of a proof standard	121
5.5	Concept lattice to guide the choice of a proof standard (adapted)	122
5.6	Argument graph for a multiple criteria context	142

5.7	Example of an argument graph	144
5.8	State Transition Diagram	152
6.1	Introductory screen	159
6.2	Main elements of the arguments graph	159
6.3	Screen shot of the XML file	160
6.4	Problem statement screen	161
6.5	Information on criteria screen	161
6.6	Information on alternatives screen	161
6.7	Stating a recommendation	162
6.8	A first level of explanation	163
6.9	Minimal explanation	164
6.10	Graph for a minimal explanation	165
6.11	Example of update	166
6.12	Graph after the update	167
6.13	New minimal explanation	168
6.14	Graph with the new minimal explanation	168
6.15	Example of a counter-argument	169
6.16	Example of user's attack	170
6.17	User's attack (graph)	170
6.18	User's critical response	171
6.19	Challenging the proof	171
6.20	Presenting a new recommendation	173
6.21	Attack against the user's proposition	173
6.22	representation of a strong negative reason	174
B.1	Toulmin under Carneades(1)	198
B.2	Toulmin under Carneades(2)	198
B.3	Toulmin under Carneades (3)	199
D.1	The search tree	205

List of Tables

1.1	Examples of problem statements	23
2.1	Tweety's scheme	48
2.2	Toulmin's scheme	50
2.3	The necessary condition Scheme	51
2.4	The sufficient condition Scheme	52
2.5	A Rebut for Tweety's scheme	53
2.6	Exception in Toulmin's schemes	53
2.7	Exception in Tweety's scheme	54
2.8	Scheme for Argument from expert Opinion	56
2.9	Examples of speech acts	59
2.10	Types of dialogue [Norman et al., 2003]	60
3.1	Fox and Parsons' argument Scheme	75
3.2	Fox and Parsons' scheme: a multi-criteria vision	77
3.3	Atkinson's scheme	77
3.4	Atkinson's scheme: a multi-criteria vision	78
3.5	Amgoud and Prade's scheme: a multi-criteria vision	80
4.1	Scheme for UNICRITERIA PAIRWISE EVALUATION (UC-PW-EV)	95
4.2	Example of scheme for UC-PW-EV	95
4.3	Scheme for UNICRITERIA INTRINSIC EVALUATION (UC-IN-EV)	96
4.4	Scheme for UNICRITERIA PAIRWISE EVALUATION (version 2)	96

4.5	Scheme for UNICRITERIA INTRINSIC ACCEPTABILITY (UC-IN-ACC)	97
4.6	Example of scheme for UC-IN-ACC	97
4.7	Scheme for UNICRITERIA RELATIVE ACCEPTABILITY (UC-PW-ACC)	98
4.8	scheme for the UNICRITERIA TOP RANKING (UC-TOP)	99
4.9	Scheme for UNICRITERIA GOOD RANKING (UC-GOOD)	99
4.10	Scheme for SUPPORTING REASONS (SR)	100
4.11	Scheme for Argument from the Majority method (MAJO-SCHEME)	101
4.12	Scheme for Argument from the Lexicographic method (LEXICO-SCHEME)	101
4.13	Scheme for Argument for the Weighted Majority (WEIGHTEDM-SCHEME)	101
4.14	Scheme for Argument from the weighted Sum (WEIGHTEDS-SCHEME)	102
4.15	Set of properties for identifying proof standards	105
4.16	Scheme for opposing reasons	106
4.17	Scheme for MULTICRITERIA PAIRWISE EVALUATION (MC-PW-EV)	106
4.18	Scheme for recommendation (RECOM-SCHEME)	108
5.1	Example of a formal context	117
5.2	A formal context for objects consisting of proof standards and attributes consisting of properties describing them.	119
5.3	Properties of minimal cost explanations	134
5.4	Examples of Critical Questions	139
5.5	Examples of critical responses against the properties	140
5.6	Decision aiding process dialogue	147
5.7	Dialogue statements (\mathcal{DS})	150
5.8	Locutions rules for a decision aiding dialogue	151
6.1	Performance Table	162
6.2	Examples of statements acceptability (Figure 6.19)	172
E.1	Examples of statements and their dialectical status	208
E.2	Locutions	209

INTRODUCTION

Context study

Deciding is not an easy task: we are often facing problem situations where we are not sure what to decide (it is the case particularly when several points of view can be taken to assess the quality of the different alternatives). Decision analysis is concerned with the process of providing decision support in such situations. We call such an activity “decision aiding”. Decision aiding is the result of an interaction between an “analyst” (or expert) and a “client” (or decision maker), where the aim of the analyst is to guide the decision maker to find a solution to his problem and to be convinced that this solution is a reasonable one. For that, the analyst and the decision maker are committed in an interactive process, where the main objective is eventually to reach a consensus upon the solution to a decision problem. Systems which aim to assist 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 (MCDA) 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 *explanation* that the proposed solution is indeed the best (see [Bouyssou *et al.*, 2000], [Belton and Stewart, 2002]).
- ▷ it should be possible, for the client, to *refine*, or even *contradict*, a given recommendation. Indeed, the decision-support process is often constructive, in the sense that the client refines its formulation of the problem when confronted to potential solutions.

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

the following examples:

- ▷ Ann is not an experienced analyst, but she has good knowledge of some decision-support tools that she used herself 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.

This means 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) which 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 device 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), 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 there was until recently very little attention in the decision analysis literature 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, studying for instance the impact on the efficiency of recommendations when greater interaction with the user is allowed, or when explanations are provided [Pu and

Chen, 2007]. Due to 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 options as opposed to another one).

Our perspective in this work is different, as it is not to fully automatize the process. Instead, we rely on existing decision-aiding tools as developed in the MCDA community, and seek to enhance them with additional features as discussed above. Under such a perspective we consider that *argumentation theory* provides a useful framework within which develop such a dialogue.

A Motivating Example

Suppose that a person wishes to buy a watch. The problem is that once in the store, the person is faced with a big choice of models with different colours, sizes, and prices. Impressed and afraid of making mistakes in the choice, she decides to ask the seller for help. The seller tries to understand what the customer wants and what are her preferences. After a brief discussion, he notes that regarding the size, she prefers a small watch than a medium or a big one; she prefers also the steel to the leather. As for the colour, she specifies that she likes the white more than the red or the pink, and that the watch should be fashion than classical or sport. Finally, the model should be the less expensive possible, but she would like to have a guarantee against breaking along with the watch. From this information, four models were selected and their characteristics are in the table below.

	Size	Material	Price	Colour	Style
<i>a</i>	small	Steel	450	Red	Classical
<i>b</i>	big	Leather	300	White	Fashion
<i>c</i>	medium	Steel	320	Pink	Classical
<i>d</i>	small	Leather	390	Pink	Sport

The following dialogue was recorded:

1. **Expert:** Given your informations, *b* is the best option.
2. **Client:** Why is that the case?
3. **Expert:** Because *b* is globally better than all other options.
4. **Client:** What does that mean?
5. **Expert:** Well... *b* is top on a majority of criteria considered: the price, the colour, and especially the style, it is so trendy!

6. **Client:** But, why b is better than c on the price?
7. **Expert:** Because c is 20 euros more expensive than b .
8. **Client:** I agree, but I see that the guarantee is very expensive especially for this watch. In fact I'm not sure to want the guarantee.
9. **Expert:** But c remains 5 euros more expensive than b .
10. **Client:** I see, but this difference is not significant. And also I changed my mind: I would rather to have a classical model, I think it's more convenient for a daily use.
11. **Expert:** OK. In this case I recommend c as the best option.
12. **Client:** ...

Let us briefly analyse this dialogue. In turn (1) the expert (here the seller) suggests to the client that b would be the best option for her. The client challenges this proposition, in turn (2) and asks for a justification, which is given by the expert in the turn (3). The justification is based on the fact that the option is better than any other one. Not fully satisfied with this explanation, the client asks the expert to be more explicit on the reasons motivating his choice. Thus, the expert, in turn (5), explains that b is ranked first on the majority of criteria considered. But, in turn (6), The client seeks clarifications on the fact that b is better than an other option on a specific criterion. The expert explains that this is due to the fact that the price of c is greater than the price of b . We note, that this explanation is different from the one given at the turn 5. In fact, unlike the turn (4) where the client wanted to know why b was declared as being the best choice, in Turn (6) he is interested in comparing the model b to another model on a particular criterion. Thus, in turn (5), the seller highlights more explicitly the set of positives points in favour of b regarding the set of all options. In the second case, i.e. the turn (6), the expert gave more details on the comparison between two specific models from a specific point of view. Confronted, now, to such explanation, the client rejects it by indicating that the comparison is not appropriate because she doesn't want to include the guarantee in the price, actually. However, in turn (9), the expert maintains that c cannot be better than b because its price is still expensive comparing to the price of b . The client, in turn 10, indicates that the difference is not significant for her and at the same time, she mentions that she changes her mind on her preferences on the style of the watch. This need to refine or correct old information is very common in practice, because a decision maker never fully aware of what he wants or what he prefers, at the beginning of the process. Finally, taking into account the remarks of the client, the expert suggests that, now, c is the better choice.

This example dialogue illustrates how different types of explanations can be asked (and be provided), how the available information may change and be corrected (because the decision-maker

really changes his mind, but also because the expert necessarily make some assumptions that only hold by default, both aspects are involved in the exchange about the price of the watch). This is especially true when the decision-maker is confronted to explicit justifications because it helps him to identify relevant questions, and possible critics.

What this PhD is *not* about

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 hiking on planet Mars), AI researchers pointed out the need to deal with missing or incomplete information, to revise some objectives to adapt to the current context, 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 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 the opportunity to propose alternative actions, for example. Under such a perspective, preferences result from these underlying motivations. The approach is attracting because it offers a natural and powerful way to express, to specify agents’ preferences, because it naturally cater for partial specification of preferences, and because makes explicit many aspects that are usually somewhat hidden in decision models. In the present work, we will not base our model on *cognitive attitudes* and try to represent the underlying motivations of agents, but we rely on information that we suppose provided by the decision maker.

A second very influential contribution of AI (related to the previous point) has been to develop techniques to reason in the presence of conflicting (possibly heterogeneous) information. In particular, one trend of argumentation theory follows the seminal contribution of [Dung, 1995] and studies exhaustively the theoretical properties of “abstract” argumentation systems. Different semantics are proposed, which interpret differently what (sets of, in this case) arguments should be considered acceptable, only based on their interaction. Although part of this trend is relevant to our work, as we shall see in the rest of this document, it is by no means central.

Firstly, one important distinctive feature is that in the present work, we will not assume the existence of a fixed knowledge base to construct arguments, which departs from the classical Dungian framework (see also [Gordon *et al.*, 2007] for a similar remark). Indeed, as illustrated on our example, we are often faced with decision situations where we are far from knowing with

certainty all data and information necessary to make this decision. Thus, we build our conclusion on the basis of available information at that moment and we reserve the right to change it at any time. Thus, argumentation in our context will be viewed as a process to produce justifications of the recommendations of a decision problem.

More fundamentally, this work should not be regarded as a theoretical contribution to argumentation-based reasoning. In particular we shall not propose new methods, based on argumentative notions, to aggregate the preferential information as provided by the decision-maker. These methods, when simply understood as techniques to aggregate “pro” and “cons”, can be expressed as multicriteria methods. The added-value of argumentation, as we shall see, clearly lies in the use of possible complex chains of attacks, when some information can invalidate or reconstitute previous statements. However, due to their recent development, very few experiments exist to validate these underlying principles with human decision-makers [Madakkatell *et al.*, 2009]. In the context of our study, and in the applications that we foresee, the objective is clearly to build upon decision-support tools that clients understand well, find valuable, and would be reluctant to drop for a completely new tool. This explains our choice to concentrate the use of argumentation on the new features that we propose to integrate, leaving aside the aspects that are already well captured by current tools.

This work has also a clear limitation regarding the decision-aiding perspective. According to the formal model of a decision aiding process as described by [Tsoukiàs, 2007], four outcomes summarise the overall process: (i) situation (offering a representation of the problem situation), (ii) formulation problem (the analyst may provide the decision maker with one or more problem formulation(s)), (iii) evaluation stage (organise the available information in such a way that it will be possible to obtain a formal answer to a problem statement), and (iv) recommendation (translate the output into the decision maker’s language). It is clear that the four outcomes of such process are equally important for decision aiding and that conducting such process is not a linear process in which the four models are established one after the other. In this work however we will not consider the whole process but will concentrate on the evaluation stage and the recommendation that we can provide to the decision maker. For the sake of simplicity, we will also assume that the two first stages (situation and formulation problem) are done by taking care of the decision maker cognitive profile and of the different aspects of the decision problem.

What this PhD is about: main contributions

The kind of system that we foresee and sketch here would then allow: (i) to present a recommendation that can be explicitly justified; (ii) to revise any piece of reasoning involved in this process, and be informed of the consequences of such moves; and (iii) to stimulate the user by representing naturally (sometimes visually) the exchanged arguments, possibly generating

counter-arguments. These different aspects trigger different problems and involve different models and mechanisms. As a consequence, our contributions in this work are of different nature, from the modeling exercise to the implementation of a prototype exhibiting the different features discussed in this thesis. More specifically, the following points have been investigated:

A hierarchical structure of arguments. We propose a first approach allowing to specify in argumentative terms the steps involved in an evaluation stage of a decision aiding process [Ouerdane *et al.*, 2008]. To do that, we make use of the popular notion of argument schemes. Argument schemes are forms of arguments that capture stereotypical patterns of humans reasoning, especially defeasible ones. Different approaches have investigated the use of argument schemes to decision-making [Ouerdane *et al.*, 2007, 2009a]. They greatly extend our understanding of the construction of argument schemes for action. Our claim in this work is that by presenting the reasoning steps of an evaluation process under the form of an arguments schemes it makes justification possible and offers in turn the possibility to identify how these steps can be dialectically defeated. Thus, we propose a hierarchical structure where we identify three levels of argument schemes that are embedded. At the highest level the *multi-criteria level*, which is based on the *aggregation level*, which is in turn based on the *Unicriteria level*.

Choosing an aggregation procedure. One of the problems encountered during an evaluation process is the choice of the aggregation procedure for solving the decision problem. In fact, a delicate step in decision aiding is to identify what kind of model or aggregation mechanism is appropriate in the decision context considered. However, multi-criteria analysis provides a variety of aggregation procedures and each one makes a number of assumptions regarding the decision maker's preferences, not always easy to identify. Thus, on what basis the system will choose a procedure to provide recommendations? An idea is to advance methods that are *simple to understand for the user*. When the user provides more preferential information (during the interaction), the system should be able to adapt and jump to the new favoured method. To do that, it will use a *Concept Lattice* in order to guide its choice. Such a lattice is constructed on the basis of a set of properties, that we identified, allowing to distinguish each procedure from another one.

Generating minimal explanations. At each step of the evaluation process, the decision maker can ask for explanations to clarify the reasoning steps assumed by the system, as well as the constructed recommendations and solutions. The idea is that the system generates automatically explanations by instantiating argument schemes constructed in the hierarchy. Moreover, the system will seek to provide the most "simple", "natural", explanation for a given recommendation. More precisely, the system will provide the *minimal evidence which allows to support a given*

conclusion. Thus, in this work we propose to examine different definitions of “minimality”, depending on the language used to construct the explanation.

Handling revisions and updates. An important task for the system is to manage the various changes and updates occurring during the dialogue. In fact, there are very different reasons to revise in such a process: in some case the user may simply want to correct/refine one of its previous statement, introduce new information. In other cases it will contradict one of the system’s assumption. To inform the user of the consequences of these changes, we propose to endow the system of different tools, such as an *acceptability function* and a mechanism to construct an *arguments graph* during the interaction [Ouerdane *et al.*, 2009b; Ouerdane, 2009]. The acceptability function will provide a way for evaluating the exchanged arguments during the dialogue, and therefore the decision for which such arguments were constructed. The graphical representation of the arguments facilitates the understanding of their structure and helps to see “points of attack” in criticizing an argument .

Managing the dialogue. Finally, our system should be able to structure the dialogue on a formal basis in order to be able to control the interaction with the user. To account for that, we propose to construct a dialogue game, that specifies the rules and conditions under which we can have a coherent dialogue in a decision aiding context. More precisely the dialogue is based on recent extensions that incorporate argument schemes within dialogues. [Reed and Walton, 2007]. Such a dialogue will allow the system to manage and guide the interaction with the user.

Plan and overview of the thesis

The thesis is divided into two parts. The first part *Literature review*, containing Chapter 1, Chapter 2 and Chapter3, adresses on the one hand the different concepts and tools related to decision aiding process and argumentation theory, and on the other hand different argumentation-based decision making approches. The second part *MCDA: a dialectical perspective*, containing Chapter 4, Chapter 5 and Chapter 6, is devoted to present our proposals within this thesis.

Chapter 1. Multi-criteria Decision Aiding (MCDA). The aim of this chapter is to give an introduction to the Multicriteria Decision Aiding (MCDA) field by presenting some important notions and definitions. The first part is dedicated to a model of the “decision aiding process”: the interaction between a decision maker and an analyst where the aim of the analyst is to guide the client to find a solution and to convince him that this solution is reasonable. The second part deals with basic elements of MCDA, such as: actions, criteria and problem statements. We devote the next one to present different concepts related to the representation and aggregation

of decision maker's preferences. Finally, in the last part we discuss some limits of the model of decision aiding process.

Chapter 2. Argumentation Theory. The aim of this chapter is to give an overview of different concepts and notions of an argumentation process. We introduce the general idea of non monotonic reasoning and discuss the different steps of an argumentation process. Indeed, different levels can be associated to such process: the logical level which provides the logical structure of a single arguments; the dialectical level which address the notion of conflict between arguments and the procedural level which introduce how we can use arguments within a dialogue. Our purpose, through such levels, is to address the different concepts required to our work, such as argument schemes and critical questions. The last part of this chapter introduces the representation and the visualization of arguments by graphs structure.

Chapter 3. Arguing for decision: a *critical review* . There has recently been many proposals to adopt an argumentative approach to decision-making. As the underlying assumptions made in these different approaches are not always clearly stated, we review these works, taking a more classical decision theory perspective, more precisely a multi criteria perspective. It appears that these approaches seem to have much to offer to decision models, because they allow a great expressivity in the specification of agents' preferences, because they naturally cater for partial specification of preferences, and because they make explicit many aspects that are usually somewhat hidden in decision models.

Chapter 4. Different levels for different reasons. Our ambition in this chapter is to show that it is possible and useful to introduce argumentation in a decision aiding process. In fact, we propose to specify in argumentative terms the steps involved in a multi criteria evaluation process. To do that, 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.

Chapter 5. Towards an argumentation-based decision aiding tool. The aim of this chapter is to present a first dialectical system for designing the interaction between an analyst and a decision maker during the evaluation process. Such a system is based on the popular model of dialogue games, and more precisely it is based on recent extensions that incorporate argument schemes within such models. It has the aim to automatically generate explanation for the recommendation of a decision problem. Moreover, it includes methods and tools to calculate the status of arguments, critical questions , acceptability functions, to establish how the revision procedure

can be handled. Finally, to inform the user of the consequences of changes during the dialogue (e.g. adding new information, challenging an assumption...) a discussion graph with tree like structure is used. The graph allows to visualize and evaluate the exchanged argument during the dialogue.

Chapter 6. *ArgDEC*: a prototype for an argumentation based decision aiding tool. The aim of this chapter is to show how the different elements and concepts presented in this work, can be used in a “real” application. To do that, we implemented a system realised under the form of a simple GUI (Graphical User Interface) in Java. Such a system allows to simulate a dialogue between an analyst (here played by the system) and a decision maker in a decision aiding situation. In general, the system will allow to: structure the decision problem, calculate the recommendation and to generate automatically the minimal explanation that allows to justify it, select the proof standard on the basis of the rules and properties described in the previous chapters; and finally to inform the user of changes during the interaction by both a textual and a graphical representation.

Part

Literature Review

1 Multi-criteria Decision Aiding (MCDA)

The aim of this chapter is to give an introduction to the Multicriteria Decision Aiding (MCDA) field by presenting some important notions and definitions. The first part is dedicated to a model of the “decision aiding process”: the interaction between a decision maker and an analyst where the aim of the analyst is to guide the client to find a solution and to convince him that this solution is reasonable. The second part deals with basic elements of MCDA, such as: actions, criteria and problem statements. We devote the next one to present different concepts related to the representation and aggregation of decision maker’s preferences. Finally, in the last part we discuss some limits of the model of decision aiding process.

1.1 Introduction

Decisions are not just an “act of choice”, they are the result of a “decision process”, a set of cognitive activities enabling to go from a “problem” (a state of the world perceived as unsatisfactory) to its “solution” (a state of the world perceived as satisfactory, if any exists). Even if we consider at the place of a human decision maker an automatic device (such as a robot or other device with some sort of autonomous behavior) we can observe, describe and analyze the process through which a “decision” is reached. However, it is clear that it is not a process only about solving a problem: a decision process implies also understanding and shaping a decision problem.

On the other hand research conducted in what is known as “Problem Structuring Methodologies” ([Rosenhead, 1989; Franco *et al.*, 2006; Shaw *et al.*, 2007]) emphasized that decision aiding is not just to offer a solution to well established mathematically formulated problem, but to be able to support the whole decision process, representing the problem situation, formulating a problem and possibly constructing a reasonable recommendation. In other terms to the concept of decision

process we can associate the concept of “decision aiding process” ([Tsoukiàs, 2007]).

A decision aiding context is quite different from the context of decision making. In fact they are two different situations¹ [Tsoukiàs, 2008; Bouyssou *et al.*, 2006]. In a *decision making* context we are only concerned by the decision maker activities and the hypothesis that he is endowed with decision power and therefore he is responsible for the decision to make. In such a setting, the decision maker might use a decision theoretic tool in order to establish potential actions to undertake. In a *decision aiding* context, however, there is at least two distinct actors: the client and the analyst, both playing different roles with respect to the concern of the client. As it was emphasized by [Tsoukiàs, 2007]:

“A decision aiding context only makes sense with respect to one or more decision process, the ones where the client’s concerns originate. The set of activities occurring within such a setting is called “decision aiding process”. The ultimate objective of this process is to arrive to a consensus between the client and the analyst. On the one hand the client has a domain knowledge concerning the decision process. On the other hand the analyst has a methodological knowledge, that is domain independent”.

In this chapter we will try to introduce a general description of what such a decision aiding process is and what are the tools and the models used within this process in order to construct a recommendation. We note that we are concerned only by models using multiple criteria.

The chapter is organized as follows: In Section 1.2, we address a model of decision aiding process. Section 1.3 introduces three basic elements of multi-criteria decision aiding: actions, criteria and problem statements. In Section 1.4 we address the steps involved in handling the preferences: preference modeling, preference aggregation and exploiting preferences. In Section 1.5, we discuss the problems encountered in the model of decision aiding process.

1.2 A model of Decision Aiding Process (DAP)

We present in this section a model of a decision aiding process [Tsoukiàs, 2007]. Such a process represents the interaction between at least two actors: an analyst and a decision maker where the aim of the analyst is to guide the client to find a solution and to convince him that this solution is a reasonable one. This model can be described and characterised by four artefacts:

- ▷ a representation of the problem situation;
- ▷ a problem formulation;

¹Other works discussed the difference between decision aiding and decision making [Roy, 1993; Brown, 1989].

- ▷ an evaluation model;
- ▷ a final recommendation.

In what follows we discuss each artefact. To illustrate the different models, we will use the Example 1.1, borrowed from [Stamelos and Tsoukiàs, 2003], where the authors present an application of these concepts in practice.

Example 1.1. (*Selection of a Billing system*)

A new mobile telecommunication operator has been established in a small but highly competitive European market. One of the basic operational tools of such companies is their billing system (BS). This system allows both a structured accountancy of the traffic and a flexible policy towards the existing and potential clients.

Some years after the establishment of the company the necessity to upgrade or to substitute the existing billing system became evident to the management. A decision process has therefore been triggered and the authors have been asked to provide decision support.

1.2.1 The problem situation

The first model of the process allows to represent the problem situation for which the decision maker has asked the analyst to intervene. This representation is the result of an effort at replying to questions of the type: Who has a problem? Why is this a problem? What is important for the decision maker? etc. The construction of such an artifact allows, on the one hand the client to better understand his position within the decision process for which he asked the decision support and on the other hand the analyst to better understand his role within that decision process.

A representation of the problem situation can be conceived as a triple:

$$P = \langle \mathcal{A}, O, S \rangle$$

where :

- ▷ \mathcal{A} is the set of participants to the decision process;
- ▷ O is the set of objects (or stakes) of each participant brings within the decision process;
- ▷ S is the set of resources the participants commit on their stakes and the other participant's stakes.

Example 1.2. (*Example 1.1 cont.*)

- ▷ *The participants involved were: the acquisition manager; the information system manager; the marketing and sales manager; ...*

- ▷ *The objects involved in the process were: the market share of the company; the policy towards the suppliers; the company's internal organisation and the billing system itself.*
- ▷ *The resources implied in the process included the necessary funds for the BS, the knowledge about the BS and the relation with the software suppliers.*

1.2.2 Problem formulation

Given a representation of the problem situation, the analyst may provide the decision maker with one or more problem formulations. A problem formulation reduces the reality of the decision process within which the decision maker is involved to a formal and abstract problem. The result is that one or more of the decision maker'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. A problem formulation can be conceived as a triplet:

$$\Gamma = \langle \mathbb{A}, \mathbb{V}, \Pi \rangle$$

where:

- ▷ \mathbb{A} is the set of potential actions that the client may undertake within the problem situation as represented in \mathcal{P} ;
- ▷ \mathbb{V} is the set of points of view under which the potential actions are expected to be observed, analyzed, evaluated, compared, including different scenarios for the future;
- ▷ Π is the problem statement, the type of application to perform on the set \mathbb{A} , an anticipation of what the client expects.

Example 1.3. *(Example 1.1 cont.)*

The strategic decision with which the management was faced consisted of choosing one among the following options: upgrade the existing BS, buy a BS created ad-hoc for the company by an external supplier, develop an ad-hoc BS in collaboration with an external supplier. However the management was not able to choose an option without analysing what the billing system would be eventually in all such options. Therefore, three problem formulations were provided, but here we present only the one concerning the developing option.

- ▷ \mathbb{A} : *a set of suppliers whom it could be possible to co-develop a new BS.*
- ▷ \mathbb{V} : *costs, requirement analysis and satisfaction, timing, benefits for the company, etc.*
- ▷ Π : *selection of a co-developer to establish a co-makership policy and therefore a long-term collaboration.*

1.2.3 Evaluation Model

For a given problem formulation, the analyst may construct an evaluation model, that is to organise the available information in such a way that it will be possible to obtain a formal answer to a problem statement. An evaluation model can be viewed as an 6-tuple:

$$\mathcal{M} = \langle A, D, E, H, U, R \rangle$$

Where :

- ▷ A is the set of actions on which the model applies. It establishes the universe of discourse of all relations and functions which are going to be used in order to describe the decision maker's problem.
- ▷ D is the set of dimensions (attributes) under which the elements of A are observed, described, measured, etc
- ▷ E is the set of scales associated to each element of D .
- ▷ H is the set of criteria under which each element of A is evaluated in order to take in account the client's preference.
- ▷ U is a set of uncertainty structures. Depending on the language adopted, U collects all uncertainty distributions or the beliefs expressed by the client.
- ▷ 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.

Example 1.4. (*Example 1.1 cont.*)

It is clear that each problem formulation may generate a quite different evaluation model. For instance for the developing option, the main elements of the evaluation model are:

- ▷ A : *co-developing suppliers;*
- ▷ H : *the implication of the information systems departement in the developement process, the benefits of "selling" the new billing system, ...*
- ▷ R : *procedure for a choice problem.*

1.2.4 Final recommendation

The final recommendation is the final deliverable which translate the output of the evaluation model into the decision maker's language. In other words, the final recommendation should be

able to translate the conclusion of the decision aiding process into a format that can be used within the client's decision process. Moreover, the nature of this recommendation depends on a central element which is the problem statement Π , that has been agreed upon the problem formulation stage of the decision aiding process. This will be discussed further in Section 1.4.3.

Assumption 1.1. *In this work, we focus on the evaluation model and the final recommendation. Moreover, we will consider only evaluation model with multiple criteria.*

In what follows, we detail the main tools and concepts used in the decision aiding process.

1.3 Basic concepts of MCDA

Three concepts usually play a fundamental role for analyzing and structuring the decision aiding process, namely: actions, family of criteria and problem statements. The presentation of these concepts in the three next subsections is obviously succinct. It aims to introduce the different elements used in this work. For more details, we refer the reader interested, in the literature on this domain [Vincke, 1992; Figueira *et al.*, 2005; Roy and Bouyssou, 1993; Bouyssou *et al.*, 2006; Roy, 1996; Keeney and Raiffa, 1976; Schärliig, 1985; Vincke, 1992; Roy and Bouyssou, 1993; Belton and Stewart, 2002; Belton and Pictet, 2001].

1.3.1 Set of actions

When facing a decision problem, the first step may be to identify the different objects submitted to the decision aiding process. Depending on the step of such process, these objects can be set of stakes (at the problem situation), a set of potential actions (at the formulation problem) or a set of actions (at the evaluation model). According to the hypothesis 1.1, we are interested by the last one, i.e. set of actions, which can be defined by:

- ▷ listing its members when it is finite and sufficiently small for a possible enumeration; or
- ▷ stating the mathematical properties or characterization when the set is too big or infinite.

In this work we shall consider only finite set of actions and it is noted A .

1.3.2 Criterion and family of criteria

A criterion is a tool constructed for evaluating and comparing actions according to a point of view which must be (as far as possible) well defined. This evaluation must take into account for each action a , all the relevant effects or attributes associated to the point of view considered.

Formally,

Definition 1.1. [Bouyssou et al., 2006]

A criterion is any dimension to which it is possible to associate a preference model, even a partial one, such that the decision maker should be able to make a choice along this single dimension.

This definition highlights a clear distinction between an **attribute** (or a dimension) and a criterion. The former represents the “empirical” knowledge available or collected about A , but says nothing about the preferences of the decision maker. The fact that such knowledge may use a structure such as an order does not establish any knowledge about the client’s “desires”. These are modeled in a criterion where preferences are explicitly represented. The example 1.5 illustrates such distinction.

Example 1.5. *Let us take three objects which can be described according to the attribute “colour”, as is shown in the table below.*

	<i>attribute</i>
<i>Object₁</i>	<i>black</i>
<i>Object₂</i>	<i>red</i>
<i>Object₃</i>	<i>yellow</i>

*In the absence of any other information, it is clear that we are not able to express a preference for any of them. To do that, we need to build a **criterion** “colour” by constructing a preference relation (see subsection 1.4.1) among the different values of such attribute.*

For instance, one can say that “red” is preferred to “yellow” which is preferred to “black” and “red” is preferred to “black”. According to this information we can conclude that the second object is preferred to the first which is in turn preferred to the last object.

Thus, a criterion plays an important role in the process of actions evaluation. Indeed, the construction of the set of criteria or the *Family* $H = \{h_1, h_2, \dots, h_n\}$ of criteria is a central activity in the decision aiding process [Bouyssou, 1990; Roy and Bouyssou, 1993]. It can be either the result of a direct process (creating from dimensions through direct questioning of the client) or of an indirect process (establishing criteria “explaining” global preferences expressed by the client on examples or already known cases [Bouyssou et al., 2006; Greco et al., 2008; Jacquet-Lagrèze and Siskos, 2001]). Moreover, it is possible to have preferential information on the set of criteria, such as degrees of importance, an order on all criteria, or a weight associated with each criterion. Such information reflect either the fact that the decision maker attaches a particular importance to each criterion (e.g. in Lexicographic method) or that it exists trade-offs between criteria (see Weighted sum for example)

1.3.3 The choice problem statement

In general, when searching for a solution in decision problems, we model the situation such that the set of considered actions is fixed once for all, such that every solution is exclusive from the others and such that solutions can be ranked incontestably from the worst to the best. However, the set of actions doesn't necessarily fulfil these three characteristics. This is the reason why it is sometimes preferred to analyze the problem differently. Having the set of actions A and a set of criteria H , a decision maker may be facing different problem statements [Roy and Bouyssou, 1993; Roy, 1996; Bana e Costa, 1996; Vincke, 1992].

To identify the different problem statement, we adopt a classification which is based on the partitioning of the set A [Bouyssou *et al.*, 2006]. Depending on:

- ▷ whether or not the categories are predefined;
- ▷ whether or not the categories are ordered;
- ▷ whether or not the number of categories is predefined.

Thus, several problem statements arise. Examples are presented in Table 1.1.

Category	Predefined	No Predefined
Ordered	Sorting	Ranking
No ordered	Classification	Clustering

Table 1.1 — Examples of problem statements

In the rest of this work, we will focus only on one problem statement, namely, **the choice problem**. The fact of identifying the nature of the problem to resolve allows to anticipate what is expected to be done with the elements of A . Then, having a set of actions A and a set of criteria H , the description of the problem is as follow.

The choice problem statement : we have at most two categories that are ordered and not predefined. The first category will contain a subset of actions $A^0 \subset A$ considered as the best according to the criterion-set H and the second one all the remain actions $A \setminus A^0$

In other terms, the choice problem statement leads to the construction of a *choice set*, noted $C(S)$, such that S is the global preference relation built on A using an aggregation function (see subsection 1.4.2). Such set will include the “best” elements of A according to the decision maker's preferences. The construction of such a set is done either directly, because it is possible to distinguish clearly the best elements or indirectly by using special procedures (see subsection 1.4.3). However, the construction must meet certain conditions. In fact, the choice set should:

- ▷ be as small as possible given the available information,
- ▷ be such that there are clear “arguments” to justify the elimination of the alternatives in $A \setminus C(S)$, i.e., the alternatives which are not selected;
- ▷ be such that there is no built-in bias in favour of some alternatives, i.e., that the only argument that can be taken into account in the determination of $C(S)$ are how these alternatives are related in terms of the relation S .
- ▷ be monotonic, i.e., if $a \in C(S)$ and S' is identical to S except that $aS'b$ and $a \neg Sb$ or $b S'a$ and bSa , for some $b \in A$ then we should have $a \in C(S')$.

Within this work, we are interested by the second condition for building a choice set, i.e., the existence of arguments for justifying why certain actions are in the choice set and not others. In fact, we aim to build formal explanations justifying the fact that an action belongs to the choice set. In other terms, to explain why an action is the recommendation of the considered decision problem. We shall see in Chapter 5 that we can have different types of explanations, depending of the technique used to construct such recommendation. Two techniques are presented in subsection 1.4.3

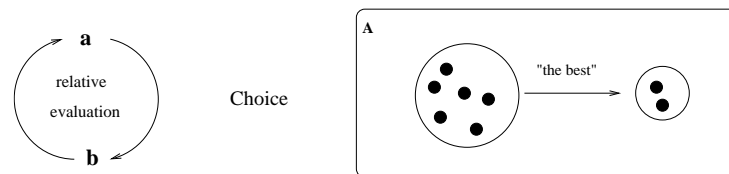


Figure 1.1 — Relative evaluation

On the other hand, as shown in Figure 1.1 (adapted from [Bana e Costa, 1996]), the choice problem leads to a *relative or pairwise evaluation*. Thus, the selection, in the choice problem, is to be made on the sole basis of the comparison of actions. In other terms, instead of judging the consequences of an action independently of the others, we try to construct binary relations among the set of actions, translating therefore the evaluation of the consequence of each action in comparison with the others. Thus, the “best” actions is not defined with respect to outside norms but with respect to other actions.

Moreover, there exist another way to evaluate an action, which is the *intrinsic or absolute evaluation*. In this case, the idea is to partition the set of alternatives into several categories, by comparing each action to some pre-established norms, called limited profiles or reference points, noted P . Such evaluation is used, for instance in *Sorting problem statement* but also in argumentation.

In fact, argumentation relies, in general, on the intrinsic evaluation for constructing arguments in favour and against a given action (see Chapter 2). The idea is to judge whether the action has good or bad consequences, independently of the others. In decision models, this would typically be done by using an ordered scale defining the different values that can be used to assess the action (for instance, marks from 0 to 20 for students). Now what counts as a positive or negative outcome is specific to each decision maker, and depends of its (subjective) preferences. That is, you must classify the outcome of the actions. In decision models, one approach is that the decision maker uses an evaluation scale and specify a frontier, that is, a *neutral zone (or point)*, thus inducing a bipolar scale. On the basis of this scale, the positive part allows us to conclude that that action has good (resp. bad) consequences because it is better (resp. worst) than the neutral point.

The concept of “bipolarity” in scales measuring value is not really new in the literature. [Rescher, 1969] has been the first to introduce this concept. [Roy, 1991] has introduced the concept of concordance/discordance in Multiple Criteria Decision Analysis (through the outranking procedures) and [Tsoukiàs and Vincke, 1997] used a specific logic formalisms in order to extend preference models under the presence of positive and negative reasons, among others. In this work, the concept of bipolarity refers to the existence of two independent types of information, positive and negative. The first provides support to the action and the second allows to express a disagreement against this action. Such information will correspond to two different catégories separated by a single point.

Assumption 1.2. *We only consider choice problem statement.*

1.4 On the notion of preferences

Decision support is based on the elaboration of preferential information. The basic idea in decision aiding methodology is that, given a decision problem, we collect preferential information from the decision maker such that his/her system of values is either faithfully represented or critically constructed and thus we are able to build a model which, when applied, should turn a recommendation for action to the decision maker. Then the fundamental step in decision aiding is the modeling and the representation of the decision maker’s preferences on the set A [Bouyssou *et al.*, 2006; Roubens and Vincke, 1985; Oztürk *et al.*, 2005; Keeney and Raiffa, 1976; Vincke, 1992; Brafman and Domshlak, 2007].

Furthermore, handling the preferences of a decision maker in a decision aiding process implies going through the following steps:

- ▷ *Preference learning.* Acquire from the Decision Maker preferential information under the form of preference statements on a set of “alternatives” A . Such statements can be on single

attribute comparisons or assessments (“I prefer red shoes to brown shoes”; “red shoes are nice”) or multi-attribute ones (“I prefer shoe x to shoe y ”; “ x is a nice shoe”, x and y being vectors of information on a set of attributes). Possibly such statements can carry some further quantitative information or take more complex form: “my preference of x over y is stronger than the one of z over w or twice stronger” etc.. Problems arising here include what to ask, how to ask, what rationality hypothesis to do about the decision maker, what degrees of freedom allow to the decision maker’s replies, how much the interaction protocol influence the decision maker (see more in [Bana e Costa and Vansnick, 1994; Blum *et al.*, 2004; Dias and Mousseau, 2006; Greco *et al.*, 2008; Ha and Haddawy, 2003; Haddaway *et al.*, 2003; Jacquet-Lagrèze and Siskos, 2001; Salo and Hamalainen, 2001; Sandholm and Boutilier, 2006; Wang, 1994]).

- ▷ *Preference Modeling*. Transform the preference statements in models. These can take the form of binary relations on the set of actions A , on the set $A \times A$, on the set $A \times P \cup P \times A$, (P being a set of reference points) or of functions [Keeney and Raiffa, 1976; Oztürk *et al.*, 2005; Roberts, 1979; Roubens and Vincke, 1985]. Once again the models may concern single or multiple attributes.
- ▷ *Preference Aggregation*. In case we have several attributes on which we constructed preference models we may consider the problem of aggregating such preferences in one single model [Roy and Bouyssou, 1993; Roy, 1996; Vincke, 1992]. It is the typical problem of both social choice and multi-attribute utility theory. There exist several procedures and methods proposed for this purpose. We discuss some of them in subsection 1.4.2.
- ▷ *Exploiting Preferences*. Constructing a preference model (either directly or through preference aggregation) does not necessarily imply that we can get an “operational result”. That is we do not necessarily have an “order” such that we can identify a subset of maximal elements or at least a partial ranking etc. It might be that it is necessary to make some further manipulation in order to get such a result. To do that, several procedures have been suggested in the literature [Bouyssou and Vincke, 1997; Vincke, 1992]. Some examples are presented in subsection 1.4.3

In what follows, we present different notions and concepts related to preferences, such as preference structures, aggregation procedure, etc. Let us remark that we will address only the concepts relevant for this work. Therefore, the learning phase will not be addressed.

1.4.1 Preference modeling

Preference modeling aims to construct preference relations on a set of actions which are evaluated with respect to a criterion. The representation of the decision maker’s preferences over

the set A of actions constitutes a crucial step in decision aiding. Depending on the context of the problem, the nature of information that we are able to handle and the expectations of the decision maker, different situations may appear. In the context of decision aiding, [Bouyssou *et al.*, 2006; Vincke, 2001] distinguished two type of problems called “comparison problem” and “numerical representation problem”.

- ▷ In the *comparison problem*, alternatives are assessed according to different points of view. Such assessment may be of different nature: symbols, linguistic expressions, numbers, etc. Naturally, some additional information about the nature of the points of view, the scale type used and the meaning that the decision maker gives to such evaluation may be added. The aim is to define the preference relations capable to represent such assessment with the additional information.

Example 1.6. *suppose that we have to define preference relations between three alternatives a , b and c which are evaluated with respect to their performances such that: the performance of a : 10, the performance of b : 12 and the performance of c : 14.*

Let us imagine that the decision maker gives additional information such that: he prefers one alternative to another if the first one is greater than the second one and if the difference between their performance is greater than three; otherwise he is indifferent between them.

In light of these remarks we can construct, for instance, the following preference relations: $c P a$ which represents the sentence “ c is preferred to a ” and $b I a$ which represents the sentence “ b and a are indifferent”.

- ▷ In the *numerical representation problem*, the decision maker expresses his preferences for each pair of alternatives according to one or several points of view. Preferences can be completed by some additional information like the presence of incomparability or thresholds, etc. The aim is assigning a numerical representation to each alternatives, which is not necessarily a single number (e.g., interval, distribution).

Example 1.7. *Let us suppose that the decision maker gives us his preferences among three alternatives a , b and c such that: he is indifferent between a and b only if the difference between their performances is greater than three; he is also indifferent between b and c under the same condition, and he prefers c to a .*

A possible numerical representation of such informations can be the evaluation given in the Example 1.6.

We draw the attention of the reader to the fact that we do not seek, within this thesis, to build preference models or to address issues related to preference modeling. Our aim through this section is introducing some preferences’ representation that are mentioned in the rest of this work. Such representations can be based on different *preference structures* [Oztürk *et al.*, 2005; Keeney and Raiffa, 1976; Roubens and Vincke, 1985].

1.4.1.1 Preference structures

The mathematical concept of *binary relation* (see Annex A) is communally used as a formal representation of preference relations (models) defined on a finite set since such relations are results of the comparison of two elements.

Thus, considering that A is a finite set, the pairwise comparison of its elements may result in different binary relations having different properties. We say that such relations construct a **preference structure** if they satisfy some conditions:

Definition 1.2. (*A preference structure*) [Oztürk et al., 2005]

A preference structure is a collection of binary relations $\{S_1, \dots, S_m\}$ defined on the set A such that:

- ▷ $\forall x, y \in A \exists i \in \{1, \dots, m\}, xS_i y$ or $yS_i x$, which means for each couple x, y in A ; at least one relation is satisfied;
- ▷ $\forall x, y \in A, xS_i y \Rightarrow \forall j \neq i, \text{not } (xS_j y) \text{ and not } (yS_j x)$, which means for each couple x, y in A , if one relation is satisfied, another one cannot be satisfied.

In other terms a preference structure defines a partition of the set $A \times A$.

A large binary relation, also called an *outranking relation*, denoted by S , can be used to characterise a preference structure. This relation is interpreted as “ a is as least as good as b ”, noted $a \succeq b$ or aSb , such that $\succeq \subset A \times A$.

Definition 1.3. (*$\langle P, I \rangle$ preference structure*)

A $\langle P, I \rangle$ preference structure on the set A is a pair of preference relation (P, I) on A such that:

$$\begin{cases} aPb \text{ (noted } a \succ b) & \Leftrightarrow aSb \text{ and } b \neg Sa \\ aIb \text{ (noted } a \sim b) & \Leftrightarrow aSb \text{ and } bSa \end{cases}$$

The reader will remark that in a $\langle P, I \rangle$ preference structure we admit that the case $a \neg Sb$ and $b \neg Sa$ is not possible, thus implicitly assuming that all pairs of actions are comparable. Indeed, it is easy to check that in such case S is a *complete relation* (see Annex A).

Moreover, it is possible to add some other properties to the relations P and I , allowing to construct various kinds of preference structures. In what follows, we present some examples of such structures or orders, namely *linear orders* and *weak orders*. The definitions presented in this section can be found in [Oztürk et al., 2005].

Definition 1.4. (*Linear order*)

A reflexive $S = \langle P, I \rangle$ on a finite set A , is a linear order if there exists a real valued function h , defined on A , such that $\forall a, b \in A$:

$$\begin{cases} aPb & \Leftrightarrow h(a) > h(b) \\ a \neq b & \Rightarrow h(a) \neq h(b) \end{cases}$$

The linear order structures consists of a ranking of elements from the best one to the worst one without any ex aequo. With this relation, we have an indifference between two elements only if they are identical. In the literature, one can find different terms associated to this structure: complete order, total order.

Definition 1.5. (*Weak Order*)

A reflexive $S = \langle P, I \rangle$ on a finite set A , is a weak order if there exists a real valued function h , defined on A , such that $\forall a, b \in A$,

$$\begin{cases} aPb & \Leftrightarrow h(a) > h(b) \\ aIb & \Leftrightarrow h(a) = h(b) \end{cases}$$

This structure is also called complete preorder or total preorder. It is obtained by considering both P and I transitive. We note that considering I as a transitive relation is empirically falsifiable. Literature studies on the intransitivity of indifference show this; undoubtedly the most famous in that of [Luce, 1956], who gives the example of a cup of sweetened tea². Relaxing the property of transitivity of indifference results in other structures, such as semi-orders and interval orders, that we should not discuss here.

We draw attention to the fact that numerical representations of preferences are not unique. All monotonic strictly increasing transformations of the function h can be interpreted as an equivalent numerical representation.

Assumption 1.3. *Within this work we will use only very simple numerical representation of the type $h_i(a)$: the value of the action a on attribute h_i (called also performance). Thus, the different valuations of the different actions of A on each attribute h_i are represented by a matrix called **performance Table**. Furthermore, we will use very simple preference model of the type aPb representing a preference between two actions and aIb which represents an indifference between the two actions.*

The construction of preference structures or models may concern one attribute or several attributes. In the later case, we may consider the problem of aggregating of such preferences into one single model. To do that, different procedures and functions exist. We present some of them in the next subsection.

²one can be indifferent between a cup of tea with n milligrams of sugar and one with $n + 1$ milligrams of sugar, if one admits the transitivity of the indifference, after a step of transitivity, one will have the indifference between a cup of tea of n milligrams of sugar and that with $n + N$ milligrams of sugar with N large enough, even there is a very great difference of taste between the two; which is contradictory with the concept of indifference

1.4.2 Preference aggregation

The ultimate objective of the decision aiding methodology is to establish a recommendation based on the preferences of the decision maker. In a multiple criteria context, the development of this recommendation requires a multi-criteria aggregation procedure to synthesize preferences on each criterion and contribute to the definition of an outcome [Roy and Bouyssou, 1993]. This aggregation is an important point in the building of an “Evaluation model” (see Section 1.2). The idea of aggregation, as it is illustrated in the Figure 1.2, is either to construct a *global preference relation* $a \succeq b$ or a *global aggregation function* $H(a, b)$. It depends if we have a set of preference relations between alternatives on each dimension $a \succeq_i b$ or a set of evaluation of each alternatives on each dimension $h_i(a)$.

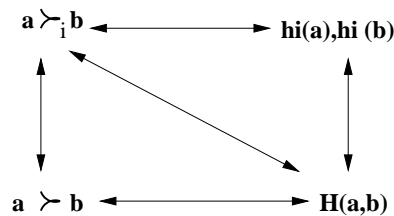


Figure 1.2 — How to aggregate preferences?

Thus, we propose, in this chapter, to describe some aggregation methods. A method can be seen as an operator or mechanism that transform a certain type of input information related to the evaluation of the alternatives on several dimensions into a global output, most of the time a relation. In addition, we will discuss the problem of choosing an aggregation procedure during a decision aiding process. Indeed, the analyst has the delicate task of selecting an aggregation procedure among the variety of procedures that exist. The problem is that this choice is not obvious because it depends on the decision-maker’s preferences that are not always clearly identified. We will see that there are theoretical solutions, identified by the Conjoint Measurement, but these are difficult to implement in practice.

1.4.2.1 Aggregation procedures

We present in what follows a sample of aggregation procedures, namely: the simple Majority method, the weighted majority method, the lexicographical method and the weighted sum procedure. Such procedures are all designed to build a weak order among the set of alternatives by taking into account the information and the evaluation of each action on each criterion. In order to illustrate the different methods, we will use the Example 1.8 (when it is necessary, further information can be added).

Example 1.8. Suppose that we have the following performance table

	h_1	h_2	h_3
a	16	14	16
b	10	18	12
c	18	12	6

The simple majority method. This procedure works as follows. Take any pair (a, b) of alternatives. If the number of criteria such that a defeats b is larger than the number of criteria such that b defeats a , then a is globally preferred to b . If the two numbers are equal, then a is globally indifferent to b . Formally,

$$a \succeq b \Leftrightarrow |\{i \in N : a S_i b\}| \geq |\{i \in N : b S_i a\}|$$

We illustrate this method with the Example 1.9.

Example 1.9. On the basis of the performance table of the Example 1.8, we can construct the following three weak orders.

1. $c \succ_1 a \succ_1 b$;
2. $b \succ_2 a \succ_2 c$;
3. $a \succ_3 b \sim_3 c$.

Such that \succ_i and \sim_i correspond, respectively, to the asymmetric and symmetric part of \succeq on a dimension i .

Alternative a defeats b twice. Therefore, a is globally better than b . Alternative a defeats c twice also. Therefore, a is globally better than c . Alternative b defeats c twice. Therefore, b is globally better than c . Finally, we obtain the weak order $a \succ b \succ c$.

The weighted simple majority. The procedure is a generalization of the classical simple majority. In this procedure, for each criterion i is assigned a weight w_i reflecting its importance. We will say that a is globally at least as good as b if the coalition of criteria supporting a against b is sufficiently strong. The strength of the coalition is the sum of the weights of the criteria such that a is at least as good as b . Formally,

$$a \succeq b \Leftrightarrow W_{ab} = \sum_{i: a S_i b} w_i \geq W_{ba} = \sum_{i: b S_i a} w_i$$

We illustrate this method with the Example 1.10.

Example 1.10. Let the weight of the three criteria be: 3, 1 and 1. We have $W_{ab} = 4$ and $W_{ba} = 1$. Therefore, $a \succ b$. Also $W_{ac} = 2$ and $W_{ca} = 3$. Therefore $c \succ a$. Finally, $W_{bc} = 2$ and $W_{cb} = 3$. Therefore, $c \succ b$. The final ranking is thus: $c \succ a \succ b$.

The lexicographic method. This very simple method works as follows: first, you need that the criteria are linearly ordered (see subsection 1.4.1) and are considered in that order when comparing alternatives. Then you look at the first criterion, if a is strictly better than b on this criterion, then a is declared globally preferred to b without even considering the other criteria. Similarly, if b is strictly better than a on this criterion, then b is considered as globally preferred to a without considering the other criteria. But if a and b are indifferent on the first criterion, you look at the second one. You make the comparison again between a and b and if you still can not make the difference between them, you proceed with the next criterion and so on. Formally,

$$a \succeq b \Leftrightarrow \begin{cases} a \sim_i b & \text{for all criteria} \\ \text{or} \\ a \succ b & \text{for the first criterion } i, \text{ w.r.t the linear order, for which } a \approx b \end{cases}$$

We illustrate the lexicographic method with the Example 1.11.

Example 1.11. Suppose that the second criterion is more important than the first one which is more important than third one. Consider the pair a, b . Begin with criterion 2 on which b is strictly better than a . Therefore, $b \succ a$. Consider now c and b . Alternative b is strictly better than c on criterion 2. So, $b \succ c$. Similarly, $a \succ c$. The final result is thus the linear order $b \succ a \succ c$.

The particularity of the lexicographic method is the existence of the *linear order among the set of criteria* and the fact that each criterion is totally or infinitely more important than all other criteria lower in this order. If a is better than b on the first criterion w.r.t the order, it will be globally better than b , even if b is better than a on ten or one hundred other criteria. No compensation is possible.

The weighted sum. It is a very common and simple aggregation method. For each alternative a score $S_a(h)$ is defined as the weighted sum of its evaluation.

$$S_a(h) = \sum_{i \in N} w_i h_i(a)$$

Once all the scores are calculated, alternatives are then ranked in the decreasing order of their score. The weights w_i have a very precise and quantitative meaning, and can be considered as trade-offs to compensate for a disadvantage of k_i units for criterion j , we need an advantage of k_j on a criterion i . The determination of weight is a tricky problem and involves the participation of

the decision maker and several strategies can be used. For more details about how to determine the weights, the reader can refer to [Bouyssou *et al.*, 2000, 2006].

The Example 1.12 illustrates how the weighted sum works.

Example 1.12. *According to the decision maker h_1 is the most important criterion, h_3 is the second one and h_2 is the less important. A numerical representation of these informations is represented by the following weight-vector $w = (0.6, 0.1, 0.3)$, we can conclude that the score of a (resp. b and c) is 15.8 (resp. 11.4 and 13.8). The ranking is thus $a \succ c \succ b$*

We presented in this section a variety of aggregation functions. Our presentation focused on the operation and parameters of each procedure. In what follows, we address the problem of choosing an aggregation procedure during a decision aiding process.

1.4.2.2 How to choose a procedure?

A delicate step in decision aiding is to identify what kind of model or aggregation mechanism is appropriate in the decision context considered. Indeed, depending on the preference statements provided by the decision maker, it is possible to use certain procedures but not others. However, the main difficulty in choosing a procedure is that it is not easy to know a priori whether the numerical model ($H(a, b)$) of the procedure is the one that fits the model of the decision maker's preferences ($a \succeq_i b$). Each procedure makes a number of assumption regarding the user's preferences structure; that are not always easy to identify.

Thus, on what basis can we choose a particular procedure? One way is to rely on the characterization of the aggregation procedure established by *Conjoint Measurement Theory*:

*“Conjoint Measurement theory examines the conditions under which a relation on a set of objects described by a vector of evaluations is determined by a sort of synthetic measurement that takes the relevant attributes of the objects into account in an appropriate manner [Krantz *et al.*, 1971].”*

This theory was first developed in Economics [Debreu, 1960] and in Psychology [Luce and Tukey, 1964]. Soon after its development, people working in decision analysis realized that the techniques of Conjoint Measurement could be also used to structure preferences [Raiffa, 1969; Edwards, 1971]. Indeed, Conjoint Measurement theory has been useful in finding conditions under which it is possible to build a convenient numerical representation of the global relation \succeq and to study the uniqueness of this representation. In other terms, if a global preference relation satisfies some conditions, then it admits a description within a particular numerical model. Thus, each aggregation function is described by a specific theorem (or characterisation). This theorem includes axioms that represent the conditions under which a procedure can be used. For example, the following theorem characterise the weighted sum procedure.

Theorem 1.1. *Suppose that, for each criterion i and each alternative a , the performance $h_i(a)$ can be any real number. Then, the only aggregation function satisfying Weak Order, Cardinal Pareto, Cardinal independence of Irrelevant Alternatives, Cardinal Neutrality, Invariance w.r.t. Independent Translations and Invariance w.r.t. a Common Multiplication is the weighted sum, i.e. the alternatives are ranked in the decreasing order of their weighted sum [Roberts, 1980]*

where,

- ▷ *Weak Order.* The global preference relation is always a weak order (a ranking, possibly with ties).
- ▷ *Cardinal Pareto.* If alternative a is strictly better than b on all criteria, then a is globally preferred to b .
- ▷ *Cardinal Independence of Irrelevant Alternatives.* The global preference between a and b depends only on their evaluation and not on the evaluation of others alternatives.
- ▷ *Cardinal Neutrality.* The result of the aggregation does not depend on the labels of the alternatives but only on their evaluations.
- ▷ *Invariance w.r.t Independent translations* Suppose that an alternative a is globally better than b . Now, if the performances on each criterion are translated by a different quantity, then a must remain globally better than b in the new relation. This axiom translates the fact that the performances are not really important. What really matters are the differences between the performances.
- ▷ *Invariance w.r.t a Common multiplication* Suppose that an alternative a is globally better than b . Now if the performances on each criterion are multiplied by the same amount, then a must still be globally better than b after the multiplication. This axiom, combined with the previous condition, implies that only ratio of differences between performances are important.

Consequently, this theory seeks to identify systems of axioms characterizing the different aggregation procedures. These axioms can be used to *specify conditions* (on preferences structures) under which it makes sense to apply a given procedure. Thus, during the decision aiding process, these axioms are may be useful to the analyst and decision maker to choose an aggregation function that fit the decision maker's preferences.

The idea is that if the analyst can test indirectly the axiom or explain the content of each axiom to the decision maker and if the latter find theme acceptable, then he should probably use the method that satisfies this set, no other satisfies the same axioms. Thus, the characterisation of

Conjoint Measurement models has the advantage to allow to test whether the model is likely to be able to fit the preference.

In practice, however, it is not always obvious to see how these axioms should be used or tested. One problem is that, there exist different necessary axioms that are very technical (e.g. Archimedean axiom) which require the construction of an important number of questions in order to check if the decision maker accept them or not. Thus, such axioms can hardly be tested: in the most favourable case, they can indirectly be proved, which provides some hints as to whether they hold (that is, whether the preferences of the decision maker exhibit this kind of structure). Typically, it can only be proved that they don't hold. On the other hand, there exist, for some procedures, axioms that are sufficient but not necessary. The consequence is that we have several axioms that are equivalent but difficult to choose among them because they are just necessary.

A further difficulty is the result of the fact that a decision maker may, after giving its consent for an aggregation procedure, challenge the outcomes of this procedure. This challenge may be the result of various reasons. For instance, he may provide a new preferential statement that is not compatible with the axioms chosen at the beginning, even if he had fully accept such axioms. Indeed, it is difficult to know from the beginning of the process all the decision-maker's preferences. This is due mainly to the fact that he does not have a clear vision of his decision problem or he can not express fully and clearly what are his objectives or preferences. Thus, during the interaction new information may contradict some assumptions of the procedure in use. Finally, although some axioms (usually) convey an intuitive notion, it is often difficult to the non-expert in decision aiding methodology to understand clearly what it involves to accept them.

To summarize, in order to synthesize the decision maker's preferences expressed on different points of view, we need an aggregation function. Now how to choose the appropriate function is not an easy task. Conjoint Measurement proposed to specify for each procedure a set of conditions (on the preference structures) that allow to use a procedure rather another. However, in practice, as we have explained, it is not obvious to rely only on such axioms in order to make the choice. Thus, instead of using such axioms, we propose, in this work, to identify a set of "*properties*" allowing us to describe each procedure. As we shall see later in this work, such properties will offer a way to build formal evidences to the selection and the use of an aggregation procedure. We do not seek to replace the Conjoint Measurement framework, but to propose something that seems, from our point of view, more intuitive and easier to use in practice.

1.4.3 Exploiting preferences

The construction of the final recommendation, based on the decision maker's preferences, is an important step in a decision aiding process. The central element for its construction is the "problem statement" (see Section 1.3). Indeed, the final recommendation may takes different

forms, according to the manner in which a problem is stated [Roy, 1996; Bana e Costa, 1996; Roy and Bouyssou, 1993] In this work, we focus on the choice problem, where the aim is to build a *choice set* (see section 1.3) by using a choice procedure.

According to [Bouyssou *et al.*, 2006]:

“a choice procedure C is a function associating a nonempty $C(S)$ of A with each element S of \mathcal{S} , such that S represent the preference relation built on A using an aggregation technique and \mathcal{S} is the set of all conceivable preference relations that can be obtained using such techniques (e.g. \mathcal{S} consists of all complete binary relations of one is using simple majority).”

There exist many techniques for building recommendations for a choice problem (e.g., [Laslier, 1997; Henriet, 1985; Roy, 1968]). We present, in what follows, three techniques.

The first technique, illustrated by the example 1.13, is the *Condorcet rule*, where the choice set contains the alternatives that are preferred to all other alternatives (this choice set can eventually be empty).

Definition 1.6. (*Condorcet rule*)

Compare action a with every other action x . If a defeats every other action in majority comparison, thus a is the **Condorcet winner**.

for all $x \neq a$ a majority of criteria prefer a to x than x to a

Example 1.13. Consider the set of alternatives $A = \{a, b, c, d, e\}$ evaluated on five criteria $\{h_0, h_1, h_2, h_3, h_4\}$. Suppose that, on each criterion, alternatives are weakly ordered by a binary relation \succeq_i , as follows:

$$h_0 : a \succ_0 b \succ_0 c \succ_0 e \succ_0 d$$

$$h_1 : b \succ_1 a \succ_1 d \succ_1 e \succ_1 c$$

$$h_2 : c \succ_2 b \succ_2 a \succ_2 d \succ_2 e$$

$$h_3 : b \succ_3 c \succ_3 a \succ_3 d \succ_3 e$$

$$h_4 : d \succ_4 b \succ_4 c \succ_4 a \succ_4 e$$

Suppose that the above information is aggregated into a binary relation \succeq using simple majority. It is not difficult to see that \succeq corresponds to $b \succeq c \succeq a \succeq d \succeq e$.

Thus, in this case, letting $C(S) = b$ seems to be the only reasonable choice, because there is direct evidence to conclude that b is the best choice, based on the fact that it is possible to show that b is better than all the other actions.

However, it is not always possible to find an alternative that is better than all the other ones.

Thus, to build a recommendation, we have to rely on other procedures. One of the most natural extension of the Condorcet winner is defined in Definition 1.7 for the *Copeland winner*.

Definition 1.7. [Moulin, 1991]

Compare action a with every other action x . Score $+1$ if a majority of criteria prefers a to x , -1 if a majority prefers x to a , and 0 if it is a tie. Summing up those scores over all x , $x \neq a$, yields the Copeland score of a . An action with the highest such score, called a **Copeland winner**, is selected.

The Copeland rule selects the alternative with the largest Copeland score, which is the number of times an alternative beats other alternatives minus the number of times that alternative loses to other alternatives when the alternatives are considered in pairwise comparisons. We illustrate such a rule in the Example 1.14

Example 1.14. Consider the set of alternatives $A = \{a, b, c, d, e\}$ evaluated on five criteria $\{h_0, h_1, h_2, h_3, h_4\}$. The ranking on each criterion is as follows:

$$\begin{aligned} h_0 : a \succ_0 e \succ_0 c \succ_0 d \succ_0 b \\ h_1 : b \succ_1 a \succ_1 e \succ_1 c \succ_1 d \\ h_2 : c \succ_2 d \succ_2 b \succ_2 a \succ_2 e \\ h_3 : e \succ_3 a \succ_3 d \succ_3 c \succ_3 b \\ h_4 : d \succ_4 b \succ_4 a \succ_4 c \succ_4 e \end{aligned}$$

The application of the Copeland rule gives the following results: $score(a)=2$, $score(b)=0$, $score(c)=0$, $score(d)=-2$, and $score(e)=0$. Thus, a has the greatest Copeland score and is therefore the Copeland winner.

In the above example, the choice set $C(S) = \{a\}$ and we can justify that by the fact that a has the greatest score and even it does not beat b , this latter can not be in the choice set because it has a smallest score than a .

1.5 What is missing in the DAP model?

The decision aiding process (DAP) is the result of a dialogue between an analyst and a decision maker. During this process the four artefacts are evolving, changing and undergoing revisions. Moreover, since a decision aiding process always refers to a decision process which has a time and space extension it is natural that the outcomes of the DAP remain *defeasible cognitive artefacts* in the sense that new informations, beliefs, values may invalidate them and require an update or a revision. We draw the attention of the reader to the fact that there is a distinction between *revision* and *update*. For instance, [Katsuno and Mendelzon, 1991] make a clear distinction between such two notions:

“We make a clear distinction between two kinds of modifications to a knowledge base. The first one, *update*, consists of bringing the knowledge base up to date when the world describe it changes. The second type of modification, *revision*, is used when we are obtaining new information about static world.”

Thus, update and revision refer to two different concepts: revision supposes a static world, and an agent revising its view of the world on the basis of new information. On the other hand, an update corresponds to an agent modifying its view of the world as a consequence of an observed evolution of the world. In our case, there is no observed world per se, but from the system point of view the world corresponds to the decision-maker preferences, and what can be observed from the world are the different responses and statements made during the process. Following this, an *update* of the system is a modification triggered by an evolution of the decision-maker’s preferences (this is the case for instance at turn 10 of our motivating example, when the client changes its mind and now favours a classical model); whereas a *revision* occurs when the system modifies its view on the basis of new information provided by the decision-maker (this is the case on the same turn of the dialogue, when the decision-maker specifies that the difference is not significant enough).

Get back to the model of decision aiding process, we present the example 1.15 that offers a glimpse of changes on different models of the decision aiding process model during an interaction.

Example 1.15. [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’s say cost, quality and transportation capacity;
- ▷ $\mathbb{\Pi}$ 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 which the offers are analysed: price and management costs, technical features (for the quality point of view), etc.
- ▷ H are the criteria that client agrees to use in order to represent these preferences;

- ▷ *there is no uncertainty;*
- ▷ *R could be a multi-attribute value function provided 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” since two different types of buses may fit the demand better.*
- ▷ *uncertainty is associated now the different scenarios of bus use;*
- ▷ *R could be multi-attribute utility function provided 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.

Through this example, we observe that it is necessary to update the contents of different models as the decision aiding process involved in time and space. We note that once confronted to a result, the decision maker realised 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 formulation problem and the evaluation model.

It is clear that the four artefacts are equally important for decision aiding and that conducting such process is not a linear process in which the four outcomes are established one after the other. However, in this work, on the one hand we will not consider the whole process but we will concentrate to the evaluation stage and the recommendation that we can provide to the decision maker, and on the other hand, for the sake of simplicity, we will assume that the two first stages (situation and formulation problem) are done by taking care of DM's cognitive profile and different aspects of the decision problem.

Thus, during the evaluation process the analyst builds a recommendation taking into account all the parameters of the evaluation model. These parameters represent the formalization of all

preferences and information provided by the decision maker. This construction is the result of the interaction between the analyst and the decision maker. Thus, the decision maker can intervene at any time to make changes or provide new information. The consequence is that we can have different versions of that model at different stages of the interaction. Such versions are due on the one hand to the fact that the client doesn't know how to express clearly and fully, at the beginning of the process, what is his problem and what are his preferences and objectives, on the other hand, to the update state of the decision process and the new information available. However, such different versions are strongly related to each other since they carry essentially the same information and only a small part of the model has to be revised. The problem that arises is that there is no formal representation of how the evolution occurs between the different versions.

Indeed, in practice, these updates are handled directly by the analyst who relies on the model. The latter provides him a general framework that can be adapted to different contexts of decision-making. Thus, as we have seen in the previous example, for each modification or revision, the analyst presents a new model. But, the decision maker may express doubts and request clarification, because in many cases he would not foresee the various consequences of a seemingly local modification. In this case, which support the analyst will use to explain to his client the consequences of changes made during the interaction? Indeed, the current model of decision aiding offers a rich theoretical support for the conduct of the process and the construction of the recommendation but fails in providing a mechanism or a guide on how to represent and keep track of such evolutions.

Moreover, among the tasks of the analyst during the decision aiding process is to *convince* and to inform the decision maker on his position within the decision problem. However, the current model does not provide any information on what can be an explicit and a formal justification or explanation that the analyst can present to the decision maker during the dialogue. In other terms, what type of information the analyst can use to construct a justification that is both "simple" and "natural" for the decision maker? Indeed, it is certainly important to convince the client but it is also important to choose an explanation that is the most intuitive for the decision maker. An explanation that can easily inform the decision maker on the reasons leading to a given choice.

We believe that it is necessary to have a tool that will take into account these two aspects, *revision and explanation*, during the evaluation process and the construction of the recommendation. On the one side, the tool will provide an update mechanism to take into account the evolution of the evaluation model (and the associated recommendation) during the dialogue. Through this mechanism the decision maker can be informed of the consequences of the changes. On the other side, in order to build or generate arguments or justifications throughout the interaction, an explanation mechanism can be integrated into this tool. More generally, such a tool will provide a formal support to better manage and guide the evaluation process, and especially to convince the

decision maker by explaining the reasoning steps that allow to reach a certain recommendation. To answer these different issues and needs, we propose in this work to rely on *argumentation theory*.

1.6 Conclusion

Our aim through this chapter was to introduce the reader to the different concepts and tools related to the decision aiding process: the interaction between a client (a decision maker) and an analyst, aiming to aid the client within a decision process. Decision aiding cannot be seen as just the construction of a formal decision model. It is a complex activity, which can be described and characterised by its outcomes, summarised through four cognitive artifacts.

What we can observe is that shifting from the traditional approach where decisions are viewed as solutions of elegant mathematical problems to the one where the processes conducting to a possible final recommendation are the basic analytical elements implies the necessity to use formal languages enabling to: (i) capture the feedback loops present in such processes; (ii) account for the inconsistencies which may appear during the process; (iii) account for irreducible uncertainties, possibly of qualitative nature; (iv) consider the necessary revisions and updates that may occur along such processes. Conventional mathematical languages used in decision theory do not necessarily fit such requirements. Thus, it is necessary to look for languages explicitly allowing to take them into account (see for instance [Tsoukiàs, 1991]).

Under such a perspective, we aim to propose in this thesis an approach based on *argumentation theory*. However, our contribution is much more modest, in the sense that we do not seek to address all the requirements cited above, we will focus essentially on *revision and explanation* in the DAP. More precisely,

- ▷ Hypothesis 1.1: we focus on a multi criteria evaluation model and the final recommendation;
- ▷ Hypothesis 1.2: we consider only the choice problem statement.
- ▷ Hypothesis 1.3: we use only very simple numerical representation of the type $h_i(a)$ (called performance). In other terms, the information available at the beginning of the process is represented mainly by a performance Table. Furthermore, we will use very simple preference model of the type aPb representing a preference between two actions and aIb which represents an indifference between the two actions.

2 Argumentation Theory

Argumentation Theory is a growing field of Artificial Intelligence. In short, it is the process of constructing and evaluating arguments in order to justify conclusions. The aim of this chapter is to give an overview of different concepts and notions of an argumentation process. We introduce the general idea of non monotonic reasoning and discuss the different steps of an argumentation process. Indeed, different levels can be associated to such process: the logical level which provides the logical structure of a single arguments; the dialectical level which address the notion of conflict between arguments and the procedural level which introduces how we can use arguments within a dialogue. Our purpose, through such levels, is to address the different concepts required to our work, such as argument schemes and critical questions. Moreover, we propose to discuss different points of view on the problem of generating “natural” arguments. The last part of this chapter introduces the representation and the visualization of arguments by graphs structure.

2.1 Introduction

Under the classical logical reasoning (propositional, predicate,...), we can infer that a conclusion is true despite the additions in the set of proposition which allowed us to reach this conclusion. That is what we call *monotonicity*. In other words, no additional information can cause conclusions to be modified or withdrawn. There are no rules which allow to draw conclusions which may be faulty, but are nonetheless better than indecision. This is obvious if our reasoning concerns a mathematical demonstration (indeed classic formal logic has been developed mainly for such a purpose [Whitehead and Russel, 1910]). It is far less obvious if we are concerned by more general reasoning languages where conclusions are not necessarily definite truths.

For instance, if we look at our daily life reasoning, we can observe that this reasoning is not necessarily monotonic. Indeed, we can change our minds and move from one to another conclusion on the simple fact that new information is available or not. Besides, we are often faced with decision situations where we are far from knowing with certainty all data and information necessary to make this decision. We build our conclusion on the basis of available information at that moment and we reserve the right to change it at any time. Indeed, we do not have the time or mental capacity to collect, evaluate, and process all the potentially relevant information before deciding what to do or think. In such cases monotonicity in reasoning is not very useful. In the sense that it does not offer ways to face this type of reasoning. Another example is where we take into account beliefs. Indeed, a human reasoning is not based solely on facts or action but also on beliefs. In this case, classical logic offers no theory about how to deal with such beliefs. For instance, which beliefs to prefer given that certain things are known in a particular case.

These limitations of classical logic caused a number of Artificial Intelligence researchers to explore the area of *non-monotonic logics*. The emergence of these logics were initially developed by [McCarthy, 1977; McDermott and Doyle, 1980; Reiter, 1980]. Part of the original motivation was to provide a formal framework within which to model phenomena such as defeasible inference and defeasible knowledge representation, i.e., provide a formal way to take into account the possibility to deduce conclusions and to change them in the light of further information. A familiar example in the literature of this kind of reasoning is the one of [Reiter, 1987]:

Example 2.1. (*Tweety's example*)

- ▷ *first premise: Birds fly;*
- ▷ *second premise: Tweety is a bird;*
- ▷ *conclusion: therefore, Tweety flies.*

Now, what is if Tweety is a penguin, a type of bird that does not fly?

If we add this kind of information, then the conclusion “Tweety flies” will be false. Thus, knowing that the second premise is valid, and to have a valid deduction, the first premise must be false. But, this interpretation is problematic because the first premise, in reality, still seems to be true. As Reiter said:

“a more natural reading of this premise is one that allows for possible exceptions and allows for the possibility that Tweety could be an exceptional type of bird with respect to the property of flying, that is, ‘Normally, birds fly’ or ‘typically the birds fly’ or ‘if x is a typical bird, then we can assume by *default* that x flies.’” [Reiter, 1987]

The *default* refers to the fact that we should consider that Tweety flies until we can say or prove that it is atypical.

Much interest has been brought to non-monotonic reasoning from researchers in Artificial Intelligence, in particular, from those interested in model human intelligence in computational terms. The challenge has been to formalize non-monotonic inference, to describe it in terms of a precisely-defined logical system which could then be used to develop computer programs that replicate everyday reasoning. Different non-monotonic reasoning formalism emerged, within AI, such as: default logic, developed by [Reiter, 1980], autoepistemic logic [Moore, 1985].

In this chapter we are interested by one kind of these non-monotonic reasoning formalisms which is *argumentation theory*.

Indeed, argumentation provides an alternative way to mechanise non-monotonic reasoning. Specifically, argument-based frameworks view this problem as a process in which arguments for and against conclusions are constructed and compared. Non-monotonicity arises from the fact that new premises may enable the construction of new arguments to support new conclusion, or stronger counter-arguments against existing conclusions. Thus, argumentation is a reasoning model based on the construction and the evaluation of interacting arguments. Those arguments are intended to support, explain, or attack statements that can be decision, opinions, preferences, etc. Its most important characteristic is that during this process, a conclusion originally justified by some argument can become unjustified. This is the result of the defeasibility of arguments [Loui, 1995].

We propose to organize this chapter around the various levels of an argumentation process, as it was identified by [Prakken and Sartor, 2002]: *the logical level* which provides the logical structure of a single arguments; *the dialectical level* which address the notion of conflict between arguments and *the procedural level* which introduce how we can use arguments within a dialogue. On the other hand, recently, there has been a great interest on how to construct and structure the content of an argument to be the most convincing one. Thus, we propose to address in this chapter some points of view on that. Moreover, we plan to discuss, at the end of this chapter, the graphical representation of arguments.

2.2 Argumentation theory: an overview

The interest with argumentation can be justified by the fact that in everyday life, in a discussion, one needs to add gradually new information and to use a non-monotonic reasoning. This possibility to supply gradually additional information is not possible, for instance in a demonstrative reasoning or proof where all the information must be available at the beginning of the process. Thus, contrary to proofs, argumentation will take place in situations where it is dif-

difficult to have complete and certain information. This point was emphasized by [Perelman and Olbrechts-Tyteca, 1969] who state.

“Logic underwent a brilliant development during the last century when, abandoning the old formulas, it set out to analyze the methods of proof used effectively by mathematicians. [...] One result of this development is to limit its domain, since everything ignored by mathematicians is foreign to it. Logicians owe it to themselves to complete the theory of demonstration obtained in this way by a theory of argumentation.”

According to the authors, argumentation has the potential to provide us with means to complement mathematics by addressing the issues that cannot be solved by mathematics alone. They tried to find a description of techniques of argumentation used by people to obtain the approval of others for their opinions. They called this “new rhetoric”. Toulmin, on the other hand, developed his theory in order to explain how argumentation occurs in the natural process of an everyday argumentation. He called his theory “The uses of argument” [Toulmin, 1958].

Since, several works were interested by using argumentation in different fields, such legal reasoning, multi-agent systems, decision making, etc. [Prakken and Sartor, 2002; Amgoud and Cayrol, 2002; Benferhat *et al.*, 1993; Parsons and McBurney, 2003]. In most of such approaches, arguments are constructed from an inconsistent knowledge base where the knowledge base is a set of propositions represented in some formal logics (classical or non-monotonic). In this case, argumentation can be viewed as a method for deducing justified conclusion from an inconsistent knowledge base. Which conclusion are justified depends on the attack and the defeat relations among the arguments which can be constructed from the knowledge base. Instantiations of Dung’s abstract argumentation framework [Dung, 1995] are typically models of this kind, which consider an abstract set of arguments together with defeat relation between these arguments.

A special feature in such a framework is that the internal structure of arguments is not specified, what is important is to identify the set of acceptable¹ arguments on the basis of the defeat relation. The second element of the framework is the binary relation that represents the way by which arguments are in conflicts.

We note that in such a framework and in many AI oriented approaches the existence of a (conflicting) knowledge base is assumed to start with. However, in the decision making context, it is not always possible to assume the existence of a such base to start the process. This point has been emphasized by [Gordon and Walton, 2009], who state:

“in decision-making processes, we cannot assume the existence of a knowledge base as input into the process. Problems for which all the relevant information and knowledge have been previously represented in formal logic are rare.”

¹see section 2.2.4

Indeed, we are often faced with decision situations where we are far from knowing with certainty all data and information necessary to make this decision. We build our conclusion on the basis of available information at that moment and we reserve the right to change it at any time. Therefore, argumentation can be seen as:

“a kind of process for making justified, practical decisions [. . .] The goal of the process is to clarify and decide the issues, and produce a justification of the decision which can withstand a critical evaluation by a particular audience.”[Gordon and Walton, 2009]

As the context of our work deals with decision, this work is in line with this view of argumentation.

On the other hand, argumentation systems formalise non-monotonic reasoning in terms of the dialectical interaction between arguments and counterarguments. According to [Prakken, 2004]:

“They tell us how arguments can be constructed, when arguments are in conflict, how conflicting arguments can be compared, and which arguments survive the competition between all conflicting arguments.”

Thus, an argumentation process can be described as a succession of different steps. In this document, to present the different notions and concepts of argumentation theory used in this work, we follow the steps proposed by [Prakken and Sartor, 2002]. Indeed, they suggest to distinguish four levels or layers, in an argumentation process, summarised as follows.

- ▷ **Logical layer.** It is concerned with the language in which information can be expressed, and with the rules for constructing arguments in that language. In other terms, it defines what argument are, i.e. how pieces of information can be combined to provide basic support for a claim.
- ▷ **Dialectical layer.** It focuses on conflicting arguments and introduces such notions as counter argument, attack, rebuttal, etc.
- ▷ **Procedural layer.** It regulates how an actual dispute can be conducted, i.e., how parties can introduce or challenge new information and state new argument. In other words, this level defines the possible speech acts, and the discourse rules governing them.
- ▷ **Strategic layer.** It provides rational ways of conducting a dispute within the procedural bounds of the third layer.

According to the authors,

“All four layers are to be integrated into comprehensive view of argumentation: the logical layer defines, by providing a notion of argument, the objects to be evaluated at the dialectical layer; the dialectical layer offers to the procedural and heuristic layers a judgement of whether a new argument might be relevant in the dispute; the procedural layer constrains the ways in which new inputs, supplied by the heuristic layer can be submitted to the dialectical one, the heuristic layer provides the matter which is to be processed in the system.”

We note, however, that the next sub-sections will cover only the first three steps of an argumentation process. As the last step is not addressed in this work, it will not be discussed. Moreover, we propose to include an additional stage devoting to diagramming and visualization of arguments during a dialogue. More precisely, this level allows to build graphical representation under the form of graph structures.

2.2.1 Structures of arguments

The first step of an argumentation process is to build the set of arguments. These arguments have the aim to justify or to explain conclusions that can be preferences, decisions, beliefs, etc. The main challenge in this step is to determine the form or the structure of the arguments. Indeed, there are many ways to address the form of an argument: as *trees of inferences* [Karacapilidis and Papadias, 1998a], as *a sequences of inferences* (deductions) [Verheij, 1996], or as *simple premise-conclusion pairs*. The different forms of arguments depend on the language and on the rules for constructing them [Prakken and Vreeswijk, 2002; Chesñevar *et al.*, 2000; Bench-Capon and Prakken, 2005; Toulmin, 1958; Walton, 1990; Amgoud *et al.*, 2004].

For instance, if we use $\alpha, \beta, \gamma, \dots$ to denote formulae and Δ, Φ, \dots to denote a set of formulae and the symbol \vdash to represent the deduction in classical propositional logic, then, an argument can be defined as follow.

Definition 2.1. *An argument is a pair $\langle \Phi, \alpha \rangle$ such that: (1) $\Phi \not\vdash \perp$; (2) $\Phi \vdash \alpha$ and (3) there is no $\phi \subset \Phi$ such that $\phi \vdash \alpha$. We call α the consequence (or the conclusion) of the argument and Φ the support (or the premise) of the argument.*

The choice between the different options depends on the context and the objective sought through the use of argumentation. The approach proposed in this thesis is based upon the structure of *Argument Schemes* [Walton, 1996].

Argument schemes are forms of arguments that capture stereotypical patterns of humans reasoning, especially defeasible ones [Walton, 2005; Norman *et al.*, 2003]. The first attempt to give an account of scheme, was in the work of Aristotle. Indeed, he has introduced scheme in a common forms of argumentation called *topics* in *Topics* [Aristotle, 1939], *On Sophistical Refutations*

and *Rhetoric* [Aristotle, 1928]. After that, argument schemes have been employed, in *The New Rhetoric* [Perelman and Olbrechts-Tyteca, 1969], as tools for analyzing and evaluating argument used in everyday and legal discourse. More recently there has been considerable interest in schemes in computer science, especially in AI, where they are increasingly being recognized, in fields like multi agent system, for their usefulness to refine the reasoning capabilities of artificial agents [Reed and Norman, 2003; Verheij, 2003]. For special use in Artificial Intelligence systems Pollock’s OSCAR identified some ten schemes [Pollock, 1995]. In addition, other examples of applications of argument schemes are specified in [Reed and Walton, 2001].

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 due to 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. For instance, the Example 2.2 presents a version of an argument schemes of Reiter’s example about Tweety.

Example 2.2.

<i>Premises</i>	Birds fly Tweety is a bird
<i>Conclusion</i>	Tweety flies

Table 2.1 — Tweety’s scheme

Another popular argument scheme is the one of [Toulmin, 1958]. This scheme incorporates, contrary to standard arguments consisting of the traditional premises and conclusion, additional elements to describe the different roles that premises can play in argument. Thus, it allows for more expressive argument to be asserted. Toulmin’s scheme comprises the following elements:

- ▷ **a claim (C)**, which is the conclusion of the argument;
- ▷ **a qualifier (Q)**, which gives the strength of the argument for the claim;
- ▷ **the data (D)**, which is like a traditional premise;
- ▷ **the warrant (W)**, which licences the derivation of the claim from the data;
- ▷ **a rebuttal (R)**, which is a proposition which would refute the claim, if the rebuttal were to be proved true;
- ▷ **the backing (B)**, which represents the authority for the warrant.

The elements of this schema are connected as shown in Figure 2.2. An example based on this scheme is presented in Example 2.3.

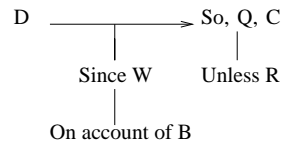


Figure 2.1 — Toulmin’s Scheme

Example 2.3. (Toulmin, 1958)

In support of the claim (C) that Harry is a British subject, we appeal to the datum (D) that he was born in Bermuda, and the warrant can then be stated in the form, A man in Bermuda may be taken to be a British subject: since, however, questions of nationality are always subject to qualifications and conditions, we shall have to insert a qualifying ‘presumably’(Q) in front of the conclusion, and note the possibility that our conclusion may be rebutted in case (R) in turns out that both his parents were aliens or he has since become a naturalized American. Finally, in case the warrant itself is challenged, its backing (B) can be put in: this will record the terms and the dates of enactment of the acts of parliament and other legal provisions governing the nationality of persons born in the British colonies. The result will be an argument set out as is shown in the Figure 2.2.

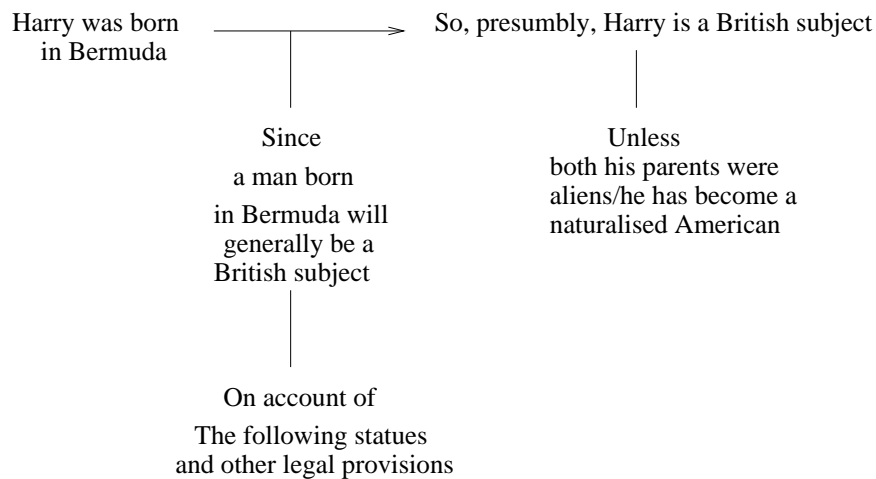


Figure 2.2 — Toulmin’s Scheme

This example can be represented under the form of an argument scheme as it is illustrated in the Table 2.2. According to [Gordon and Walton, 2006], the element “backing” can either be a premise of the scheme or a premise of an argument pro the warrant .

Premises	Harry was born in Bermuda; A man born in Bermuda will generally be a British subject; Legal provisions state that a person born in Bermuda are generally British;
Conclusion	Harry is a British subject

Table 2.2 — Toulmin's scheme

Other schemes were identified by [Walton, 1996]. Indeed, the author presents 25 different argument schemes, such as argument from cause to effect, argument from popular practice, argument from analogy, etc. Other such typologies of argument schemes of varying sizes have also been given by [Hastings, 1963; Kienpointer, 1986; Katzav and Reed, 2004], among others.

Another way to distinguish argument scheme was emphasized by [Bench-Capon and Prakken, 2005]. According to the authors the schemes are not classified according to their logical form but according to their content. Many argument schemes in fact express *epistemological principles* (such as the scheme from the position to know [Walton, 1996]) or principles of *practical reasoning*² (such as the scheme from consequence [Walton, 1996]). The difference between these two types of arguments is discussed in what follows.

2.2.1.1 Argument for beliefs vs for decisions

Argumentation is usually conceived as a process for handling (potentially conflicting) *beliefs*. Thus, we build *epistemic arguments*. In Artificial Intelligence, many systems have been proposed that allow to capture the defeasible nature of this kind of reasoning. Under this perspective, the basic building block (the argument) can typically be defined as a premise/conclusion pair, whereby you state that this conclusion should be reached under these premises. What is discussed here is the truth-value of the conclusion, so an argument supporting a conclusion basically asserts some evidence to believe that this conclusion holds.

When it comes to decision-making though, this rather crude argument scheme needs to be refined. Indeed, as it has been recognised for a long-time now, a significant difference exists between argumentation for beliefs and argumentation for actions (or *practical argument*) [Fox and Parsons, 1997, 1998]. This is best explained by means of a simple example, inspired by [Fox and Parsons, 1997]. Saying that some symptoms “support” a given diagnosis, and that this diagnosis in turn “supports” a given medication are two different things. The first —*epistemic*— argument

²“The need for practical reasoning has emerged from the recent growth of interest in software agent technologies (e.g.[Wooldrdige, 2000]), that puts action at the centre of the stage. Indeed, for software agents to have the capability of interacting with their environment they also need to be equipped with an ability to reason about what actions are the best to execute in given situations.”

is typically a defeasible proof of the doctor’s diagnosis. The latter —*practical*— argument is a recommendation that this course of action should be chosen, which can for instance be defeated by the fact that other medications may turn out to be better options. Thus, when we say “these symptoms support a diagnosis” and “this diagnosis support use of given medication”, the word “support” must be interpreted differently.

To clarify this distinction between epistemic argument and practical argument, we present in the following the Example 2.4, borrowed from [Amgoud, 2009; Amgoud and Prade, 2009]. In these papers, the author summarized her point of view on how to use the arguments to make and explain decisions.

Example 2.4. (Amgoud, 2009)

This example is about having a surgery or not, knowing the patient has colonic polyps. The knowledge base contains the following information: having a surgery has side effects, not having a surgery avoids having side-effects, when having a cancer, having a surgery avoids loss of life, if a patient has cancer and has no surgery, the patient would lose his life, the patient has colonic polyps, having colonic polyps may lead to cancer.

The argument a_1 = [“the patient has a colonic polyps” and “having colonic polyps may lead to cancer”] is considered as an epistemic argument believing that the patient may have cancer. While the argument a_2 = [“the patient may have cancer”, “when having a cancer, having a surgery avoids loss of life”] is a practical argument for having a surgery. This argument is in favour (or supports) the option “having a surgery”.

As the subject dealt with in this thesis concerns the topic of decision and action we shall only refer here to schemes that pertain to practical reasoning (i.e., schemes for actions). Among the early works on schemes for actions, we quote [Walton, 1996], who distinguished two arguments schemes for an action: *the necessary condition scheme* and the *sufficient condition scheme* presented, respectively, in Table 2.3 and Table 2.4.

Premises	G is a goal for agent a Doing action A is necessary for agent a to carry out goal G
Conclusion	Therefore agent a ought to do action A .

Table 2.3 — The necessary condition Scheme

Such schemes reflect the idea that we can support an action if doing this action is sufficient or necessary for the decision maker to reach the desired goal. Obviously, other examples of schemes exist. We will examine some of them in Chapter 3. There we will highlight some issues associated with such schemes as well as discuss them under a multiple criteria decision analysis. Indeed, our

Premises	G is a goal for agent a Doing action A is sufficient for agent a to carry out goal G
Conclusion	Therefore agent a ought to do action A .

Table 2.4 — The sufficient condition Scheme

context deals with an evaluation process based on models with multiple criteria. Thus, among our objectives is to construct argument schemes for actions, that take into account the fact that such actions can be judged or evaluated according to several points of view.

Finally, going back to Toulmin’s scheme, we note that such scheme has proved to be useful due to the expressivity which is offered in the presentation and justification of arguments. However, it lacks elements that allow us to identify conflicts in arguments. For instance, Toulmin’s scheme says little about the manner in which the argument can be attacked. Although the scheme does take into account that the claim could be challenged through the use of the rebuttal, it does not provide a detailed manner in which one can explicitly attack elements of the argument. Indeed, using this schema there is no way to distinguish between different kinds of attack, such as a *rebutter* or an *undercutter* [Pollock, 1995], thus the precise nature of the disagreement may not always be easy to identify.

In the following, we discuss such different types of attacks and conflicts between arguments.

2.2.2 How arguments interact?

Once the arguments constructed, they cannot be considered independently. Indeed, most of the arguments are in interaction: arguments may be conflicting or on the contrary, arguments may support other arguments. [Pollock, 1995] drew an important distinction between two kinds of arguments that can attack and defeat another argument, calling them *rebutting defeaters* and *undercutting defeaters*.

A rebutting attack concerns arguments that have contradictory conclusions. Let \equiv represents the equivalence relation in classical logic and \neg , the negation. Then formally,

Definition 2.2. (*Rebutting Defeaters*)

Given two arguments³ (ϕ_1, α_1) and (ϕ_2, α_2) , (ϕ_1, α_1) *rebut* (ϕ_2, α_2) if and only if $\alpha_1 \equiv \neg\alpha_2$.

If we consider the structure of an argument scheme, a rebuttal can be modeled as an argument in the opposite direction for the same conclusion. For instance, the Example 2.5 can be seen as a rebut for the Tweety’s scheme (see Example 2.2)

³In the sense of the Definition 2.1

Example 2.5. (*Rebutting defeaters*)

Premises	Tweety is a bird Tweety is a penguin
Conclusion	Tweety doesn't fly

Table 2.5 — A Rebut for Tweety's scheme

An undercutting defeater has a different claim. It attacks the inferential link between the conclusion and the premise rather than attacking the conclusion. In other words, we are in presence of undercutting when one argument challenges a rule inference of another argument. Formally,

Definition 2.3. (*Undercutting Defeaters*)

Given two arguments (ϕ_1, α_1) and (ϕ_2, α_2) , (ϕ_1, α_1) undercuts (ϕ_2, α_2) if and only if $\exists \alpha \in \phi_2$ such that $\alpha_1 \equiv \neg\alpha$.

For argument schemes, undercutting defeaters, according to Pollock, are **exceptions** to defeasible reasons. It means that they represent exceptional situations in which a scheme should not be used. For instance, In Toulmin's scheme the sentence "Harry has become an American citizen" constitutes an exceptional situation where Harry should not be regarded as a British subject. Thus, the scheme can be represented as in the Example 2.6. Until the situation where Harry is an American citizen is not true then the conclusion remains valid.

Example 2.6. (*Undercutting defeaters-1*)

Premises	Data	Harry was born in Bermuda.
	Warrant	A man born in Bermuda will generally be a British subject.
	Backing	legal provisions state that a person born in Bermuda are generally British.
	Exception	<i>Harry has become an American citizen</i>
Conclusion	Harry is a British subject	

Table 2.6 — Exception in Toulmin's schemes

In the case of Tweety's example, we considered as a rebut the fact that Tweety is a particular bird (see Example 2.5). However, it is possible to consider or to interpret such information as an exceptional situation where the conclusion becomes false. Thus, we obtain the following argument scheme.

Example 2.7. (*Undercutting defeaters–2*)

Premises	Birds fly
	Tweety is a bird
<i>Exception</i>	<i>Tweety is a penguin</i>
Conclusion	Tweety flies

Table 2.7 — Exception in Tweety’s scheme

We will see, later in this section, that the construction of a rebut or an undercut against an argument scheme, is based on a specific tool, called *Critical questions* [Hastings, 1963].

Moreover, let us remark that both types of defeaters, discussed above, represent negative relations between arguments. However, recent studies have proposed another kind of relation between argument, namely a positive relation, called *support relation* [Verheij, 2002; Amgoud *et al.*, 2004; Karacapilidis and Papadias, 1998a]. Indeed, an argument can defeat another argument, but it can also support another one. This new relation is completely independent of the defeat relation (i.e., the support relation is not defined in terms of the defeat relation, and vice-versa). As it was pointed by [Cayrol and Schiex, 2005]:

“This suggests a notion of *bipolarity*, i.e. the existence of two independent types of information which have a diametrically opposed nature and which represent contrasting forces”

In decision making context such situation occur frequently (see [Oztürk *et al.*, 2005; Tsoukiàs *et al.*, 2002]): there always exist positive reasons supporting a certain decision and negative reasons against it. We can illustrate this idea by the example of the decision process of the Security Council of the United Nations. This is composed by 15 members (10 elected and 5 permanent). The decision rule for adoption a resolution requires that **at least 9** out of the 15 members **agree** and that **no permanent** member uses its **veto**. It is easy to observe that the above decision rule there exist agents having a **negative power**. Such a negative power is not compensated by the positive power of each agent when forming the majority. It acts independently and only in a negative sense. These two powers cannot be combined between them, although they both influence the final decision. Thus, according to the majority, positive arguments are advanced and the veto can be represented by a negative argument against the decision.

Moreover, if we take again the Example 2.4, we can have the following arguments pro and con a decision (see example 2.8).

Example 2.8. (*Example 2.4 Cont.*)

The two practical arguments $a_2 = [“the patient may have cancer”, “when having a cancer, having a surgery avoids loss of life”]$ and $a_3 = [“having a surgery has side-effects”]$ represent respectively, an argument in favour and an argument against the option “having a surgery”. The second one points out negative consequence of the option.

Another way to challenge an argument is to use the concept of *Critical Questions* [Hastings, 1963]. Critical questions are associated to an argument scheme. They represent attacks, challenges or criticisms that, if not answered adequately, falsify the argument fitting the scheme. For instance, unlike in Toulmin’s scheme (see example 2.3), Walton associates to each scheme a mean which allows to identify conflicts between arguments. In fact, according to [Walton, 1996], argument schemes can play two roles:

- ▷ when constructing arguments, they provide a repertory of forms of argument to be considered, and a template prompting for the pieces that are needed;
- ▷ when attacking, arguments provide a set of critical questions that can identify potential weaknesses in the opponents case.

Then, as [Walton and Reed, 2002] put 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.”

Thus, critical questions represent a kind of attacks. Asking such questions throw doubt on the structural link between the premises and the conclusion. They can be applied when a user is confronted with the problem of replying to that argument or evaluating that argument and whether to accept⁴ it.

For instance, to Walton’s schemes for practical reasoning (see Table 2.3 and Table 2.4), the attached critical questions are as follow:

Question 1: Are there alternative ways (other than A) of realizing G ?

Question 2: Is it possible for a to do A ?

Question 3: Does a have goals other than G that should be taken into account?

Question 4: Are there other consequences of bringing about A that should be taken into account?

⁴ The notion of acceptability in argumentation is discussed in the subsection 2.2.4.

To better understand the concept of critical questions, let's take the following example.

Example 2.9. [Walton and Gordon, 2005]

The medical examiner (ME) said that the tissue sample found at the crime scene matches the DNA of the suspect. Therefore, the tissue sample found at the crime scene matches the DNA of the suspect.

This example is an instance of the argument scheme from expert opinion. This scheme was formulated in [Walton, 1997, 2007] as follows:

Premises	Source <i>E</i> is an expert in subject domain <i>D</i> containing proposition <i>A</i> <i>E</i> asserts that proposition <i>A</i> (in domain <i>D</i>) is true (false)
Conclusion	<i>A</i> may plausibly be taken to be true (false)

Table 2.8 — Scheme for Argument from expert Opinion

Now, to evaluate this scheme, we can use one of the six basic critical questions matching the appeal to an expert opinion, as indicated in [Walton, 1997]:

Expertise question: How credible is *E* as an expert source?

Field question: Is *E* an expert in the field that *A* is in?

Opinion question: What did *E* assert that implies *A*?

Trustworthiness question: Is *E* personally reliable as a source?

Consistency question: Is *A* consistent with that other experts assert?

Backup Evidence question: Is *E*'s assertion based on evidence?

Here, the *expertise question* challenges the premise that the witness, *E*, has expert knowledge in some domain. Credibility is the notion that because *E* is an expert in a domain of knowledge, *E* is in a position to know, and therefore what *E* says has more value as evidence than what a non-expert would say. The *field question* challenges the premise that the domain of expertise of *E* is the relevant domain, *S*. The *trustworthiness question* challenges the honesty and objectivity of *E* as a source of knowledge. The *consistency question* challenges the correctness of the expert's testimony with contradictory statements by other acknowledged experts. Finally, the *backup evidence question* challenges the expert to support his claim with evidence.

In the case of the Example 2.9, one natural critical question to begin with would be to ask about the ME's qualifications. *Is she a medical doctor? Is she a specialist?*

Through the previous example we can observe that critical questions may have different roles and can attack various parts of the argument. [Verheij, 2003] distinguished the following roles for a set of critical questions:

1. They can be used to question whether a premise of a scheme holds;
2. They can point to exceptional situations in which a scheme should not be used;
3. They can set condition for the proper use of a scheme;
4. They can point to other argument that might be used to attack the scheme.

Such different roles of the critical questions are based on the distinction established by [Pollock, 1995], between undercutter and rebutter (see, respectively, definition 2.3 and definition 2.2). Thus, critical questions of type (2) undercut an argument, those of type (3) refute specific implicit assumptions on which the argument rests, while those of type (4) point to “rebutting” counter-arguments.

For instance, in the case of Tweety’s scheme, a critical question of type (2) would be: *is Tweety an abnormal bird?*.

Other works distinguished the roles of critical questions on the basis of their impact on what we call the *burden of proof*. Indeed, this point was emphasized by [Gordon *et al.*, 2007; Walton and Gordon, 2005]. The authors state:

“The critical questions evidently need to be formalized along with the scheme, in order to capture the logic of each scheme as a defeasible argument. The biggest problem is how to carry out this task, since the relationship of a set of critical questions to scheme brings in the notion of a dialogue sequence between a questioner and answerer. The method of evaluating an argument is by a shifting of *burden of proof* in a dialogue.”

For instance, any argument fitting the scheme for argument from expert opinion is supposed to be evaluated in a *dialogue framework* in which another party can ask one of the six critical questions. In this case a burden of proof shifts back to the proponent’s side to respond appropriately. But not all the questions have the same effect. Indeed, for some asking the question is enough to shift the burden of proof back to the party who put forward the argument to answer the question. For other critical questions, the party who raised the question also has the burden of answering it. How to distinguish between these two kinds of critical questions and how to evaluate arguments is discussed further in subsection 2.2.4.

2.2.3 Arguments in dialogue

Arguments are embedded in a procedural context, in that they can be seen as having been put forward on one side or the other of an issue during a dialogue between human and/or artificial agents. In other terms, one way to define argumentation logics is in the dialectical form of *dialogue games* (or dialogue systems). Such games model interaction between two or more players, where arguments in favour and against a proposition are exchanged according to certain rules and conditions [Carlson, 1983]. According to [Gordon *et al.*, 2007]

“The information provided by a dialogue for constructing and evaluating argument is richer than just a set of sentences. Indeed, the context can tell us whether some party has questioned or conceded a statement, or whether a decision has been taken to accept or reject a claim.”

2.2.3.1 Dialogue system

Dialogue systems essentially define the principle of coherent dialogue and the condition under which a statement made by an individual is appropriate. Different formal dialogues exist, taking into account various information, such as: participants, communication language, roles of participants, the dialogue goal, etc.

In what follows we present only the necessary elements for the presentation of our proposal in this thesis. For more details on dialogue systems, we refer the reader to [Prakken, 2005a; McBurney and Parsons, 2003; Amgoud *et al.*, 2000a; McBurney and Parsons, 2009].

Locutions rules (speech acts, moves) Rules which indicate what utterances are permitted. Typically, legal locutions permit participants to assert propositions, permit others to question or contest prior assertions, and permit those asserting propositions which are subsequently questioned or contested to justify their assertions. Justifications may involve the presentation of a proof of the proposition or an argument for it. The Table 2.9 presents examples of locution rules identified by [Prakken, 2005a], such that ϕ , A and B represent statements.

Commitments rules. Rules defining the effect of the moves in the “commitment stores”. Indeed, associated with each player is a commitment store, which holds the statements players have made and the challenges they have issued. There are then rules which define how the commitment stores are updated. For example, a question posed by one agent to another may impose a commitment on the second to provide a response; until provided, this commitment remains undischarged.

Acts	Attacks	Surrenders
claim ϕ	why ϕ	concede ϕ
why ϕ	argue A ($\text{conclusion}(A) = \phi$)	retract ϕ
argue A	why ϕ ($\phi \in \text{premise}(A)$) argue B (B attacks A)	concede ϕ ($\phi \in \text{premise}(A)$ or $\phi = \text{conclusion}(A)$)
concede ϕ		
retract ϕ		

Table 2.9 — Examples of speech acts

Dialogue rules (protocol). Rules for regulating the moves. It specifies for instance the set of speech acts allowed in a dialogue and their allowed types of replies. Various dialogue protocols can be found in the literature, especially for persuasion [Prakken, 2001] and negotiation [Parsons *et al.*, 1998; Amgoud *et al.*, 2000b]

Termination rules. Rules that define the circumstances under which the dialogue ends.

Now, it is clear that different “types of dialogue” involve different set of rules. For example, the set of rules given in Table 2.9 correspond to a *persuasion dialogue*. There are other types of interaction as identified for instance by [Walton and Krabbe, 1995]. Indeed, the authors have identified a number of distinct dialogue types used in human communication: Persuasion, Negotiation, Inquiry, Information-Seeking, Deliberation, and Eristic Dialogues. This typology has proved to be influential in the study of argumentation theory and its application to agent systems. All these types are characterised by their initial situation, main goal and the aims (or the benefits) of the participants. The type of dialogue that has been most studied in argumentation theory is the persuasion dialogue [Prakken, 2005a, 2006]. The different types are summarised in Table 2.10 and briefly described in Annexe C.

2.2.3.2 Argument schemes in dialogue

Our aim in this thesis, among others, is to represent formally the interaction between an analyst and a decision-maker during the evaluation phase of a decision aiding process. For such a purpose, we need different classes of argument schemes to construct the whole evaluation process (see Chapter 4). Thus, to represent such interaction, we need to investigate how we can use argument schemes in a structure of a dialogue game.

Argument schemes have a clear dialectical aspect in that they come with a set of critical questions, which enumerate ways of challenging argument created using scheme. To our knowledge, however, there are few studies that have addressed the construction of dialogue game (protocol) on the basis of argument schemes and critical questions. In this section, we quote two examples.

First, [Atkinson *et al.*, 2005] use an argumentation scheme for proposing dialogue game protocol for arguing about action. The protocol, called the Persuasive Argument for Multiple Agents (PARMA) Protocol, embodies an earlier theory by the authors of persuasion over action [Atkinson *et al.*, 2004] which enables participants to rationally propose, attack, and defend, an action or course of actions (or inaction). Another example is the one of [Reed and Walton, 2007] who defined an Argumentation Scheme Dialogue (ASD). The authors used as basis [Walton, 1984] game in order to explore how a dialogue game can be extended to encompass argumentation schemes. The author extend the original set of locutions to include supporting claims with arguments constructed from schemes. Moreover, they set new rules that allow to take into account critical questions, such that it is possible to distinguish between the different roles of questions: exception and assumption (see subsection 2.2.4).

Type of dialogue	Initial Situation	Participant' Goal	Goal of Dialogue
Persuasion	Conflict of opinions	Persuade other party	Resolve or clarify issue
Inquiry	ignorance	find and verify or falsify evidence	proof or disproof
Negotiation	Conflict of interest	get what you most want	reasonable settlement that both can live with
Information seeking	unequal spread of information	Acquire or give information	spreading information
Deliberation	dilemma or practical choice	influence and contribute to outcome	Decide best course of action
Eristic	Personal conflict	Verbally hit out at opponent	Reveal deeper basis of conflict and reach some accommodation

Table 2.10 — Types of dialogue [Norman *et al.*, 2003]

2.2.4 Computing acceptability

In an argumentation process, it is important to define the status of arguments (or to evaluate them) on the basis of all the ways in which they interact. Thus, the best or *acceptable* arguments must be identified at the end of the argumentation process.

Most of argumentation systems are based on the notion of acceptability as it was identified by [Dung, 1995]. Dung has proposed an abstract framework for argumentation in which he focuses only on the definition of the status of arguments. In such framework, the acceptability of an

argument depends on its membership of some sets, called *acceptable sets or extensions*. In other terms, the acceptability of arguments is defined without considering the internal structure of the arguments.

However, in our context, the structure of an argument will play an important role. In fact, very briefly, we aim to construct arguments during the evaluation process of a decision aiding process. Such arguments have the goal to justify conclusions of different types. Indeed, an evaluation process represents different stages with different types of information and assumptions at each stage (aggregation, recommendation, etc.). Such information depends on the decision problem and the preferential information provided by the decision maker. Therefore, we need for each level a specific structure of an argument (see Chapter 4). Moreover, in a decision aiding process, the arguments are certainly important but the interest of the decision-maker, at the end of the process, is to know what is the best course of action to take on the basis of such arguments. In other words, what is the impact of the arguments on the final recommendation. This point was emphasized by [Gordon and Walton, 2009], who state:

“ . . . since in the end the people are not interested in which arguments are the best, but rather which consequences they can draw from these arguments to make decision”

Thus, instead of using Dung’s acceptability, we address acceptability as it is defined by [Gordon *et al.*, 2007; Gordon and Walton, 2006] in the *Carneades model*. The Carneades is a tool for constructing and visualizing arguments in graphs⁵ of nodes and links. Moreover, it is a model that applies *proof standards* to determine the acceptability of statements. Thus, Unlike in Dung’s framework, the argument evaluation structure, in Carneades, is used to determine the *acceptability of propositions* in an *argument graph*. Intuitively, a statement is acceptable if a decision to accept the statement as true can be justified or explained given the arguments which have been put forward in the dialogue.

The evaluation of the set of arguments in such graph depends, first, on the *current dialectical status* of each statement in the dialogue, i.e. whether it is *stated, questioned, accepted, or rejected*. This status information is pragmatic; the status of statements is set by speech acts in the dialogue, such as asking a question, putting forward an argument or making a decision. Secondly, for each statement is assigned a *proof standard*. The proof standard allows to aggregate the set of arguments pro and con the statement, in order to resolve the conflict. Several proof standards were identified in the Carneades model, the principle ones are:

1. *Scintilla of Evidence*. It is satisfied if the issue is supported by at least one pro argument.
2. *Preponderance of Evidence*. It is satisfied if the strongest defensible pro argument outweighs the strongest defensible con argument, if there is one.

⁵see subsection 2.2.6

3. *Clear and convincing evidence*. Not only the pro arguments outweigh the con arguments but the difference in weight must exceed some threshold.
4. *Beyond reasonable doubt*. Not only must the weight difference exceeds some threshold but, as the name of the standard suggest, the weight of the strongest con argument must be below the threshold of “reasonable doubt”.

Usually, in argumentation the notion of proof standard that should be used is determined by a given context. This is especially true in the legal domain, where the proof standards that should be used are well identified. However, this is not restricted to legal field and can be used in a context of decision-making. For instance, *Hermes*, a group decision support system where argumentation is used to enhance group decision making, used proof standards to calculate an *activation label* associated to the component of the system [Karacapilidis and Papadias, 2001]. Thus, the activation label is used to indicate the status of each component. Such status allows to accept (or reject) a position and to distinguish the recommended alternatives from the rejected one. The problem in *Hermes* is that the status can be either *active* or *not active*, which is not fully satisfied in a dialogue game, where a statement may take different status, as it was mentioned in the Carneades model.

Finally, they assume a *strict partial ordering* on arguments, denoted with $>$. Let $a1$ and $a2$ be arguments. If $a1 > a2$ therefore $a1$ has priority over $a2$. To summarise, to evaluate a set of arguments in the Carneades model, it is required: the status of statements, the proof standard and a partial order among arguments. Such requirements are formalized by “an argument context” as follows.

Definition 2.4. (*Argument Context*) [Gordon et al., 2007]

Let C , the argument context, be a tuple $\langle \text{status}, ps, > \rangle$, where *status* is a function of type $\text{statement} \rightarrow \{\text{stated}, \text{questioned}, \text{accepted}, \text{rejected}\}$, *ps* is a function of type $\text{statement} \rightarrow \text{proof-standard}$ and $>$ is a strict partial ordering on arguments.

More precisely, the acceptability of a statement s , in an argument graph, depends on the satisfaction of a certain proof standard ps . Formally,

Definition 2.5. (*Acceptability of statements*) [Gordon et al., 2007]

Let *acceptable* be a function of type $\text{statement} \times \text{argument-graph} \rightarrow \text{boolean}$. A statement s is acceptable in an argument graph G if and only if it satisfies its proof standard:

$$\text{acceptable}(s,G) = \text{satisfies}(s,ps(s),G) \tag{2.1}$$

Moreover, whether or not a statement’s proof standard is satisfied depends on the *defensibility* of arguments both in favour and against the statement s . For instance,

- ▷ with the *Scintilla of Evidence* proof standard, a statement meets this standard if and only if it is supported by at least one *defensible* pro argument. In other terms, even the smallest piece of evidence is enough to make the claim justifiable.
- ▷ a statement meets the *Dialectical Validity* proof standard if and only if it is supported by at least one *defensible* pro argument and none of its con arguments are *defensible*.
- ▷ preponderance of evidence: there is at least one *defensible* pro argument and the average weight assigned to the argument pro is greater than the average weight of the argument con, where the average weight is defined to be the arithmetic mean.

On the other hand, the determination of the defensibility of arguments depends on the dialectical status of the statement (stated, questioned, etc.) and on the types of the premises of the argument. Indeed, in the Carneades model, different kinds of premises are distinguished:

1. **Ordinary premises**: those that always must be supported with further ground;
2. **Assumptions** : those that can be assumed until they have been questioned; and
3. **Exceptions**: those that don't hold in the absence of evidence to the contrary.

This distinction is due to the problem of how to distribute the *burden of proof* [Gordon *et al.*, 2007]. With such a distinction, evidence for ordinary premises and assumptions must be produced by the proponent of the argument with such premises, whereas evidence for exception must be produced by the respondent. In addition, this distribution allows to represent *critical questions* in Carneades as assumptions and exceptions. Indeed, as pointed out by [Walton, 2007]:

“The original motivation of the Carneades system was to accommodate two different variations of what happens when a respondent asks a critical question. On one theory, when a critical question is asked, the burden of proof shifts to the proponent's side to answer it. On another theory, merely asking the question does not defeat proponent's argument until the respondent offers some evidence to back it up.”

Thus, if the respondent, the person who poses the question, should have the burden of proof, then the critical question should be modeled as an exception. If, on the other hand, the proponent, the party who used the scheme to construct the argument, should have the burden of proof, then the question should be modeled as an assumption. Thus, knowing the type of premises and the dialectical status of a statement, an argument is therefore defensible in an argument graph G if and only if its *premises holds* in that graph. Different cases can be distinguished:

if p is an ordinary premise, then:

$$\text{holds}(p, G) = \begin{cases} \text{acceptable}(s, G) & \text{if } \text{status}(s) = \text{stated} \\ \text{acceptable}(s, G) & \text{if } \text{status}(s) = \text{questioned} \\ \text{true} & \text{if } \text{status}(s) = \text{accepted} \\ \text{false} & \text{if } \text{status}(s) = \text{rejected} \end{cases}$$

if p is an assumption, then:

$$\text{holds}(p, G) = \begin{cases} \text{true} & \text{if } \text{status}(s) = \text{stated} \\ \text{acceptable}(s, G) & \text{if } \text{status}(s) = \text{questioned} \\ \text{true} & \text{if } \text{status}(s) = \text{accepted} \\ \text{false} & \text{if } \text{status}(s) = \text{rejected} \end{cases}$$

if p is an exception, then:

$$\text{holds}(p, G) = \begin{cases} \neg \text{acceptable}(s, G) & \text{if } \text{status}(s) = \text{stated} \\ \neg \text{acceptable}(s, G) & \text{if } \text{status}(s) = \text{questioned} \\ \text{false} & \text{if } \text{status}(s) = \text{accepted} \\ \text{true} & \text{if } \text{status}(s) = \text{rejected} \end{cases}$$

To sum up, the definition of the acceptability of statements in the Carneades model is recursive. First, it depends on its proof standard. Whether or not a statement's proof standard is satisfied depends on the defensibility of the arguments pro and con this statement. The defensibility of an argument depends on whether or not its premises hold. Finally, whether or not a premise holds can depend on whether or not the premise's statement is acceptable.

2.2.5 Constructing “natural” arguments

In this work, we are interested by the construction of explanations in order to convince the decision maker or to justify the choices and the recommendations presented during the evaluation process. Such explanations are important because they allow, among others, to inform the decision maker on how such choices have been decided. Several questions arise: what is a meaningful explanation in a decision aiding context? How to provide explanations that are “simple” and “natural” for the decision maker?

Recently, several works have focused on developing models for generating automatically explanations and presentation of arguments (e.g. [Carberry *et al.*, 1999; Morik, 1989]). Different issues were addressed such as how to structure the content of an explanation to be the most “natural” for a decision maker. In fact, an explanation is not interesting for a decision maker if it is too general or not enough meaningful. Thus, we must pay a special attention to the contents of this explanation. In this section, we propose to give an overview on this subject by present-

ing three different approaches. Each approach takes into account the user's preferences by using the multi attribute value model, the branch of decision theory that addresses tradeoffs among multiple competing objectives.

The first work that we quote in this section, that is based on such a model, is the one of [Klein, 1994], where strategies for automatically explaining decisions are presented. A part of motivations of this work is that:

“Computer-based tools for decision analysis produce meaningful results for any set of inputs that satisfies a well defined set of constraints, but they usually lack intuitive facilities for justifying choices and for modifying choice parameters.”

To meet such needs, [Klein, 1994] proposed IVA (Interpretive Value Analysis), a framework for explaining and refining choices in the context of intelligent systems. According to the authors:

“IVA increase the transparency of multi attribute value theory, a formal model of value, by reformulating the theory and embedding it in a framework for explaining and iteratively refining value based choices ”

Regarding the explanation point, the author tries to provide answers to questions such as: what is a convincing justification for a choice? How to provide such explanation? what information beyond value function is required? etc. the author claimed that an explanation for choices should do more than displaying of the model and of the parameters that compute those choices, as it is showed in the Example 2.10.

Example 2.10. [Klein, 1994]

Suppose that a user has generated a large dataset that is to be printed on a device that presently disabled, and that a choice among the following alternative plans for manipulating that dataset must be made to free space on the operating system queue:

- ▷ COPY: *copy the dataset to tape, and print it later when the requested printer has been repaired;*
- ▷ EXPENSIVE-PRINTING: *Print the dataset on a faster printer that uses forms more expensive than those requested by the user;*
- ▷ CHEAP-PRINTING: *Print the dataset on a slower printer that uses forms of lower quality than those requested by the user;*
- ▷ DELETE: *Delete the dataset form the queue;*
- ▷ INSTALL: *Connect a duplicate of the requested printer for use until the requested printer for use until the requested printer has been repaired;*

- ▷ DASD: *Transfer the dataset to the user’s private disk storage, so that the user can transfer it back to the queuer for printing after the printer has been repaired.*

Such a problem is modeled by a multi attribute value model, such that the following attribute were considered: x_1 : additional operator time in minutes; x_2 : additional turnaround time in minutes; x_3 : form similarity (subjective index); x_4 : additional cost; and x_5 : problem resolution time in minutes. After that, each alternative is presented as a vector of attribute values, as it is shown in the following table:

Alternative	x_1	x_2	x_3	x_4	x_5
COPY	10.0	34.2	1.0	1.00	15.1
EXPENSIVE-PRINTING	0.1	0.0	1.0	100.00	25.0
CHEAP-PRINTING	0.1	10.0	0.8	0.00	0.1
DELETE	0.1	inf.	0.0	0.00	0.1
INSTALL	180.0	180.0	1.0	5000.00	210.0
DASD	0.1	32.1	1.0	0.50	1.0

Next, the following value function is used for assessing queue space management actions: $v(x_1, x_2, x_3, x_4, x_5) = .1v_1(x_1) + 0.25v_2(x_2) + 0.25v_3(x_3) + 0.2v_4(x_4) + 0.2v_5(x_5)$. Finally the different actions were evaluated and compared, based on the previous model. The best choice is: DASD and the following explanation was given:

Explanation: *Your best option is DASD because your value function $v(x_1, x_2, x_3, x_4, x_5) = 0.1v_1(0.1) + 0.25v_2(32.1) + 0.25v_3(1) + 0.2v_4(0.5) + 0.2v_5(1) = 0.85$ and DASD maximizes v over all available alternatives.*

Thus, [Klein, 1994] emphasized that the explanation presented in the Example 2.10 is problematic, because it does not *appeal to intuition*, for instance, or it *does not compare alternatives explicitly*. Therefore, according to the author the explanation should be *intuitive, comprehensive and persuasive*. To account for that, he proposed a set of strategies, for explaining value based choices, that organize the elements, including in the explanation, to provide an overview on how the choices are calculated. Among such strategies: *the Value-tree pruning and presentation* which involves summarizing a choice between two particular alternatives, guided by the structure of value tree⁶. The idea is that the problem of summarizing a comparison of two alternatives can be decomposed into distinct subproblems:

“First, the explanation facility must determine a desirable level of abstraction for talking about objectives by *vertically pruning* the value tree to eliminate related detailed objectives that can be summarized by a higher level objective. Second, the explanation

⁶The idea is to capture the hierarchical nature of objectives in a corresponding hierarchical structure, where the satisfaction of a given objective is measured in terms of the satisfaction of its component objectives.

facility must determine the most pertinent objectives to display at a given level of abstraction in value tree by *horizontally pruning* the value tree to eliminate some set of objectives.”

Other strategies were presented by the author that we shall not discuss here and we refer the reader to [Klein, 1994]. An example of explanation after applying one of his proposed strategies is the one presented in the Example 2.11.

Example 2.11. [Klein, 1994]

DASD is at least as good as COPY regarding all objectives that underlie the choice with respect to overall queue space management effectiveness. Additional operator time and problem resolution time provide the most compelling reasons.

However, according to [Carenini and Moore, 2006, 2000], such an approach covers only a part of the generation process which is the *selection of content*. In fact, the authors point out that most approaches that are interested on automatic generation of explanations suffer by two limitations:

“First, because of the complexity of generating natural language, researchers have tended to focus only on specific aspects of the generation process. Second, because of a lack of systematic evaluation, it is difficult to gauge the effectiveness, scalability and robustness of the proposed approaches.”

Thus, they adopt and extend the work of [Klein, 1994] on explaining decision theoretic device based on an Additive Multi Attribute Value Function. They proposed to tackle such limitations by developing a computational model for generating arguments, by taking into account principles from argumentation theory and computational linguistics. The idea, is that they propose to generate arguments that are properly arranged and concise, by following guidelines from argumentation theory and by employing a quantitative model of the user’s preferences, which is the multi-attribute value model. The guidelines allow to specify what content should be included in the argument and how it should be arranged. To account for that several decisions should be taken: what represents supporting evidence? how to position the main claim? how to order supporting and opposing evidence? How to select supporting (and opposing) evidence? etc.

Moreover, this approach relies on guidelines from computational linguistic for expressing the content of the argument in natural language. An example of an explanation according to this approach is the one illustrated in the Example 2.12

Example 2.12. [Carenini and Moore, 2006]

House 3-26 is an interesting house. In fact, it has a convenient location in the safe Eastend neighborhood. Even though house 3-26 is somewhat far from a rapid transportation stop, it is

close to work. And also the traffic is moderate on third street. Furthermore, the quality of house is good. House 3-26 offers a beautiful view of the river. And also it looks beautiful.

The last work that we present in this section is the one of [Labreuche, 2005, 2006], that have also focused on the construction of explanations in the case of the multi-attribute model (We note that the approach have been applied to another MCDA models: the weighted majority rule and the weighted maximum/minimum utility model). Through his approach, the author have proposed to address, two limitations encountered by the previous two approaches. First, in the process of selection of arguments to include in the explanation, neither of the two approaches offer a formal justification for this selection. Secondly, both approaches are focused on multi-attribute model, but they don't take into account, in the explanation, the weights of criteria, which can play an important role to distinguish between the two options.

Thus, the author proposed an approach for the argumentation of the comparison of two options by a multi criteria decision aiding model where the criteria are weighted. The approach is based on the definition of a reasoning approach for selecting the arguments that will be used to justify this comparison. However, in the context of models with weighted criteria, only one type of reasoning can not cover all situations corresponding to the possible value of the two options x , y and the weights v . In fact, due to the complexity of explaining a quantitative preference model, several argumentation reasonings are necessary to cover all cases – ranging from situations where the decision is trivial to situations where it is much more tight. The reason why some arguments can be discarded differs in each situation. The different argumentation reasonings are called *anchors*, by analogy to the concept of anchor defined by [Grize, 1976] to refer to some implicit information used to convince an audience.

[Labreuche, 2005] distinguished, then, four cases depending on the number of positive and negative arguments constructed for the conclusion $y \succ_v x$: (i) there is no negative arguments and therefore the option y is preferred to x on all criteria, (ii) the positive arguments are much more than the negative ones, (iii) there are little more positive arguments than negative ones and (iv) the decision is not trivial at all since there are basically more negative arguments than positive ones. An example of explanation constructed in the second case is presented in the Example 2.13.

Example 2.13. [Labreuche, 2005]

Even though x is better than y on average, y is preferred to x since y is better than x on the criteria that are important whereas y is worse than x on the criteria that are not important.

In this work we seek to address, in general, the same issues, but following a more simple and less ambitious approach. Our main objectif is to propose a simple way for the construction of an explanation in the context of multi criteria evaluation process.

We observe that, in general, the notion of an anchor is similar to the concept of argumentation schemes, in the sense that they represent different types of argumentation reasoning. However, there are differences between anchors and the argument schemes presented in this work. The anchors refer to knowledge commonly accepted, or rules of common sense [Grize, 1976]. The argument schemes are closer to proof or evidence in the sense that all the sufficient information is explicitly given to support a given issue (but unlike the proof the premises can be challenged). Therefore, we propose in Chapter 5 to provide explanations for a more general case than comparing two options. It is clear that it is possible to construct an explanation by making explicit all pairwise comparisons of the winner with the other actions. However, it is possible to construct explanations that are more concise and *natural*.

2.2.6 Diagramming of arguments

The last phase of an argumentation process that we aim to discuss in this chapter is that of diagramming and visualization of arguments. By diagramming we mean the representation of arguments under the form of a graph of nodes and links.

Recently, there has been a growing interest in software support tools for argument diagramming that enable their users to construct and visualize arguments [van den Braak *et al.*, 2006; Rienks *et al.*, 2005; van den Braak *et al.*, 2007]. Such tools are designed to make this laborious task easier and are claimed to be useful to their users as they are guiding them when constructing the representation of an argument. In other terms, such tools try:

- ▷ to facilitate understanding of an argument's structure;
- ▷ to help to see "points of attack" in criticizing an argument.

Our interest in the diagramming of arguments is that we plan to map the interaction in the evaluation process within a decision aiding process to a discussion graph with a tree-like structure. Such graph will allow us to represent the exchanged arguments during the process and also to evaluate and discuss their acceptability. In what follows, we will try to give an overview of some systems or tools for diagramming of arguments. It is of course impossible to propose an exhaustive bibliography of what has been done in this field, but it is not our purpose.

A first example of such a tool is the one of the HERMES SYSTEM [Karacapilidis and Papadias, 1998b]. Indeed, the HERMES system organized the knowledge under the form of a discussion graph, which consists of: *issues* (decision to be made, or goal to be achieved), there is always an issue at the top root of the discussion tree; *alternatives* (different choices attached to an issue), *position* (proposition or claims that defend the selection of an alternative) and *constraints* (a qualitative way to weigh reasons for and against the selection of an alternative). An argument in

such system is a tuple of either the form (position, link, position) or (position, link, alternative). In other words, it links together a position with an alternative or another position belonging to a different issue. However, as we are interested in argument schemes and their critical questions, HERMES does not offer a mean to represent the notion of critical questions as it was described in this chapter.

A second example of a system designed to provide support for the analysis and diagramming of arguments is *Araucaria* [Reed and Rowe, 2004]. It has been used in a number of illustrative examples such as [Reed and Walton, 2005] to show how arguments can be formally represented. *Araucaria* however, is a tool for visualising and manually constructing arguments and it offers no automated support for reasoning about arguments. This point was raised by [Walton, 2007], who states:

“*Araucaria*’s diagramming method is an excellent way for representing arguments made up premises and conclusion, but it seems to reach its limits when it comes to representing critical questions [. . .] Modelling questions in argumentation, as opposed to proposition and inferences on sets of propositions, is something this kind of diagramming is not meant to do, lest it become too complex and thereby less useful.”

Another system discussed by Walton is the *Compendium system*. Different works in [Kirschner *et al.*, 2003] showed how *Compendium* has been used to model argumentation in different context:

“It is a system of computer-supported collaborative argumentation that does utilize questioning as part of the method evaluating an argument. ”

However, the problem with this system is that there is no possibility to visualize the premises of arguments. Indeed, there is not a separate node for each premise, but only a single node of the whole argument. for this reason, it is hard to see how one can visualize an argument scheme where several premises work together to support or rebut some conclusion. A system that seems to answer such problems is the *Carneades model* [Gordon and Walton, 2006].

Carneades provides tools supporting a variety of argumentation tasks, including: argument mapping and visualization, argument evaluation, applying proof standards and respecting the distribution of the burden of proof, etc. The arguments are presented under the form of graph of nodes and links. As noted above, we intend to represent the arguments built during the interaction analyst-decision maker during the evaluation process as an arguments graph. To do that, we aim to adapt the graph proposed by *Carneades* to the context of decision aiding process. Our choice has focused on this system because it allows the explicit representation and evaluation of argument schemes, which will allow us to represent the reasoning steps of an evaluation process.

Formally, an argument graph in the *Carneades* is defined as follows:

Definition 2.6. (*Argument Graphs*)

An argument graph is a labeled finite, directed, acyclic, bipartite graph, consisting of arguments nodes and statements nodes. The edges link the arguments nodes to the statements in the premises and conclusion of each argument.

Figure 2.3 presents a first simple example of the argument diagramming method in the Carneades. It is a reconstruction of the Tweety example. **Arguments** are displayed as circles. Information about the argument is shown inside the circle. For instance, the user can label an argument with the name of the argumentation scheme applied. In this example, argument Tweety is labeled as being an instance of the Tweety’s argument scheme of the Table 2.7.

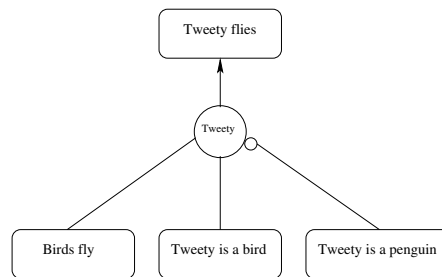


Figure 2.3 — Construction of Tweety’s scheme in Carneades

An argument links a set of **statements**, called premises, to another statement, called the conclusion. In diagrams, statements are displayed in rectangular boxes. In the Figure 2.3, “Tweety flies” is the conclusion of the argument. As for the link from an argument to its conclusion: **pro** and **con** arguments are distinguished using different kinds of arrowheads. An ordinary arrowhead is used for pro arguments, as in the Figure 2.3; open-dot arrowhead is used to indicate con arguments

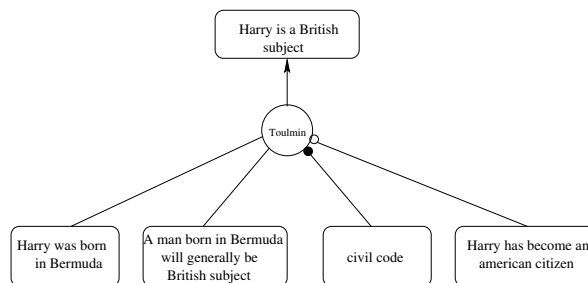


Figure 2.4 — Reconstruction of Toulmin Diagrams in Carneades

Figure 2.4 presents a second example, a reconstruction of Toulmin’s leading example [Toulmin,

1958], about whether or not Harry is a British citizen. This example also illustrates how exceptions and assumptions are diagrammed, using open and closed circular arrowheads on the links of premises. Indeed, to distinguish the different types of premises, Ordinary premises are represented as edges with no arrowheads, assumptions with closed-dot arrowheads and exception with open-dots arrowheads.

The construction of argument graph of Figure 2.4, with the Carneades software is presented in the Annex B. Moreover, we give some details on the calculation of acceptability on such a graph.

2.3 Conclusion

This chapter has been dedicated to the presentation of the different levels of an argumentation process: constructing arguments, defining relation between arguments, managing, evaluating arguments and finally representing argument under the form of graphs. Argumentation provides a powerful tool for the representation of non-monotonic reasoning and the construction of explanations and justifications. Many applications have used argumentation and in particular within a decision making process.

In fact, argumentation has been put forward as a very general approach allowing to support different kinds of decision-making [Pollock, 1987; Prakken and Vreeswijk, 2002]. Adopting an argumentation-based approach in a decision problem would have some obvious benefits. On the one hand, the user will be provided with a “good” choice and with the reasons underlying this recommendation, in a format that is easy to grasp. On the other hand, argumentation based decision making is more akin with the way humans deliberate and finally make a choice. Aspects that are, sometimes, poorly controlled in decision theory.

Thus, we propose in the next chapter is to examine the existent and potential contribution of argumentation theory to decision-aiding, more specifically to multi-criteria decision-aiding.

3

Arguing for decision: a *critical review*

There has recently been many proposals to adopt an argumentative approach to decision-making. As the underlying assumptions made in these different approaches are not always clearly stated, we review these works, taking a more classical decision theory perspective, more precisely a multi criteria perspective. It appears that these approaches seem to have much to offer to decision models, because they allow a great expressivity in the specification of agents' preferences, because they naturally cater for partial specification of preferences, and because they make explicit many aspects that are usually somewhat hidden in decision models.

3.1 Introduction

The ultimate aim of a multi-criteria decision analysis study is to build a possible recommendations that will be considered useful by the users in the decision process where they are involved. Such recommendations are based on formal preference models (see Chapter 1). Different steps (which can be implicit in a decision process) are required in order to obtain a recommendation: formulate and structure the problem, build an evaluation model which allows to obtain a formal answer to a given problem and construct a recommendation which translate the output of the process into the client's language. To reach a recommendation, multi-criteria decision analysis uses different tools for learning and aggregating preferences (see Chapter 1).

More recently, in the field of artificial intelligence, argumentation has been put forward as a very general approach allowing to support different kinds of decision-making [Bonet and Geffner, 1996; Prakken and Vreeswijk, 2002; Parsons and Jennings, 1998; Amgoud and Prade, 2006]. Typically, one will construct for each possible decision (alternative) a set of positive arguments, and a set of negative arguments. However, decision-makers do not simply *list* pro and cons: they

exchange arguments, some of them interacting with others, attacking or reinstalling previous arguments put forward (by the other party, or sometimes indeed by itself).

Our objective in this chapter is to clarify the connections between argumentation and decision-making, and more precisely to inspect the recent proposals that have been put forward to handle decision-making in an argumentative framework. Moreover, we propose to review some approaches, taking a more classical decision theory perspective, more precisely a multi-criteria perspective. In fact, we aim to examine more carefully what it means to argue for an action, especially when different points of view can be considered to assess that action. To guide our discussion, for each approach we will try to provide answers to the following questions:

- ▷ how is the notion of criterion (point of view) captured in this model?
- ▷ how are pro/con arguments constructed?
 - how are the user’s preferences represented?
 - what is the scale used to evaluate outcomes or consequences of the action?
 - is there an explicit reference to a preference model (structure)?

From our point of view, these issues include the major necessary basic elements (see Chapter 1 for details and definitions) to build an argument in favour of an action, by taking into account the different aspects of a multi-criteria evaluation.

The chapter is organized as follows: Section 3.2 is devoted to confront three different proposals put forward in the literature to our multicriteria perspective and discuss some hidden assumptions that they make. In Section 3.3 we summarize the discussion on the different approaches, trying to explain the advantage of relying on arguments and the points which seem absent in these approaches.

3.2 Argument schemes for action: some existing approaches

The aim of this section is to present and discuss three different approaches that have attempted to define an argument for an action. The Example 3.1 will be used to illustrate the different approaches.

Example 3.1. *We want to select a candidate for a given position, and for such purpose we have a number of candidates applying for it. We need to evaluate the outcome of each possible action, that is, how good is the situation induced by accepting each given candidate. For instance, a desired consequence is to have a strong enough candidate as far as academic level is concerned. Let us suppose that this is assessed by using a scale referring to marks, where 12 stands for the point separating the positive side from the negative one. Then, we could say that according to “marks”, we have an argument in favour of accepting this candidate if its mark is more than 12.*

3.2.1 Scheme for actions and values

[Fox and Parsons, 1997] is one of the first work that tried to advocate an argumentative approach to decision-making, building on Fox’s earlier work [Fox *et al.*, 1980]. They recognise and clearly state what makes argumentation for actions different from argumentation for beliefs, and put forward the following argument scheme:

We should perform A (A has positive expected value)
Whose effects will lead to the condition C
Which has a positive value

Table 3.1 — Fox and Parsons’ argument Scheme

This argument can be represented as follows:

$$\begin{array}{rclcl}
 A \rightarrow C & : & G & : & + & e_1 \\
 C & : & G' & : & + & v_1 \\
 A & : & (e_1, v_1) & : & + & ev_1
 \end{array}$$

where in the general formulae $\langle St : G : Sg \rangle$: St (Sentence) represents the claim, G (Grounds) are the formulae used to justify the argument, and Sg (a sign) is a number or a symbol which indicates the confidence warranted in the conclusion. As explained by Fox and Parsons, the advantage of this representation is that it makes explicit three inference steps:

- ▷ e_1 : that C will indeed result from action A ;
- ▷ v_1 : that C has some positive value, and eventually;
- ▷ ev_1 : that A has a positive expected value.

Clearly, steps (v_1) and (ev_1) requires additional information in order to be able to assign values to situations, and to decide whether the action has indeed a positive expected value. The valuation of the condition is subjective (dependent of the agent’s preference), and represented here by “labelling the proposition describing C with a sign drawn from a dictionary”, which can be qualitative or not and plays the role of a *scale*. Interestingly, different values can be assigned to C from different *points of view*. However, it is not clear how we can handle these different points of view in order to reach a conclusion. For instance, one can ask if these points of view are predefined.

We can apply this approach to our example 3.1, then we can say, for instance, opting for a given candidate (say a) could lead to an outcome where the chosen candidate has a mark of 14.

This would be captured by the first epistemical step e_1 of the scheme, where ga stands for the justification of this step. Together with the two following steps, this could be represented with this scheme as follows:

$$\begin{array}{llll}
 chose_a \rightarrow mark = 14 & : & ga & : + e_1 \\
 mark = 14 & : & va & : + v_1 \\
 chose_a & : & (e_1, v_1) & : + ev_1
 \end{array} \quad (case\ 1)$$

The second step (v_1) means that the condition $mark = 14$ is positively evaluated by our agent (noted by symbol +) (it then counts as a positive argument), where va is the justification for this value assignment. Although this aspect is not deeply explored in the paper, a very interesting feature of this approach is that it makes explicit the grounds allowing to assign this value to this condition: what may count as obvious candidates to justify this value assignment, if we take the view of the multicriteria-decision approach, would be *the user's preferences* ("I consider that the mark is good from 12"), as well as *the preference model* used ("a mark is good (or positive) as long as it is beyond the limit previously stated").

We could also directly encode within this scheme that opting for a given candidate would lead to an outcome where the condition that the chosen candidate has a mark over 12 is satisfied, a fact that we consider positive. This could be represented as follows:

$$\begin{array}{llll}
 chose_a \rightarrow mark \geq 12 & : & ga & : + e_1 \\
 mark \geq 12 & : & va & : + v_1
 \end{array} \quad (case\ 2)$$

meaning that the condition $mark \geq 12$ is positively evaluated by our agent (noted by symbol +) (it then counts as a positive argument), where va is the justification for this value assignment. In this case, the nature of this justification is less clear, for it leads to support the agent's preferences.

These two alternative ways of representing argument schemes about actions seem somewhat unsatisfactory. On the one hand, choosing to directly represent the neutral action (here 12) drawn from the agent's preferences (see case 2) drops the relation linking an action and its consequences. On the other hand, not representing it (see case 1) assumes it is somehow encoded within a "value assignment" mechanism. Finally, this approach does not really acknowledge that actions themselves can be evaluated against a number of meaningful, predefined, dimensions: in fact, each condition induces a new dimension against which the action can be evaluated.

To summarize, this approach offers in general, the advantage of representing the reasoning steps to evaluate an action. The conclusions of our analysis (see Table 3.2), is that some elements of a multi criteria evaluation are implicit and others are absent, but this fact remains justifiable, since the basis of this approach was not to represent that elements.

1	<i>Criterion</i>	<i>they refer to the notion of point of view which is slightly different from a criterion</i>
2	<i>User's preferences</i>	<i>they can be explicit (through the condition) or implicit (included in the justification).</i>
3	<i>Scale</i>	<i>it seems be represented through a dictionary $\{++, +, --, -\}$.</i>
4	<i>Preference Model</i>	<i>not clear</i>

Table 3.2 — Fox and Parsons' scheme: a multi-criteria vision

3.2.2 Scheme for practical reasoning

One of the most convincing proposals recently put forward to account for argument-based decision-making is the one by [Atkinson *et al.*, 2006; Atkinson, 2006]. They propose an extension of the “sufficient condition” argument scheme (see Chapter 2) proposed by [Walton, 1996], called, argument scheme for practical reasoning (see Table 3.4).

To define this scheme, the authors have taken Walton's notion of a goal and separated it into three distinct elements: *states* (a set of propositions about the world to which they can assign a truth value), *goals* (propositional formulae on this set of propositions) and *values* (functions on goals). Therefore, unlike the previous approach, the notion of value is used here in a different sense. Atkinson explains [Atkinson, 2005] that values should not be confused with goals as “they provide the actual reasons for which an agent wishes to achieve a goal”.

In the circumstances R
We should perform A
Whose effects will result in state of affairs S
Which will realise a goal G
Which will promote some value V

Table 3.3 — Atkinson's scheme

A given action can bring about a state of affairs that may satisfy many goals, hence affecting different values. Indeed, a function *value* maps goals to pairs $\langle v, sign \rangle$ where $v \in V$, and *sign* belongs to the scale $\{+, -, =\}$. Thus, the valuation of the consequences of an action is based on a scale, related to v , which express the fact the value is promoted or demoted. Therefore, unlike the previous one, this approach addresses explicitly action's consequences, and states actually desired by the agent (preferences).

For instance, using our running example, we could have

$$value(mark \geq 12) = \langle academic_level, + \rangle$$

meaning that the value (criterion) academic level is promoted when the mark is over 12.

In this approach, *values* seem to play the role of *criteria*, in the sense that we can assess the action's consequences according to a point of view (here v). Now, the particularity of a criterion is that it allows to model the agent's desires, in this approach they are specified through the goals (which are related to the values with a value function). However, the declarative nature of goals allows for more flexible classifications than what we typically have in decision models. For instance, it is possible to easily express that

$$value(age \geq 18 \wedge age \leq 32) = \langle youth, + \rangle$$

the value "youth" is only promoted when the *age* falls between 18 and 32. It is also important to note that this approach does not cater for an explicit representation of all the justifications of the value assignment (this only rely on the logical satisfaction: a goal reached or not, which justifies the value assignment). In this case, it is not possible to represent or indeed challenge the preference structured used. We also refer to [Bench-Capon and Prakken, 2006] for a detailed discussion related to this scheme.

1	<i>Criterion</i>	<i>represented by the combination of values and goals</i>
2	<i>User's preferences</i>	<i>represented only by the desired consequences (G satisfied or not);</i>
3	<i>Scale</i>	<i>the scale of V [demoted-promoted]</i>
4	<i>Preference Model</i>	<i>Goal : reached or not \Rightarrow value $\in V$ fall either on the positive or on the negative side.</i>

Table 3.4 — Atkinson's scheme: a multi-criteria vision

To sum up, this approach has greatly extend our understanding of the construction of argument schemes for action. It offers us a good guideline of how to manage different aspects of the evaluation of an action, in the same structure. However, such structure consider only an intrinsic evaluation for the action and in our context, as we shall see later in this document, we need to take into account the relative evaluation, which is the most widespread in decision analysis. Finally, the Table 3.3 synthesizes the key elements of our discussion on this approach.

3.2.3 Argument and multi-criteria decision making

In [Amgoud *et al.*, 2005], the authors proposed an approach explicitly linking argumentation to multi-criteria decision-making. They see an argument as a 4-tuple $\langle S, x, c, g \rangle$ where

- ▷ $S \subseteq \mathcal{X}$: the support of the argument.
- ▷ $x \in \mathcal{X}$: the conclusion of the argument.
- ▷ $c \in \mathcal{C}$: is the *criterion* which is evaluated for x .
- ▷ $g \in \mathcal{G}$: represents the way in which c is satisfied by x (goals).

Where \mathcal{X} represents a knowledge base gathering the available information about the world; \mathcal{X} is the set of all possible decision; \mathcal{C} is a base containing the different criteria and \mathcal{G} is the set of all goals.

It is required that S is consistent when we add the fact that the action x has taken place. Here, in a way that is reminiscent of the previous approach, each goal g is explicitly associated to a criterion by means of a propositional formula $g \rightarrow c$, although the possibility of having goals referring to different criteria is also mentioned. More precisely, the goal g refers to the satisfaction of criterion c . Indeed, each criterion can be translated into a set of consequences. In turn, the consequences are associated with the satisfactory level of the corresponding criterion. This satisfaction is measured on the basis of a bipolar scale which has a neutral point that separate the positive and the negative values. Therefore, in this approach, unlike in [Atkinson, 2005], the use of (bipolar) scale is explicitly mentioned: the goals will fall either on the negative or on the positive side, which represent two subset of consequences. In addition, this approach also allows for a quantitative measure of how good are the attained goals.

To apply this approach to the example 3.1, we may specify that the knowledge base has several layers.

$$G_2^+ = \{mark \geq 14\}; G_1^+ = \{14 > mark \geq 12\}; G_1^- = \{mark < 12\}$$

which means that the marks are considered as “good” from 12, and even “very good” from 14, while it is insufficient when it is below 12. This comes together with formulae of the form

$$mark \geq 14 \rightarrow academic_level$$

which explicitly states that the goal G_2^+ affects the criterion “academic level”. Now each decision will have some consequences, that will in turn fulfill some goals or not. An argument is in favour this decision if this later satisfies positively a criterion. In other terms it satisfies positive goals. However, an argument is against a decision if the decision satisfies insufficiently a given criterion. So, it satisfies negative goals. Thus, it is possible to identify arguments pro and cons a given decision x , by simply scanning the knowledge base and checking which positive (resp. negative) goals are satisfied by the occurrence of a given decision x . For instance, in our example

of choosing a candidate a having a mark = 14, we have an argument in favour of $choose_a$ because it satisfies the positive goal G_2^+ .

In conclusion we can notice that this approach seems to be the first tentative work that explicitly investigates the interest and the question raised by the introduction of argumentation capabilities in multiple criteria decision making. The Table 3.5 sum up the key points of our analysis of this approach.

1	<i>Criterion</i>	<i>the criterion is clearly used</i>
2	<i>User's preferences</i>	<i>represented by both positive and negative consequences</i>
3	<i>Scale</i>	<i>a bipolar scale of satisfaction</i>
4	<i>Preference Model</i>	<i>to reach or not a goal</i>

Table 3.5 — Amgoud and Prade's scheme: a multi-criteria vision

3.3 Discussion

In this chapter we discussed three approaches to argument-based decision-making. What we have seen along this chapter, is that each approach is rather marginally different from the other ones, but they share the fact that a decision process can be represented by explicit and distinct steps. Therefore, these approaches allow to focus on the different aspect of this process. Specifically, [Fox and Parsons, 1997] are the only ones to explicitly represent the justification of a value assignment, however, they do not fully explore this avenue; and hardwire the possibility of having different criteria. [Atkinson, 2005] makes this latter distinction clear, but on the other hand, do not cater for an explicit representation of all the justifications of the value assignment (this only rely on the logical satisfaction: a goal is reached or not, which justifies the value assignment). In this case, it is not possible to represent or indeed challenge the preference structures used. [Amgoud *et al.*, 2005] also rely on the logical satisfaction of goals to justify the value assignment, but the goals are ordered in a way that indeed allows to refine the preference structure, to express various degrees of satisfaction of a goal. Still, this is directly encoded in the knowledge base and cannot be discussed in the process. Also, by using a bipolar scale, they constrain the syntax of goals and prevent themselves from using the full expressivity provided by the logic.

There are, on the other hand, many similarities between these approaches. First, the evaluation is made possible by an explicit representation of the consequences of the action. By relying on logic to represent such states of affairs, it is more expressive than the ordered scale that is usually used in decision models. One further possibility that is offered by this representation is that some action evaluation may be implicit or partial, whereas in decision models you would require each action to be evaluated on each criterion. The third, perhaps most striking similarity, is that they

all rely on a method of *intrinsic evaluation*, and use more or less explicitly a neutral (or fictive) action.

However, if we consider the context of decision aiding process, such approaches do not necessarily meet the expectations of such a field. Indeed, most approaches do not refer explicitly to the criterion which is the main tool to assess and compare alternatives according to a well defined point of view. This concept does not only evaluate actions but reflects the decision maker's preferences. Moreover, unlike in decision analysis, where several different problem statements are allowed (such as choosing, rejecting, ranking, classifying, etc.), the different argumentation-based approaches [Morge, 2007] assume only one kind of decision problem, namely "choosing", where the aim is to select the best solution. Other approaches [Atkinson, 2006; Fox and Parsons, 1997] rely on the intrinsic evaluation of the consequences of an action, while many decision problems involve the relative evaluation of actions. Furthermore, they focus much more on the construction of arguments for and against an action and do not care about the construction of the final recommendation. Finally, several approaches [Amgoud *et al.*, 2005; Bonnefon and Fargier, 2006] used aggregation procedures based only on the number or the strength of arguments, while in decision analysis there exist a range of aggregation procedures. Regarding the latter, one can ask the question of how to justify the use of a procedure rather than another. Indeed, argument schemes can also be designed to make explicit aggregation techniques that can be used on the basis of preferential information.

A first work that tried to introduce argumentation within a decision aiding process is the one of [Dimopoulos *et al.*, 2004]. The aim of the authors was, 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 artifacts of the decision aiding process, in order to detect the parameters that should be updated and therefore to adapt the final decision. To do that, they based their work on an argumentation framework proposed in [Kakas *et al.*, 1994] and developed further in [Kakas and Moraitis, 2003]. Such framework allow to model a priority among a set of rules of a given theory. In fact, unlike the previous approaches discussed in this chapter, this approach is not based on argument schemes but 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. Thus, the authors claimed that such rules allow to handle the update of the artifacts of the decision aiding process.

Thus, the framework proposed by [Dimopoulos *et al.*, 2004] is based on three levels of rules: (i) a number of object level rules showing the relations between problem formulation and evaluations models; (ii) the default context priority rules which help in applying the object level ones, and (iii) the specific context rules which will give priority to the exceptional conditions rules. It is clear that this approach represents a first step toward using argumentation in decision aiding

process. However, some features remains not clear or unsatisfactory. For instance, a decision aiding process is an interaction between an analyst and a decision maker, and in this framework it is not very clear how we can model this interaction, even through an automatic system.

3.4 Conclusion

The primary aim of this chapter was to offer a critical review of three existing approaches adopting an argumentative stance towards decision-making, adopting the viewpoint of multi criteria decision theory. We have seen that these approaches have much to offer to decision model in terms of expressivity, specification of preferences, etc.

In particular, we have seen that the construction of an argument scheme for an action allows to explicitly state what justifies a chosen course of action. They can be based on various notions: underlying motivations, goals, or direct comparison of alternatives based on user's preference statement. Note that by relying on underlying goals, we must then chose a specific criterion to be able to compare two possible states of the world (do I prefer a situation where many secondary goals are satisfied vs. one in which only, but prominent, is?). There are of course many possible options here (see [Bonneton and Fargier, 2006]), that we shall not discuss further. From our brief review, it came out that different approaches make explicit different steps of the process. Moreover, argument schemes are of primary importance: by expliciting the inference steps of an argument, we can also define what counts as valid "critical question", that is how arguments will interact with each others (how they can be attacked and so on). Such critical question will allow to handle revision during the decision process. More prospectively, argument schemes can also be designed to make explicit aggregation techniques that can be used on the basis of preferential information. In fact, as we shall see in Chapter 4, the aggregation may also be disputed, and be based on assumptions that can be challenged and/or revised.

Thus, in the rest of this document, we propose to present our proposals on the to use arguments schemes to represent the steps of a multi criteria evaluation model.

Part

MCDA: a dialectical perspective

4 Different levels for different reasons

Our ambition in this chapter is to show that it is possible and useful to introduce argumentation in a decision aiding process. In fact, we propose to specify in argumentative terms the steps involved in a multi criteria evaluation process. To do that, 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.

4.1 Introduction

In the previous chapter we have seen that there has recently been many proposals to adopt an argumentative approach to decision-making. Such approaches have greatly extended our understanding of the construction of argument schemes for action.

Our research work is also oriented in the same direction but we take a different perspective. We want to build on that but decision aiding is more than simply solving a complex decision model more or less faithful of the decision maker's values and preferences. It involves understanding, interpreting, justifying, explaining, convincing, revising and updating the outcomes of what we call a decision aiding process. Currently, the model of decision aiding process, 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. Thus, what we propose in this work is to try to meet those needs by using the concepts and tools of argumentation theory. Of course, we are aware that taking into account the whole process is a long term research issue. In this work we offer a first contribution in

that direction and address the evaluation phase. Indeed, we focus on the evaluation process that is composed of different stages, mainly: the evaluation of an action on a single criterion, the aggregation step and the construction of the recommendation. Thus, our ambition through this work is twofold.

- ▷ the first objective is a knowledge representation exercise: by casting the different steps of an evaluation process into argumentation framework, we make explicit assumption that are usually hidden in such a process, hence allowing meaningful explanations.
- ▷ the second one is to show that using tools of argumentation facilitates the revision/update occurring during such process. Indeed the argumentation approach comes along with a reasoning machinery which allows to establish which arguments are “acceptable” at a given point of the discussion.

We dedicate this chapter to the presentation of the first part of our research work, where the aim is to provide a way to construct arguments at each level of a multi criteria evaluation process. To do that, we make use of the popular notion of argument schemes. A hierarchical structure of argument schemes is constructed to represent each level of such process. Each scheme will make explicit the reasoning steps of each level. By relying on argument schemes, we will allow, on the one hand, to make justification possible and on the other hand to identify how these steps can be dialectically defeated.

The chapter is organized as follows. In section 4.2, we discuss our point of view on the construction of arguments for action in a context of decision aiding. Section 4.3 presents the different argument schemes that are involved in an evaluation process.

4.2 Arguing over actions

We presented in the previous chapter a set of approaches based on argumentation for decision making. Such approaches have identified a variety of arguments structures allowing to highlight the benefits of argumentation for decision: expressiveness, explicit representation of reasoning steps, etc. Moreover, we presented an analysis of the different approaches, discussing how the elements involved in a multiple criteria decision analysis can be represented. This study showed that, in general, the approaches are more interested in how to build arguments and less on the construction of the final recommendation; they often seek the best choice using procedures based solely on the number and/or strength of arguments; and they rely in general on an underlying intrinsic evaluation of the action.

We propose, in this section, to explain our vision of arguments for actions. More precisely, *to show what are the elements that should be included in an argument to be meaningful in a decision*

aiding context. In order to do that, it is useful to show the links that may exist between the concepts of argumentation theory and those of decision aiding methodology. In particular, we would like to investigate and to discuss more precisely the connections between the concept of criterion and the one of argument.

4.2.1 Criterion vs. Argument

We have seen in Chapter 1 that the concept of criterion is a tool for evaluating an actions (see Section 1.3). From an argumentation points of view, the evaluation of an action can be the result of the construction of positive and negative reasons for that action. Such reasons are formally represented by the mean of arguments, therefore, we can have both arguments in favour and against that action.

Consequently, we have two evaluation tools, but two different practices. An argument is designed more to justify the consequences of an action. The criterion, in turn, is built for purposes of preferences representation. Indeed, the structure (or more precisely the premises) of an argument provides explicit evidence that will be used to support (or not) a certain action. The criterion however, does not seem to have this feature. It certainly helps to model the decision maker's preferences, which then can be used to justify why we can be in favour of an action. The problem is that this information is not explicit and visible for the decision maker. It is not easy to guess what is the model (or reasoning) that helped to promote an action rather than another one.

A further difference between an argument and a criterion, concerns the way by which the actions are compared. Decision analysis allows to identify models of preferences which can be used to compare and choose actions, either on the basis of an *intrinsic evaluation* (the evaluation of an action is based on its comparison to some pre-established norms) or a *pairwise comparison* (the choice is defined with respect to the comparison of the actions among themselves). In argumentation, however, the evaluation is rather intrinsic and the pairwise comparison of actions only comes as a by-product of the construction of arguments pro/ con each alternative. One may argue that, in decision analysis, it is always possible to retrieve pairwise comparison on the basis of intrinsic valuations. But this is more than a simple technicality, because they represent two different models.

Finally, a basic requirement on the criteria set is separability: each criterion alone should be able to discriminate between the actions, regardless of how these behave the other criteria (further conditions can apply, that we shall not discuss here; for more details the reader is referred to [Keeney and Raiffa, 1976; Roy and Bouyssou, 1993; Vincke, 1992]). With arguments, it is not possible to provide such a result on the set of action on the basis of a single argument. Each argument constructed concerns a particular action.

To summarise, the concept of criterion is devoted to model the decision maker's preferences,

and an argument is designed, in general, to explain and justify conclusions. From our point of view, argumentation can be seen as a way to make explicit the reasons justifying each preference ranking among actions. That is, if the decision-maker were to ask the question “*why did you say that you preferred a over b?*”, we may give those reasons. In what follows, we will try to give a formal representation of the structure of such reasons.

4.2.2 Structure of an argument

In what follows, we propose to describe our vision on the structures of arguments for actions. We will see that *such structures will depend on the information available at each level of the evaluation process.*

Before describing the different structures that we propose in this work, we note that unlike most argumentation systems, which suppose the existence of a fixed knowledge base at the beginning of the process, in our context such a base does not exist. Indeed, most argumentation approaches construct their set of arguments on the basis of such base, but in decision context it cannot be the case, because it is not always possible to have all the necessary information at the beginning of the process in order to resolve the decision problem [Gordon *et al.*, 2007].

Therefore, to construct a justification or an explanation, we dispose of an initial set of information which represent the data (problem statement, the set of actions, the set of criteria, etc.) and the preferential information provided by the decision maker about the decision problem. According to the hypothesis 1.3, for our proposal such information are represented under the form of a performance table (if it is necessary, additional information can be added as input of the process). An example of the that table is presented in the Example 4.1.

Example 4.1. *In this example, the decision maker has to make a **choice** between a set of three alternatives $\{a, b, c\}$. Such alternatives are evaluated on three attributes $\{h_1, h_2, h_3, h_4\}$ and the evaluations are summarized in the **performance table** below:*

	h_1	h_2	h_3	h_4
a	350	200	50	100
b	450	100	150	50
c	350	120	100	50

According to the preferences of the decision maker, we should minimize the first criterion and maximize the three last criteria.

Now, for the construction of arguments, we will distinguish on the one side, **two categories** of arguments and on the other side, **different levels** of arguments.

First, the two categories represent the fact that we can have, during the process, two types of information: positive information and negative information. In fact, as it was discussed in subsection 2.2.2 (see Chapter 2), it is possible to build arguments expressing either a “positive” reasons supporting an action or a “negative” reasons against that action. Such positive and negative reasons can be independent so they could occur simultaneously and should be treated separately. On the other hand, we can build different types of arguments depending on the stage (or levels) of the evaluation process. Indeed, an evaluation process can be decomposed into several stages, mainly: the evaluation of an action according to a single point of view (criterion), the aggregation stage that allow to synthesize the set of information on the different criteria, by using a certain aggregation function, and the final recommendation stage, where probably a solution for the decision problem is proposed to the decision maker. Our ambition is to provide explanation at each step of the process. However, the same argument structure cannot be used for all stages. In fact, the information that we have from one level to another are of different types, and the underlying assumptions are different too. For example, at the aggregation level, to justify that an action is better than another one, we need to take into account all the evaluations of both actions on all criteria and probably to precise the aggregation procedure used to get the result, since different procedures provide different results. However, at the level of a single criterion, we need to know the evaluation of the action on this criterion and the preference model used to compare it to another action. Therefore, we plan to adopt for each level its own argument structure. Such structure will explicit the reasoning steps at that level.

To illustrate our proposal, let’s take again the example 4.1. Consider that we are at the end of the evaluation process and that the action a is presented to the decision maker as the “best” choice for his decision problem. A possible reaction from the decision maker is to ask why a is recommended and not another action. To convince the decision maker, a possible argument is the one of the Example 4.2.

Example 4.2. *“Because a is globally better than all other actions, then we have evidence to support the conclusion a as the best”.*

The idea is to start by presenting a “general” explanation without going further in details. Thus, the explanation (or the argument) will mention that the choice is justified only by the fact that it is better in comparison with the other actions that are not recommended. Now, the decision maker can either accept the recommendation or ask for further details. If required, the explanation should gives evidence that the chosed action is indeed better that any other actions. Thus, it is possible de refine the justification and present the argument of the Example 4.3.

Example 4.3. *“because a is ranked first on a strict majority of criteria $\{h_1, h_2, h_4\}$ then we have enough evidence to support the conclusion a is the best.”*

In such argument, unlike the first one, we justify that the action a is the best by *making explicit*

the way by which the result was obtained. In other words, we refer to the aggregation procedure used to calculate the solution. In this case, it is the majority principle (see Chapter 1). Thus, such argument describes the reasoning steps followed at the aggregation level to construct the conclusion by taking into account how the majority works.

Finally, it is possible to refine the explanation and go further in details, by considering the evaluation of the action according to a particular criterion. Indeed, it is possible to refine the justification by explaining why a is better than each other action on each specific criterion. In other terms, to explain each pairwise comparison according to each criterion. For instance, if we consider the action b and the criterion h_1 , the argument of the example 4.4 justifies why a is better than b on that criterion.

Example 4.4. *Because we have on h_1 : $h_1(a) <_1 h_1(b)$ then we have evidence to support the conclusion “ a is better than b on the dimension h_1 ”.*

Such argument explains the relation between a et b on the criterion h_1 by revealing the preference relation that allows to compare the best action to another action. Such relation is built on the basis of the information provided by the decision maker. Thus, such argument explain the evaluation of an action relatively to the action b . It is what we call in decision analysis “relative or pairwise evaluation ” according to a particular point of view. The construction of such argument is illustrated in the figure 4.1. It suggests that justifying a binary relation between two alternatives lies in making explicit the comparison of the consequences of two actions (valuations) using a particular preference model.

Furthermore, there is another way to evaluate actions, which is the intrinsic evaluation (see Section 1.3 in Chapter 1). In this case, the construction of an argument follows the same reasoning as the relative evaluation, except that the action is compared to an implicit action (represented by the neutral point). At the elementary level, those steps are summarised in the figure 4.2, and will be designed under the form of an argument scheme according to a particular point of view, where the structure of such argument is summarized as follows:

Conclusion of the argument Regarding the conclusion of an argument we can distinguish two types depending on whether the assessment of the alterative is intrinsic or relative:

- ▷ ***intrinsic (or absolute) valuation:*** The conclusion C of the argument in this kind of valuation can be: $C = a$ is acceptable; $C = a$ is bad; etc. The conclusion is the result of the comparison of the action against a (sometimes implicit) neutral point: $a \succeq_i P_i$ (such that p marks the limit between two categories) on a specific dimension i .

- ▷ **relative valuation (or pairwise comparison)**: the conclusion in this kind of evaluation is under the form of $C = a \succeq_i b$. The proposition must be read as “ a is at least as good as b ” on the dimension i . This type of comparison is explicit in any multicriteria decision analysis and implicit in all the other cases.

Premises of the argument In our context, the premises of the argument according to a single point of view can only be based upon the information provided by the decision-maker’s preferences and the performance table which contains the scores of the alternatives on the criteria considered.

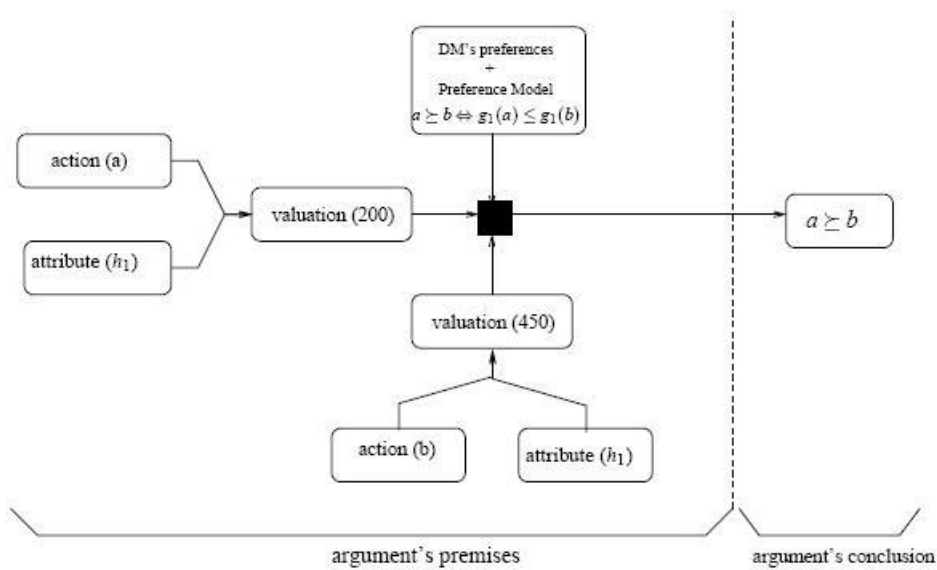


Figure 4.1 — Example of an argument for pairwise comparison

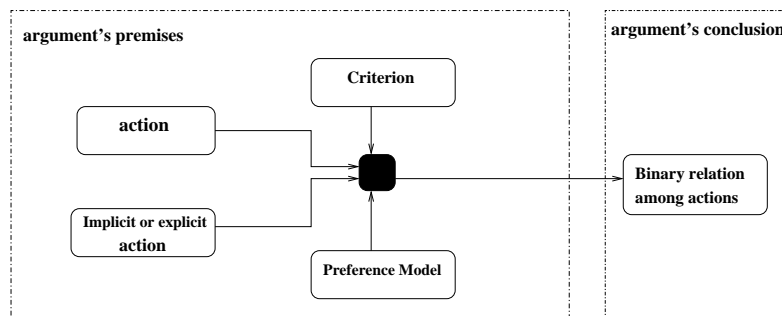


Figure 4.2 — General structure of an argument

4.2.3 Aggregation procedures vs. Proof standards

In practice, the criteria are rarely unanimous in the comparison of alternatives, and we need to resolve the conflicts. On the other, our claim is that, for each criterion it is possible to construct an argument that will reflect the position of that criterion against a given action. Thus, to resolve the conflicts between criteria, we need to aggregate the corresponding arguments.

There exist several procedure to aggregate arguments. They are generally based either on the number of arguments, or on their strength (for examples see [Bonneton and Fargier, 2006]). In the case of our model, we propose to rely on multi criteria aggregation procedures. In fact, we have the aim to take into account the fact that there exist several contradictory points of view for evaluating an action. In the argumentation based approach, it is not always clear how the multi criteria feature is taken into account when it comes to aggregate several arguments that may potentially interact. Thus, it is necessary to use a procedure that allows to handle potential interactions between arguments that refer to different criteria. Moreover, it is important to preserve at the aggregation level all information included in the basic arguments because it will allow us to build justifications that are more specific and precise. Therefore, in our context, an aggregation procedure can be seen as a mean to aggregate the different arguments pro and con a conclusion, that are constructed at the basic level (according to a single point of view) in order to provide *enough supportive reasons* for that conclusion.

At the same time, argumentation in our context can be seen as the process for justifying choices and decisions. Thus, under such a perspective, the concept of *proof standard* (see Chapter 2), is also a rule for aggregating arguments pro and con a given issue. In other terms, a *proof standard* allows to answer the following: since there will be arguments both for and against a given issue, when shall we decide that there is enough supportive evidence?

Consequently, we can observe that it is possible to consider *proof standards and aggregation procedures* as equivalent. Indeed, it is possible to “retrieve” many aggregation procedures by means of argumentation. This is done for example in [Amgoud *et al.*, 2005] where the authors show how to capture different aggregation techniques by means of argumentation. We should be prudent with such a comparison though. On the one hand, as noted in [Amgoud *et al.*, 2005], only the most simple aggregation procedures can be captured. This is essentially due to the qualitative nature of the argumentation process. On the other hand, multicriteria aggregation makes assumptions that argumentation would *not* do when defining proof-standards. Let us inspect this in more detail. [Prakken, 2005b] identifies some requirements that formal accounts of accrual should meet. The fact that arguments cannot be considered as being independent is one of them. For instance, Prakken uses the following example: it may be enjoyable to jog in hot, raining, weather; but not in hot and dry weather, nor in cold and raining weather. In terms of multicriteria decision-making, if we were to define two criteria *dryness* and *warmth*, they could not be inde-

pendent. This is something which is explicitly prohibited in Multicriteria decision aiding, where criteria are assumed to be independent.

Thus, in the rest of this document, we will use the term proof standard or aggregation function to refer to the same thing, i.e. a rule allowing to aggregate a set of arguments at the aggregation level of the hierarchy. However, unlike a proof standard, an aggregation procedure is not fixed by the context and depends on the available informations during the process. Usually in argumentation, the notion of proof standard that should be used is determined by a given context. This is especially true in the legal domain, where the proof standards that should be used are well identified. In this work, we are more specifically concerned with a context of multi criteria decision-aiding. In this context, a specificity is that the proof standard is not exogenously given: it is clearly part of the data that are discussed by the expert and the client. This point was emphasized by [Bouyssou *et al.*, 2006]:

“It is important not to choose the method before the problem has been formulated and the evaluation model constructed, but to show that is the natural consequence of the decision aiding process as conducted up that moment.”

In other terms, the aggregation procedure is considered like any parameter of the evaluation model (see Section 1.2) and it is designed when the model is constructed. During the decision aiding process, we can have various versions of the evaluation model (see Section 1.5), according to the information provided by the decision maker. From one version to another, the aggregation procedure may change in order to be adapted to the new information. In short, the aggregation can be discussed. In this work, we propose to make explicit in our model the proof in use and to adapt for each procedure its own structure of argument (see Figure 4.5).

To sum up, we have seen in this section that in order to represent the whole evaluation process, we need to identify for each level a specific structure of argument. Each structure will represent the reasoning step at that level, and, then, will be designed by a particular argument scheme. The idea behind the use of such a structure is that they allow us to represent explicitly the reasoning steps followed in that level. Furthermore, this representation, as we shall see later, will offer a way to discuss each phase or element of the process through the concept of critical questions (see Chapter 2). This discussion plays an important role in the update and revision process that may occurs in the course of the evaluation process. In what follows, we present the different argument schemes proposed in this work.

4.3 A hierarchy of schemes

In this section we extend the use of arguments to the various stages of the evaluation process. We propose to construct different classes of argument schemes (see figure 4.3) to represent the

reasoning steps within this process. Thus, we have distinguished different classes of argument depending on:

- ▷ whether they are concerned with a single criterion or with the aggregation of several criteria, (unicriteria vs. multicriteria);
- ▷ whether they follow a pairwise evaluation or whether they use an intrinsic evaluation, the action being compared to a neutral point (pairwise vs. intrinsic);
- ▷ whether they are concerned with the evaluation of the action or its mere acceptability (evaluation vs. acceptability); and
- ▷ whether they are concerned with a positive reasons or negative reasons (positive vs negative).

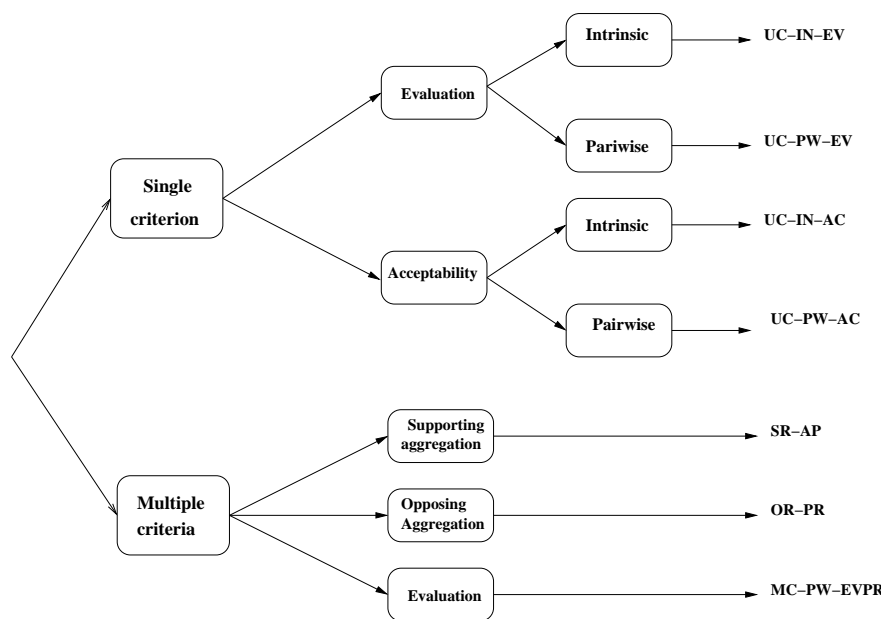


Figure 4.3 — Different classes of argument schemes

In theory, all combinations seem possible, even though some are much more natural than others. These different categories allow us to distinguish the following schemes

- ▷ Schemes based on single criterion:
 - argument scheme for UNICRITERIA PAIRWISE EVALUATION (UC-PW-EV), which establishes that an action is at least as good as another action from the single viewpoint of the considered criterion;

- argument scheme for UNICRITERIA INTRINSIC EVALUATION (UC-IN-EV) which concludes that an action is at least as good as a neutral point from a single criterion;
 - argument scheme for UNICRITERIA INTRINSIC ACCEPTABILITY (UC-IN-ACC), which establishes that the action can be acceptable, on single criterion, regarding a neutral point;
 - argument scheme for UNICRITERIA PAIRWISE ACCEPTABILITY (UC-PW-ACC), which concludes that an action is acceptable regarding another action of the set A.
- ▷ Schemes based on several criteria:
- argument scheme for MULTICRITERIA PAIRWISE EVALUATION (MC-PW-EV), which basically concludes that an action is at least as good as another action on the basis of several criteria taken together.
 - argument schemes for SUPPORTING REASONS AGGREGATION PROCESS (SR-AP), which concludes that there are enough supporting reasons in favour the claim of MC-PW-EV, according to some aggregation procedure.
 - argument schemes for OPPOSING REASONS AGGREGATION PROCESS(OR-AP) which concludes that there are strong negative reasons that block the conclusion of MC-PW-EV;

Specifically, such schemes are under the form of a *hierarchical structure*, where we identify three levels of argument schemes that are embedded, as it is illustrated in the Figure 4.4. At the highest level the *multi-criteria level*, which is based on the *aggregation level*, which is in turn based on the *Unicriteria level*. We do not claim that the presented list is exhaustive, but simply that for representing the evaluation process, we believe that the different schemes are extensively sufficient.

In what follows, we present the description of the each argument scheme. We start with the lowest level of the hierarchy.

4.3.1 Elementary (or Unicriteria) level

The first way to perform an action evaluation is to compare two actions from the point of view of the chosen criterion: this is modeled by the scheme for UNICRITERIA PAIRWISE EVALUATION (UC-PW-EV), in the Table 4.1. This argument 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 (or model), noted \succeq_i .

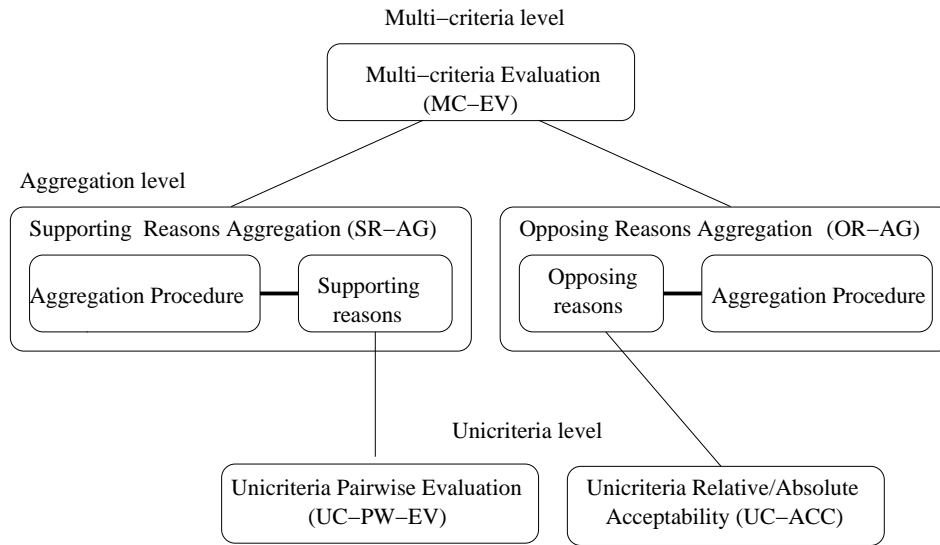


Figure 4.4 — A hierarchy of argument schemes

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 4.1 — Scheme for UNICRITERIA PAIRWISE EVALUATION (UC-PW-EV)

If we consider the example 4.4 again, then we can construct, based on table 4.1, the argument of the table 4.3.

Premises	a criterion	h_1
	an action	a
	whose performance is	200
	an action	b
	whose performance is	450
	a preference relation	$a \succeq_1 b \Leftrightarrow h_1(a) \geq h_1(b)$
Conclusion	a is better than b	$a \succ_1 b$

Table 4.2 — Example of scheme for UC-PW-EV

In the same way as the relative evaluation, we can associate a specific argument scheme to the

intrinsic evaluation. The detail of this scheme is given in the Table 4.3. Very broadly speaking, the idea is that an action must be assigned to a given category (for instance, good, bad, expensive, etc.). These categories are defined by what we call profiles or neutral points (p_i) that determine on each criterion a point ($h_i(p_i)$): this is by not a necessarily an existing action, but it allows to define to which category to affect the action. This type of evaluation is generally used in problem statements such as sorting problem but also for all the argumentation-based aggregation techniques where we have “pro” and “con” categories.

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	$a \succeq_i p_i$

Table 4.3 — Scheme for UNICRITERIA INTRINSIC EVALUATION (UC-IN-EV)

A version close to this argument is presented in the table 4.5, where the categories are a priori known (acceptable vs not acceptable) and the profile has a specific role.

Furthermore, we note in the scheme that for each action we have its corresponding performance. This is true only if the performance table is an explicit input to the evaluation model, which is the case in this work. Indeed, it can be the case where only ordinal information about the alternatives along each dimension (for instance in voting theory) is available. This information can be represented through a binary relation. Accordingly, the scheme will not take into account the performance but only the preference relation, as it shown in Table 4.4;

Premises	a criterion	h_i
	an action	a
	an action	b
	a preference relation	\succeq_i
Conclusion	a is at least as good as b	$a \succeq_i b$

Table 4.4 — Scheme for UNICRITERIA PAIRWISE EVALUATION (version 2)

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's 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 the performance of the action *a* on the other criteria). The scheme of the table 4.5, illustrates such idea.

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 4.5 — Scheme for UNICRITERIA INTRINSIC ACCEPTABILITY (UC-IN-ACC)

The example 4.5 presents a case where the scheme UC-IN-ACC can be used.

Example 4.5. *Suppose that among the preference statements given by the decision maker, we have the following information: “an action may not be considered better than another one if it exceeds the budget of 350”.*

Before adding such information, the conclusion of example 4.4 was “*a* is globally better than *b*”. However, in the light of the new information we can retract the previous conclusion and more precisely to reject it. Indeed, such information carries a negative reason against the conclusion, which can be represented under the form of a negative argument as follows: “because it exceeds the threshold (here 350), the consequence of the action on the dimension price is not acceptable. Therefore, it is quite reasonable to reject the claim *a* is globally better than *b*. This argument is represented by the table 4.6.

Premises	a criterion	<i>price</i>
	an action	a
	whose performance is	400
	a veto threshold	350
Conclusion	a is unacceptable according to h_1	$h_1(a) > 350$

Table 4.6 — Example of scheme for UC-IN-ACC

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 of the set A . In this case we talk about an ARGUMENT SCHEME FOR RELATIVE ACCEPTABILITY, represented in Table 4.7.

Premises	a criterion	h_i
	an action	a
	whose performance is	$h_i(a)$
	an action	b
	whose performance is	$h_i(b)$
	a veto threshold	δ_i
Conclusion	a is unacceptable according to h_i	$h_i(a) - h_i(b) < \delta_i$

Table 4.7 — Scheme for UNICRITERIA RELATIVE ACCEPTABILITY (UC-PW-ACC)

For instance, instead to reject the claim on the basis that the action exceeds a threshold, as in example 4.5, we can imagine that a difference of 50 between the two actions, on the first dimension, can be sufficient to reject the proposition $a \succeq b$, which is the case in this example.

Let us remark that the notion of relative acceptability is not defined only in terms of difference of performances, but we can define other cases, as it is indicated in the example 4.6.

Example 4.6. *Consider different candidates participating to a beauty contest. Different dimensions are used to judge them: the size, the hair colour, the eyes colour, etc. (obviously, on each attribute preferences are expressed). One of the jury member, influenced by the colour of a candidate x , gives a special attention to the dimension eyes colour and states: “if the eyes of a candidate y are not at least blue as those of the candidate x , therefore y can not be at least as good as x ”*

In this example, the acceptability of an action is not determined on the basis of the difference in the performances, but by the fact that the action y does not have the same value as the action x on the dimension “colour of eyes”. A consequence, the action y can not be considered better than (or at least as good as) the action x , even if it is better on all (or the most) remaining criteria.

Finally, we propose to construct two further arguments schemes, as illustrated in Table 4.8 and Table 4.9, that will allow us to justify, respectively, the fact that an action is at least as good as any other actions of the set A on a single criterion, and that an action is better than any action of a subset A' of A on a single criterion. The objective of these schemes is to offer a natural way to express which action is preferred on the criterion. The basic input of information that we need to provide to these schemes is that of pairwise comparison against each other action on a single dimension i (i.e., the output of UC-PW-EV). Thus, on each criterion, we can construct one of such schemes.

Premises	a criterion	h_i
	the set of actions	$A \setminus \{a\}$
	an action	a
	the pairwise evaluation of a and each action $x \in A$ on h_i	$a \succeq_i x$
Conclusion	a is at least as good than any other actions on h_i	

Table 4.8 — scheme for the UNICRITERIA TOP RANKING (UC-TOP)

Premises	a criterion	h_i
	a subset of $A \setminus \{a\}$	$A' \subset A$
	an action	a
	the pairwise evaluation of a and each action $x \in A'$ on h_i	$a \succeq_i x$
Conclusion	a is better any $x \in A'$ on h_i	

Table 4.9 — Scheme for UNICRITERIA GOOD RANKING (UC-GOOD)

These two schemes will represent, as we shall see in Chapter 5, the basic blocks to construct an explanation for the final recommendation of a decision problem.

4.3.2 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*.

4.3.2.1 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 facing a set of arguments pro and con the claim. Obviously, these arguments reflect the position of each criterion against this claim. What conclusion to provide depends entirely on the procedure or the rule that we should use to aggregate all the arguments that are both in favour and against that conclusion. Different procedures necessarily provide different results. Thus, we have to make a choice among the variety of existing multi criteria aggregation methods. Such a choice is difficult because it depends essentially on the decision maker’s preferences during the decision aiding process. We mentioned before, the aggregation procedure can be itself subject to challenge and discussion (see subsection 4.2.3). In order for the model to allow, on the

one hand, to link each result to its aggregation procedure and on the other hand to discuss the choice of a given procedure, we have to permit *an explicit representation of the proof* that will be used at that level of the process (see Figure 4.5). The intuitive reading of such a link is: “we have enough supporting reasons *according* to this aggregation procedure”.

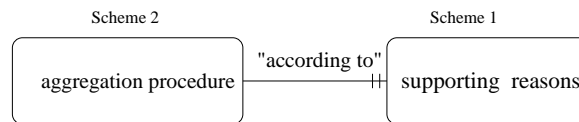


Figure 4.5 — Explicit representation of the proof standard

Therefore, the first scheme given in the table 4.10, allows to conclude that we have effectively enough supporting reasons for the claim according to some aggregation procedure $r_i \in R$. The basic input information that needs to be provided to this schemes is that of a pairwise comparison on a single dimension (the output of UC-PW-EV).

Premises	a set of pairwise evaluation of actions a and b on each h_i	$a \succeq_i b$
Conclusion	there are enough reasons to support $a \succeq b$ <i>according To</i> a selected procedure “ r_i ”	

Table 4.10 — Scheme for SUPPORTING REASONS (SR)

The second scheme formalizes the operation mode of an aggregation procedure (scheme 2 in the Figure 4.5). It provides us, among others, the explanation of how the supporting reasons were constructed. Since each procedure is different from another, this justifies the fact that we have different schemes and not just a generic one. Moreover, the use of each procedure implies certain assumptions on the preferences of the decision maker (see subsection 1.4.2.2 in Chapter 1). Thus, each scheme will include a particular premise summarizing the necessary assumptions or conditions under which it is reasonable to use such a procedure. We shall see in the next subsection that such assumptions will be represented under the form of properties. Moreover, such assumptions will help us to guide the choice between two different procedures. In fact, we discussed in Chapter 1 the problem of making a choice among methods during the decision aiding process. In this work, we propose to provide a solution by identifying different properties describing each procedure.

Before to present our idea, we address, in what follows, the schemes associated to the following aggregation procedures: the simple majority principle, the lexicographic method, the weighted sum and the weighted majority method (see Chapter 1 for a description of each procedure).

Scheme for the Majority method. the scheme of the Table 4.11 makes explicit that criteria are considered to be of equal importance. It is clear that this is not necessarily always the case and other schemes will potentially require additional information.

Premises	a set of criteria $ \{i \in N : aS_i b\} \geq \{i \in N : bS_i a\} $ a set of sufficient conditions to use the majority	$\{h_1, h_2, \dots, h_n\}$ $a \succeq b$
Conclusion	Majority is the selected procedure	$r_i \simeq \text{Majority}$

Table 4.11 — Scheme for Argument from the Majority method (MAJO-SCHEME)

Lexicographic method. 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. No compensation is possible. The Table 4.12 gives the details of the corresponding argument scheme.

Premises	a set of criteria a linear order $a \sim_i b$ $a \succeq_k b$ a set of sufficient conditions to use the lexicographic method	$\{h_1, h_2, \dots, h_n\}$ $>_l$ $\forall j > k$
Conclusion	Lexicographic is the selected procedure	$r_i \simeq \text{Lexicographic}$

Table 4.12 — Scheme for Argument from the Lexicographic method (LEXICO-SCHEME)

The weighted majority method. The Table 4.13 makes explicit the different assumptions to use the Weighted majority procedure.

Premises	a set of criteria a set of importance coefficients $W_{ab} = \sum_{i: aS_i b} w_i \geq W_{ba} = \sum_{i: bS_i a} w_i$ a set of sufficient conditions to use the weighted majority	$\{h_1, h_2, \dots, h_n\}$ $\{w_1, w_2, \dots, w_n\}$ $a \succeq b$
Conclusion	Weighted Majority is the selected procedure	$r_i \simeq \text{Weighted Majority}$

Table 4.13 — Scheme for Argument for the Weighted Majority (WEIGHTEDM-SCHEME)

The weighted sum (a particular case) Unlike the aggregation methods, cited above, which are based on comparison and then aggregation, this procedure involves the comparison after the aggregation phase. In fact, for each action a , we have to calculate a global $score(a)$ on the basis of the performances or the evaluations of a on the set of criteria H . This score will be used in pairwise comparison between different actions. It is very common to use such approach in practice (e.g. for classifying students). Consequently, the scheme of such procedure, summarized in the Table 4.14, reflects both the calculation of scores and the comparison of actions.

Premises	a set of criteria	$\{h_1, h_2, \dots, h_n\}$
	a set of trade-offs	$\{w_1, w_2, \dots, w_n\}$
	Score(a)	$\sum_{i=1}^{i=n} w_i h_i(a)$
	score(a) \geq score(b)	$a \succeq b$
	a set of sufficient conditions to use the weighted sum	
Conclusion	The weighted Sum is the selected procedure	$r_i \simeq$ Weighted Sum

Table 4.14 — Scheme for Argument from the weighted Sum (WEIGHTEDS-SCHEME)

Now, the question that arises is how to choose among these different proof standards? We have seen, in Chapter 1, that Conjoint measurement theory offers a theoretical solution to tackle such a problem during the decision aiding process. Such solution is based on axioms that are used to specify the conditions under which it makes sense to apply a given procedure. However, as it was mentioned in the same chapter, such a solution is difficult to apply in practice. Therefore, we propose in what follows, to identify a *set of properties* to distinguish between the different proofs.

4.3.2.2 Set of “properties” to support the choice of a proof

In this work, we propose a way to distinguish between the variety of aggregation procedures presented above, by identifying each one by a particular set of properties. Such properties will correspond to some characteristics that allow to describe the decision maker’s preferences. Thus, for each combination of such properties will correspond a particular aggregation procedure. Our idea is that we would have properties that are more “simple” and more “intuitive” (or less technical) than the axioms, that suffice to define a method and to justify its use during the process. In other terms, such a set of properties will correspond to the set of *sufficient conditions that support the use of a certain procedure* (see figure 4.6).

To determine such a set, we have considered the fact that we have two sides at the aggregation level: a positive side and negative one. For each side, it is possible to use a specific rule for

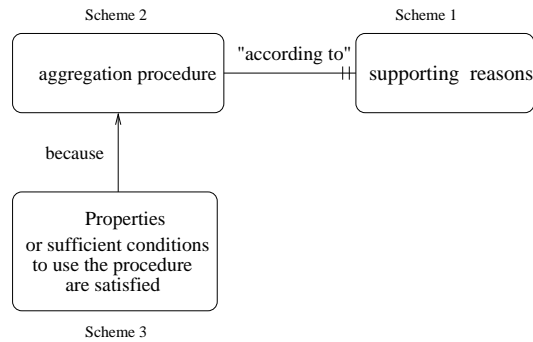


Figure 4.6 — Explicit representation of the proof standard

aggregating all arguments constructed in the lower level. Regarding the negative side, as we shall see later in this section, it will be represented mainly by a *veto*. Therefore, among the properties considered, we will take into account the fact that we may have a negative part or not. Regarding the positive side, as we have seen previously, various functions are possible. To distinguish them, we based our intuition on the following observation: in practice, the aggregation procedures that a decision maker may consider the most “simple” and “intuitive” are: the *simple majority* method and the *weighted sum* procedure. Indeed, among all the aggregation procedures that exist in the literature, these two procedures are very simple to use and do not require a big cognitive effort from the decision maker to understand them, hence, the following question: *is it possible to distinguish these two methods by a set of properties, which are easy to understand even for a non-expert in decision support?*

If we consider the weighted sum, the first thing that we can note is that this method belongs to the classe of additive value models. Thus, the main property characterizing these models is the *additivity in terms of values*. The idea is that each action is summarised by an overall utility that is the aggregation of the performances of the action w.r.t the different criteria.

Such a property cannot be considered for the simple majority method. In fact, the first assumption to use the latter is that to compare actions, only the order of performance is relevant and not the performance themselves. In other terms, the difference among performances is not considered. Thus, contrary to the weighted sum, the simple majority belongs to the class of *Ordinal* methods. Moreover, to use such method, the criteria should play the same role, i.e., there is no importance between criteria and there is no effect on the result if we substitute a criterion by another one. This is what we call the *anonymity* property.

A further difference is that the majority model assumes a different kind of additivity, called *additivity in terms of coalitions*. In fact, in the additive model, to compare two action x and y , we should take into account the contribution of the criterion and the action’s values on this criterion. Formally, we can write: $x \succeq y \Leftrightarrow u(x) = \sum_i w_i h_i(x_i) - u(y) = \sum_i w_i h_i(y_i) \geq 0$, such that

w_i is the weight associated to the criterion i . In other terms, a large difference of performance on one dimension can be compensated by a conjunction of small differences of opposite sign on other dimensions: the procedure is compensatory. However, in the case of the majority model, the only thing that counts is the contribution of each dimension in the comparison, i.e., if we consider that $\Delta_i = h_i(x_i) - h_i(y_i)$, therefore we have $\Delta_i = +1$ if $h_i(x_i) - h_i(y_i) > 0$, $\Delta_i = 0$ if $h_i(x_i) - h_i(y_i) = 0$ and $\Delta_i = -1$ if $h_i(x_i) - h_i(y_i) < 0$. In other terms, one can see that the full contribution of a criterion is credited to an alternative x , as compared to an alternative y , as soon as the performance difference is in favour of x on that criterion. Thus, the idea is to see if the contribution of each criterion is additive or not. We note that in the rest of this document, to avoid ambiguity and to distinguish the two kind of additivity, we will note the additivity w.r.t coalition by *add-c* or *additivity-c*.

Thus, in order to distinguish the two models we are going to use four properties, which are summarized as follows:

- ▷ **Ordinality**. It suggests, when we compare actions, that only ordinal information carry by the performance is relevant.
- ▷ **Anonymity**. It suggests that all criteria are exchangeable;
- ▷ **Additivity w.r.t coalitions**. It suggest that it is possible to formulate an additive value for the importance of a coalition of criteria.
- ▷ **Additivity w.r.t values**. It suggests that the value of an alternative results from the sum of its values on each criterion.

such that the simple majority has the three first properties and the weighted sum the last one. Finally, we added a further property,

- ▷ **Dependence between criteria**. It suggests that preferences expressed on a criterion depend on preferences expressed on another criterion.

Such a property allows us to distinguish the class of *non additive models*, for which the last four properties were not sufficient to identify them clearly.

Finally, considering the five properties and the veto, we tried to identify what are the procedures that we can obtain, by taking into account the fact that some combinations are theoretically not possible, such as ordinality and additivity w.r.t value, anonymity and dependence between criteria or anonymity and additivity w.r.t value. The results are summarized in the Table 4.15

Procedures	Ordinality	Anonymity	Add-c	Add	veto	Dependence
Simple majority	×	×	×			
Lexicographic method	×		×			
Weighted majority						
Weighted majority + veto	×		×		×	
Simple majority + veto	×	×	×		×	
Weighted sum				×		
Non additive model						×

Table 4.15 — Set of properties for identifying proof standards

To sum up, to justify the choice of a procedure among a set of different proofs standards, we propose to rely on a set of properties that establish sufficient conditions to use a certain procedure. From our point of view, unlike the axioms proposed by Conjoint Measurement theory, such properties seem more intuitive and easier to use in practice, because they represent natural characteristics of decision maker's preferences. We note that we do not pretend to replace the axioms presented by Conjoint Measurement, which represent a solid theoretical foundation for characterising aggregation procedures. Our modest ambition is to propose a first approach to guide and to justify the choice of a proof, during the decision aiding process, which is easy to implement. The technical details of how to use such properties for that purpose, are discussed in Chapter 5.

4.3.2.3 Argument schemes for Aggregating opposing reasons

The last step before to conclude that effectively “ 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 which 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. It acts independently and only in a negative sense.

In general, as in the positive side, we can imagine the use of any aggregation procedures to construct the negative reasons, either multi-criteria procedures, or that 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 will aim to synthesize 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 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 dimension [Roy, 1968, 1971; Maystre *et al.*, 1994].

The first condition in ELECTRE I can be mainly represented by a weighted majority, where an *importance coefficient* w_i is associated to 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. The Table 4.16 gives the details of the scheme representing such a rule.

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

Table 4.16 — Scheme for opposing reasons

4.3.3 Multicriteria Level

The last scheme that we address in this chapter is the scheme for the multicriteria pairwise evaluation, illustrated in Table 4.17. The idea behind this argument is to be able to justify that the global relation “ a is at least as good b ” is warranted because it was possible to construct enough supporting evidences (SR) and there was not enough opposing reasons (OR) to contradict it.

Premises	an action	a
	an action	b
	a set of criteria	$\{h_1, \dots, h_n\}$
	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 4.17 — Scheme for MULTICRITERIA PAIRWISE EVALUATION (MC-PW-EV)

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.

To sum up, we presented in this section several types of argument schemes to specify the reasoning steps of a multi criteria evaluation process. These schemes represent different levels, where the information is more general at the top of the hierarchy and it becomes more precise at the bottom. The construction of the hierarchy is the result of a dialogue between the analyst and his client. The analyst 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 build on the basis of the information provided by the client, and in some cases by using default instantiation (when the scheme allows for it). Thus, the levels are folded as boxes where the largest (the highest) contains the smallest (the lowest) one (see Figure 4.7). Now, if the global relation is challenged by the client, the analyst provides the different steps of reasoning by revealing parsimoniously the lower level schemes that compose the conclusion at the highest level.

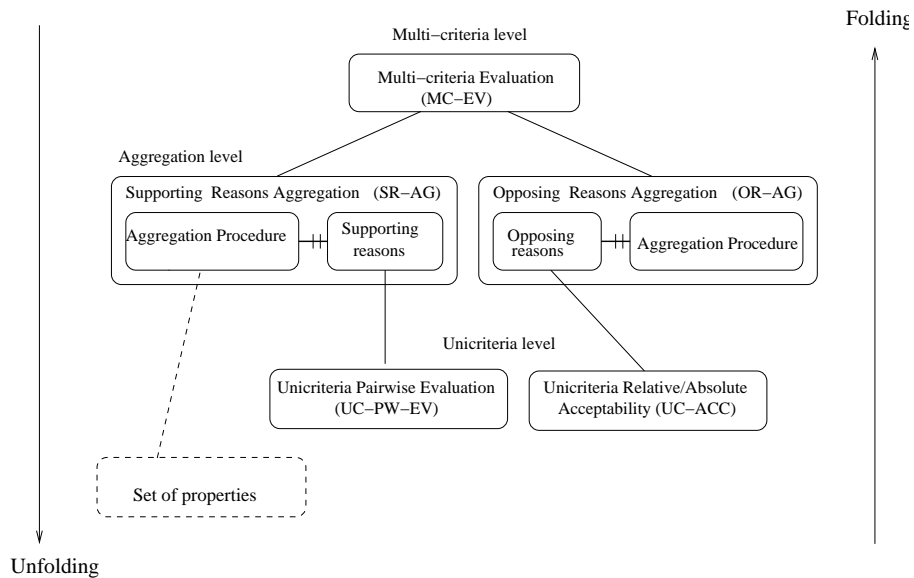


Figure 4.7 — Abstraction levels of argumentation

However, we draw the attention of the reader to the fact that the argument schemes presented throughout this chapter, will allow to build 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, according to the assumption 1.2, we need to compare each action of the set A to all the other actions, and make a choice among the set of actions of 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, as we have seen in Chapter 1 (see subsection 1.4.3), calculating such a recommen-

dation can be done directly or indirectly because the result to make to the client 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 calculation of the final recommendation of a decision problem. At this level, we suggest to build arguments in order to justify the final recommendation. 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 for instance (see subsection 1.4.3) used to provide the final output. The table 4.18 illustrates such an idea.

Premises	a set of actions	A
	a set of criteria	$H = \{h_1, \dots, h_n\}$
	Conclusion of MC-PW-EV for pairwise comparison of a and each action $x \in A$	$a \succ x$
Conclusion	a is recommended <i>according to</i> a choice procedure	

Table 4.18 — Scheme for recommendation (RECOM-SCHEME)

4.4 Conclusion

The purpose of this chapter was to provide a first approach to represent the reasoning steps of a multi criteria evaluation aiding process by means of argument schemes. To do that, we identified a hierarchical structures of argument schemes. Each level refers to one step in the classical multi criteria evaluation. The highest level represents the pairwise evaluation, which is based on the aggregation level, which is in turn based on evaluation on a single dimension (pairwise or intrinsic).

These different schemes will be the basis of the process of explanation (and justification) that we want to introduce within an evaluation process. On the one hand, the justification may serve as a basis for the analyst to convince the client of the consequence of the evolution of the model during the process, to justify the solution, to explain the usefulness of a method and the need to use a parameter rather than another, etc. On the other hand, the formal justification can serve, also, the decision maker himself. Indeed, the decision maker can be confronted with a situation where he has to explain why such a recommendation or a model to others participants involved in the decision process but not necessarily in the decision aiding process.

We believe that the originality of our work is based on the fact that we have proposed to construct argument schemes at all levels of an evaluation process and even at the aggregation phase. Indeed, to our knowledge, no approaches have looked at the construction of schemes for such phase, whereas we believe that way basic argument schemes are collected and aggregated may also be disputed, and be based on assumptions that can be challenged and/or revised.

Thus, in the next chapter we will discuss, on the one hand, how to handle such revision and update during the evaluation process, and on the other hand how to use the different argument schemes presented in this chapter to construct formal explanations during this process.

5

Towards an argumentation-based decision aiding tool

The aim of this chapter is to initiate a systematic study of the use of argumentation in decision aiding tools. The kind of system that we foresee here would allow: (i) to present a recommendation that can be explicitly justified; (ii) to revise any piece of reasoning involved in an evaluation process, and be informed of the consequences of such moves; and probably (iii) to stimulate the client by representing visually the exchanged arguments. To meet such different objectives we propose on the one hand, methods for generating explanation and on the other hand different tools for handling revision that occur during such a process.

5.1 Introduction

During the evaluation process, the analyst and the client are engaged in an interactive 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 needs of the client, the analyst needs to make assumptions and reason as if these assumptions were true. The recommendations, that is the outcomes of the evaluation process, are subject to the client validation. Rejecting of one (or all) of such recommendations means that some of the assumptions made by the analyst may be false and must be retracted or revised. In fact, there are very different reasons to revise in such a process: in some case the decision maker may simply want to correct/refine one of its previous statement, introduce new information. In other cases it will contradict one of the analyst's assumptions. Moreover, during the interaction the decision

maker may express some doubts and ask for explanation or justification.

In practice, it is usually assumed that these aspects are handled by the decision analyst, but if we are to automate (some part of) the process, it is important to understand more clearly how these aspects can be handled in a tool. In fact, the decision analysis literature is mainly focussed on the interactions between humans (mainly the clients and the analysts). There is very little attention, if any, to the use of decision theories and decision aiding methodology when this interaction is between a human (a user) and an automatic device. Thus, in this work we propose to initiate a systematic study which aims at constituting a significant step forward for forthcoming decision aiding tools. Indeed, our ambition through this work is, on the one hand, enhance the capability of the decision support to provide explanation by providing methods for justifying their advice or recommendation in an intuitive manner and on the other hand, to offer the opportunity to the user to observe step by step the effects of the interaction on the final result.

Under such a perspective, the kind of system that we foresee here would allow, as it is shown in the Figure 5.1, to :

Choose a proof standard. One of the problems encountered during an evaluation process is the choice of the proof standard for solving the decision problem. Indeed, the choice of a proof is not obvious because it must ensure that the chosen procedure will allow to represent the decision maker's preferences. The problem, among others, is that such preferences evolve and change over time and it is difficult to predict them. Our idea is to construct a mechanism that instead of trying to identify accurately the decision maker's preferences to choose the procedure, it starts with an initial set of information and made a first arbitrary choice. Such a choice may change during the dialogue, according to the reactions and responses of the decision maker (or user). Technically, such mechanism will be based on the set of properties identified in the Chapter 4. Indeed, for each procedure, we identified a combination of properties that allow us to distinguish it from other procedures. Such properties will help to guide the choice of a procedure. The idea is that when the system makes a choice, it assumes, at the same time, that its corresponding properties are satisfied. Now, to confirm that this choice is the appropriate one will depend on the responses and reactions of the user during the interaction. In fact, some user's responses can challenge indirectly the choice of the system, by attacking one of the properties of the considered proof. Thus, the system should make a new choice based on the new combination of satisfied properties. To enable the system to make such choices or changes, we will represent under the form of a lattice the different possibilities that can have the system. In other terms, a node of the lattice will correspond to the pair (proof, corresponding properties) and links to various options to replace this proof. Such links are build on the fact that a new property is satisfied or that a current property is invalidated.

Generate explanations. At each step of the evaluation process, the decision maker can ask for explanations to clarify the reasoning steps assumed by the system, as well as the constructed recommendations and solutions. Thus, the system constructs explanations by instantiating argument schemes presented in the previous Chapter. Crucially, by presenting its justifications in the form of arguments, the system will make it 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, the system provides the different steps of reasoning by revealing parsimoniously the lower level schemes that compose the recommendation. In particular, the system will seek to construct the most “simple” and “natural” explanation for a given decision. The idea is that among all possible arguments that can be constructed to justify a decision, the system should select only the ones that are sufficient to construct a minimal explanation. The notion of minimality, as we shall see in this chapter, will depend on the language that will be used to construct the explanation.

Handle (revisions and updates). Another important task for the system is to take into account the various changes and updates occurring during the dialogue. Such changes are mainly due to the fact that the information available at the beginning of the process changes with the intervention of the decision maker. There are very different reasons to revise in such a process: in some cases the user may simply want to correct/refine one of its previous statements, introduce new information. In other cases it will contradict one of the system’s assumption. To support such changes the system will use different tools of argumentation theory that allow, on the one hand to evaluate the exchanged arguments and on the other hand to represent such changes under the form of a graph. The evaluation intends to maintain the information related to the decision problem updated. In fact, by presenting its justifications under the form of arguments, the system will make it possible for the user to pinpoint those steps that pose problems. To do that, the user will use the critical questions attached to each argument schemes or counter arguments. Thus, the system should check whether the conclusion is warranted or not by computing the acceptability of the arguments at each stage of the dialogue. On the other hand, the graphical representation offers a way to inform the user of the consequences of his changes. In fact, from a practical point of view, the graphical visualization of arguments allows to facilitate understanding of an argument’s structure and to help to see “points of attack” in criticizing an argument. Thus, we propose that the system construct a graph of argument during its interaction with the user.

Manage the dialogue. Finally, our system should be able to structure the dialogue on a formal basis in order to be able to control the interaction with the user. To account for that, we propose

to construct a dialogue game, that specifies the rules and conditions under which we can have a coherent dialogue in a decision aiding context. More precisely the dialogue is based on recent extensions that incorporate argument schemes within dialogues. [Reed and Walton, 2007]. Such a dialogue will allow the system to manage and guide the interaction with the user.

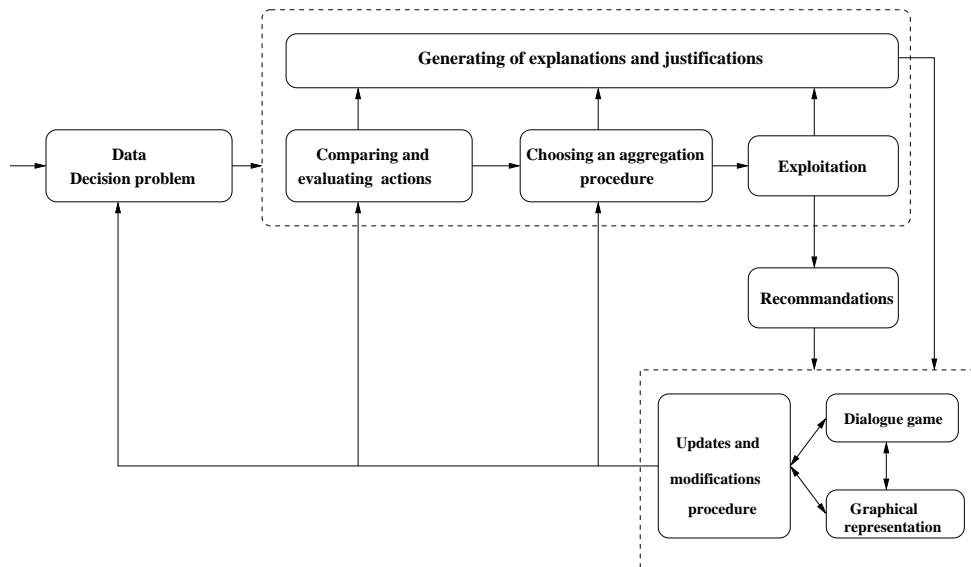


Figure 5.1 — System's functionalities

The chapter is organized as follows: In Section 5.2, after a brief introduction to Formal Concept Analysis, we present the Concept Lattice that will be used by the system as a mean to navigate among the different proof standards. Such a lattice is constructed by taking into account the fact that the properties describing a given proof can be satisfied or contradicted. Section 5.3 presents different kinds of formal minimal explanations for Condorcet winners, depending on the language used to construct such explanations. In Section 5.4, we present different ways allowing the decision maker to challenge and discuss the solutions and recommendations provided by the system during the interaction, namely: critical questions, counter arguments and critical responses. Section 5.5 is devoted to present a new version of the acceptability function inspired from the Carneads model. Finally, Section 5.6 presents the different rules to have a coherent interaction between the system and the user.

5.2 Choosing a proof standard

We have seen in Chapter 4 that it is possible to consider proof standards as aggregation procedures, and vice versa. However, unlike a proof, an aggregation procedure is not determined by

the context but depends on the preferential information provided by the decision maker during the process. Thus, we have to make a choice among the set of the aggregation procedures. In this section, we propose to address this problem, by exploiting the set of properties presented in Chapter 4. The aim is to provide a solution that allows to guide this choice. Let us start by the following example:

Example 5.1. *Suppose that a decision maker specifies that he has to choose between three alternatives $A = \{a, b, c\}$. Each alternative is evaluated on the set of criteria $H = \{h_0, h_1, h_2, h_3, h_4\}$ and that the evaluations are summarized in the performance table below.*

	h_0	h_1	h_2	h_3	h_4
a	7	6	2	3	5
b	6	4	8	4	7
c	3	2	5	2	3

On the basis of these information, the system calculates a recommendation and submits it to the decision maker for discussion. Such discussion is formalized through a dialogue game and allow the decision maker, for instance, to: ask for justifications, to add new information and/or to correct old data, etc. In this section, we focus on the discussion of the choice of the aggregation procedure and details of the construction of the dialogue are addressed in Section 5.6. Thus, we can have the following dialogue:

-
- 1. System:** I recommend b as being the best choice.
 - 2. User:** Why is that the case
 - 3. System:** b is better than any other actions on the set of criteria $\{h_3, h_4, h_5\}$.
 - 4. User:** mmm!! but for me a is the best!
 - 5. System:** why?
 - 6. User:** because it is better than any one else on the second criterion h_1
 - 7. ...** ...
-

Let us briefly analyse the dialogue. In the turn (1), the system suggests that b is the best action. It justifies that recommendation, in the turn (3) by the fact that b is better than any other action on the set of criteria $\{h_3, h_4, h_5\}$. We will see later, in Section 5.3, that such an explanation is generated automatically on the basis of the argument schemes presented in the previous chapter. Moreover, to build such a recommendation, the system should make a choice among the set of proof standards (or aggregation procedures). It was mentioned previously in this work, there exist several types of aggregation procedure and each one makes assumptions that are different from the others. In this example, it seems that the system selects the majority principle. However, the decision maker realizes that the recommendation is not the one he expected, because the system

did not take into account the fact that the criterion h_1 is sufficient, for him, to identify the best action.

Now, if we continue the dialogue, a possible move is that the system presents a new recommendation by taking into account the information provided by the user. Intuitively, the fact that the user suggests to use only one criterion to have a solution, he mentions indirectly that he follows a different reasoning to compare the actions. Therefore, the user challenges indirectly the current proof used by the system, which is the simple majority. However, according to the new information the majority is no longer warranted. The problem is that the simple majority does not assume that the criteria do not have the same importance when comparing actions (see Table 4.11). Technically, the use of the simple majority assumes that all criteria are of equally important. Therefore the procedure has the property of *anonymity* (see Table 4.15). Thus, once the decision maker has specified that the first criterion has a different importance, such property is challenged and the system should jump to a new aggregation function. The one that does not assume such a property.

Through the example above, we can observe that the choice of an aggregation procedure is not an obvious task, even if the data are very simple. The difficulty of choosing a procedure is mainly due to the fact that at the beginning of the process, it is not easy and obvious to have all the preferential information of the decision maker. The decision maker's preferences may change and evolve during the dialogue, so he can change his mind, correct a previous proposal or add new information. Thus, in some cases, according to the responses of the decision maker, the system may decide whether a procedure is appropriate or not. In other terms, we can suppose that, at the beginning of the process, the choice of a procedure may not be definitive because the procedure itself is subject to challenge during the process and depends on the available information during that process. Thus, the system should be able to adapt or to update the procedure on the basis of the information provided by the user.

To account for that, we will assume a set $R = \{r_0, r_1, \dots, r_n\}$ of potential proofs to be used by the system. The system does also import some initial preferential information I_0 provided by the user. More generally, at each time step of the dialogue t we denote by I_t the available preferential information and by R_t the methods that are compatible with I_t , that is, methods that are still eligible to be used by the system. Observe that R_0 does not necessarily equate to R , as the initial preferential information may already rule out some methods. After that, from the set of compatible methods, the system should select one method as being its favoured method, the one it will base its current reasoning on. More generally, one may wish to rank the different methods. An idea is to advance methods that are *simple for the user to understand*. When the user provides more preferential information (during the interaction), the system should be able to adapt and jump to the new favoured method. To do that, the system will be based on the *set of properties* identified in Chapter 4.

In the previous chapter we proposed to describe each procedure by a set of properties. Such properties will allow us to *justify and to guide the choice of an aggregation procedure* during the dialogue. In other terms, they will provide the sufficient support to use or to choose a specific procedure. Very broadly speaking, when the system makes choice of a first procedure, at the beginning of the process, the set of properties corresponding to this procedure is assumed to be satisfied or valid. But this validity is temporary and may change at any time of the dialogue. Indeed, if in the course of the interaction, the user provides information that seems to contradict one of the properties of the current procedure, the system should invalidate such property and update the current procedure by the one corresponding to the new combination of properties.

Thus, we need a way to guide the process to navigate among the different candidate proof standards, depending on the properties that are currently satisfied or contradicted. To account for that, we propose to formalise the relationship between the set of properties and the set of proofs by a *Concept Lattice*, adopted from the *Formal Concept Analysis*. Very broadly speaking, a node in the lattice will correspond to a particular proof and the link between the nodes will correspond to the fact that a property is satisfied or contradicted. Before giving the details of the construction of such a lattice, we present in what follows a brief overview of the Formal Concept Analysis.

5.2.1 Formal Concept Analysis (FCA)

Formal Concept Analysis (FCA) has been introduced by [Wille, 1982] and applied in many quite different fields like psychology, computer sciences, mathematics, industrial engineering, etc. According to [Ganter and Willer, 1999]:

“The method is mainly used for the *analysis* of data i.e. for investigating and processing explicitly given information. Such data will be structured into units which are formal abstractions of *concepts* of human thought allowing meaningful and comprehensible interpretation.”

The technique of Formal Concept Analysis is fairly simple. Starting from a (potentially large) set of objects and properties of those objects, FCA determines maximal groups (or clusters) of objects and properties, where:

- ▷ an object group is the set of all objects that share a common subset of attributes, and
- ▷ a property group is the set of all attributes shared by one of the object groups.

Definition 5.1. (*Formal context*).

A triple (O, A, I) is called a formal context if O and A are sets and $I \subseteq O \times A$ is a binary relation between O and A . The element of O are called *Objects*, those of A *attributes*, and I the *incidence of the context* (O, A, I) .

A context may be described as a table, with the objects corresponding to the rows of the table, the attributes corresponding to the columns of the table, and a boolean value corresponding to the fact that an object has or not a property (see example 5.2)

Example 5.2. *The following table represents the formal context for objects consisting of the integers from 1 to 10, and attributes composite (c), square (s), even (e), odd (o) and prime (p).*

	composite	even (e)	odd (e)	prime (p)	square (s)
1	0	0	1	0	1
2	0	1	0	1	0
3	0	0	1	1	0
4	1	1	0	0	1
5	0	0	1	1	0
6	1	1	0	0	0
7	0	0	1	1	0
8	1	1	0	0	0
9	1	0	1	0	1
10	1	0	0	0	0

Table 5.1 — Example of a formal context

Definition 5.2. *(Formal concept)*

A formal concept for a context is defined to be a pair (O_i, A_i) such that:

- ▷ $O_i \subseteq O$;
- ▷ $A_i \subseteq A$;
- ▷ every object in O_i has every attribute in A_i ;
- ▷ for every object in O that is not in O_i , there is an attribute in A_i that the object does not have,
- ▷ for every attribute in A that is not in A_i , there is an object in O_i that does not have that attribute.

Example 5.3. *An example of a concept is $(\{3,5,7\}, \{\text{odd}, \text{prime}\})$ which corresponds to the odd prime numbers.*

The concepts (O_i, A_i) defined above can be partially ordered by inclusion: if (O_i, A_i) and (O_j, A_j) are concepts, a partial order \leq is defined by saying that $(O_i, A_i) \leq (O_j, A_j)$ whenever $O_i \subseteq O_j$. Equivalently, $(O_i, A_i) \leq (O_j, A_j)$ whenever $A_j \subseteq A_i$. Such a relation is called, a specialization

among concepts. It is a complete lattice, called *Concept lattice*. The lattice can be illustrated by a Hass diagram such that the nodes are the concepts and the edges represent the links of specialization / generalization. The Figure 5.2 illustrates the Concept lattice corresponding to the formal context of the Table 5.1.

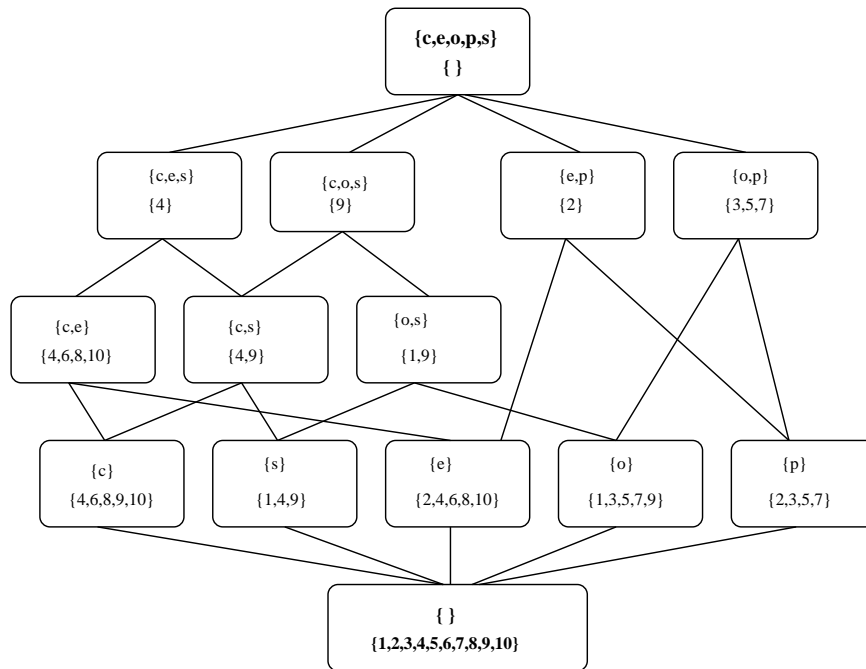


Figure 5.2 — An example of a concept lattice for objects consisting of integer numbers and attributes

5.2.2 FCA for choosing a proof standard

In what follows, we propose to use the concepts of FCA to build a lattice of concepts such that the concepts in our context would correspond to the couple: procedure and corresponding combination of properties. The idea is that this lattice will be the means that will be used by the system to guide its choice of a procedure during the dialogue: jumping from a procedure to another, represented by the links between the nodes in the lattice, will match the fact that a property has been satisfied or contradicted.

Thus, on the basis of the Table 4.15 which summarizes all proof considered in this work and their properties, we can construct the formal context (O,A,I) where: O is the set of proof standards, such that $O = \{\text{simple majority, lexicographical method, weighted majority, weighted sum, non additive model, simple majority+veto, weighted majority+veto}\}$; A is the set of properties, such that $A = \{\text{ordinality (ord), anonymity (ano), additivity w.r.t coalition (add-c), additivity w.r.t val-}$

ues (add-v), veto (v), dependence between criteria (dep)} and I the relation between O and A .

	Procedures	Ord	Ano	Add-c	Add	v	Dep
1	Simple majority	1	1	1	0	0	0
2	Lexicographic method	1	0	1	0	0	0
3	Weighted majority						
4	Weighted majority+ veto	1	0	1	0	1	0
5	Simple majority + veto	1	1	1	0	1	0
6	Weighted sum	0	0	0	1	0	0
7	Non additif model	0	0	0	0	0	1

Table 5.2 — A formal context for objects consisting of proof standards and attributes consisting of properties describing them.

Thus, for instance, the example 5.4 illustrates a concept in our context.

Example 5.4. A formal concept is $(\{majority\}, \{ord, ano, add-c\})$.

We can also use the Hass diagram to construct the concept lattice corresponding to the table 5.2. The lattice is depicted in the Figure 5.3. We note that for sake of simplicity we have replaced the name of the proof by the number of its row in the table.

Such a lattice offers a classification of different proofs, according to the specialization/generalisation relation. However, the lattice cannot be directly used by the system for choice procedure because the links do not provide information on how to move from one procedure to another during the interaction. Thus, we propose to reconstruct the lattice, such that the links between nodes will correspond to the fact that a property has been satisfied or contradicted, depending on the responses of the decision maker during the dialogue. Such lattice is depicted in the Figure 5.4.

The idea is that the system, always start, with the most “simple” procedure, depending on the available information, by assuming that its corresponding properties are satisfied. Such a choice is a default choice and not definitive, based on the assumption that the properties describing the procedure are consistent with the decision maker’s preferences. From our point of view, the notion of simplicity refers to the fact that the procedure requires the fewest possible assumptions on the data.

For instance, in the Example 5.1, the set of procedures R corresponds to the set of objects O and the set I_0 corresponds to the performance table constructed on the basis of the information provided by the user. Thus, on the basis of I_0 , the simplest and most intuitive procedure that the system should select is $r_0 = \{\text{the simple majority principle}\}$, which corresponds in the figure 5.4

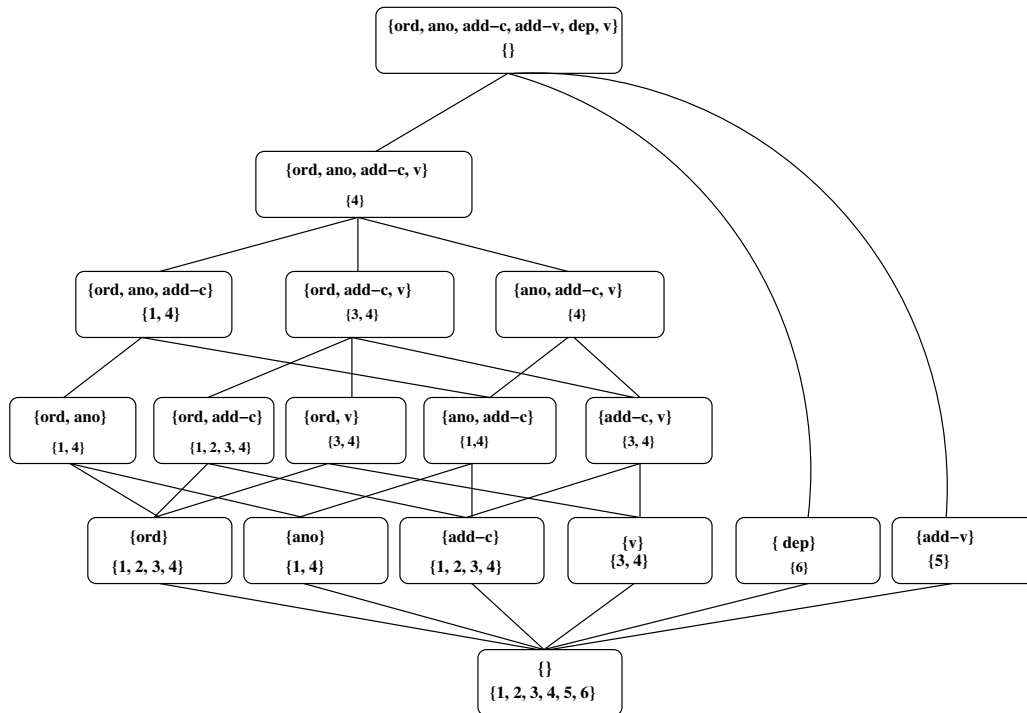


Figure 5.3 — A concept lattice for objects consisting of proof standard and attributes consisting of properties.

to the node (simple majority, 111000). In fact, with such data, the system does not need to add any further information to use the procedure r_0 . It has just to compare each action to the other and retain the one that beats all the actions on the majority of criteria.

At the same time, one can suppose that the weighted sum (with equal weights for all criteria), for instance, is also a possible method. Indeed, we have mentioned previously that facing a decision problem the simplest methods for a decision maker are majority principle and the weighted sum. However, the system can not base its choice on the latter one, because it has no *certainty* that the user's preferences are really not ordinal. In other terms, it is not obvious to deduce, explicitly, from the set I_0 that the decision maker takes into account the compensation among performances. Thus, to use the weighted sum, the system should check whether or not the property of ordinality is satisfied, by asking for instance further questions. The problem is that probably we should construct a large number of questions and even if we can get answers from the user, the conclusion may not be reliable because such answers can be interpreted in different ways.

Therefore, we chose to take a less ambitious way and relatively more simple, by considering the properties by default true, and making changes during the dialogue if it is necessary. Such changes will depend on the feedback from the user, when he is confronted by the recommenda-

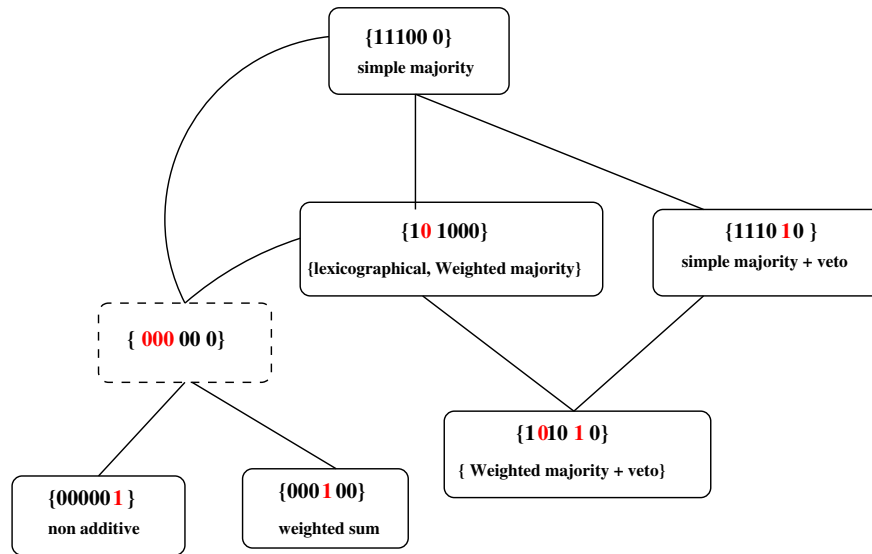


Figure 5.4 — Concept lattice to guide the choice of a proof standard

tion proposed by the system. If we take again the Example 5.1, the new information added by the user indicate that the recommendation should be calculated by taking into account exclusively the first criterion, and therefore that the current procedure is no longer warranted

This information can be interpreted as the existence of important differences between the different criteria considered in the problem, and that the first is the most important one. This interpretation contradicts the property of anonymity which is assumed by the majority method. The result is that the system should, on the one hand update the set of preferential information: $I_1 = \{\text{performance table}, h_1 \text{ is the most importante}\}$, and on the other hand consider the new combination of properties (101000). Thus, according to the lattice (see figure 5.4) the procedure that should be taken into account by the system can be either the lexicographic method or the weighted majority. But the choice of the system will be the lexicographic method because it is the most simple given the available data. Indeed, in order to use the weighted majority, the system needs the weight of each criterion, which are not given explicitly.

To sum up , for a set of initial information corresponding to a performance table (see Hypothesis 1.3), the system will consider the simple majority as the most simplest procedure among the set R , by assuming that its properties are satisfied. The update of the procedure during the dialogue will depend on the responses of the user. Such responses will help to guide the choice of the new procedure, by challenging the properties of the current procedure.

It is clear that in practice it is difficult to exhaustively identify or clearly interpret all possible answers of a decision maker in a dialogue, and this is not the purpose of this work. Our aim was to

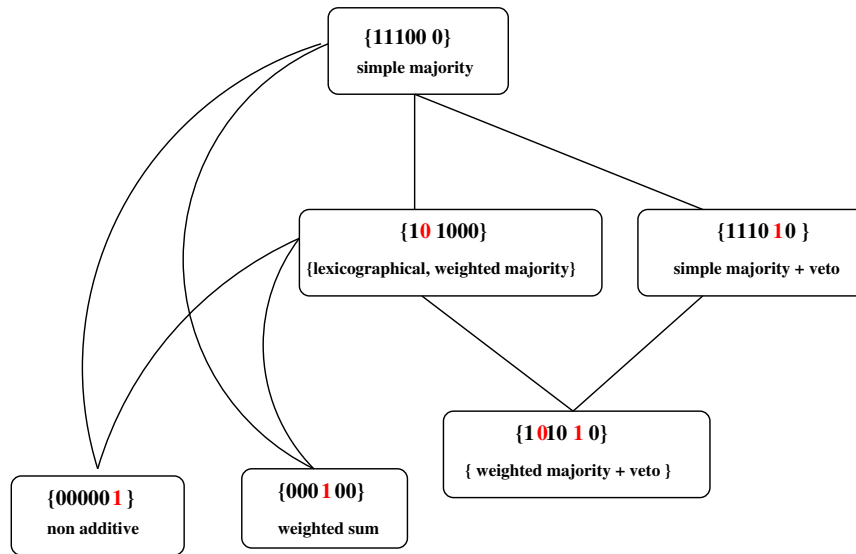


Figure 5.5 — Concept lattice to guide the choice of a proof standard (adapted)

provide a possible solution to the problem of choosing an aggregation procedure, which is based on concepts that are easy to use in practice. In general, starting from a simple majority method, if the user's responses refer to coalitions of criteria, order on criteria, etc. it is the property of anonymity that is challenged. However if the answers refer to a difference in performance between actions we can have two ways to interpret that. The first one is that such difference represents a penalty that can not be compensated. In other terms, it represents a strong negative reason against a given conclusion. Thus, in this case we are in presence of a kind of veto and the corresponding combination is (111010). We note that theoretically, having a veto contradicts the facts that the preferences are ordinal. However, in our context we distinguished two sides in the aggregation: a positive side and a negative one. Therefore, we can consider that we are ordinal in the positive side and having a veto on the negative one, which are two different situations but occur simultaneously.

In the second case, such a difference is interpreted as the existence of a compensation, and therefore, the ordinality is challenged. In this case, as it is shown in the figure 5.4, the new combination is (000000). Theoretically, challenging ordinality implies also challenging at the same time anonymity and additivity (in terms of coalitions). Thus, we move either towards the additive models (000100) or to non additive model (000001). But, as the node (000000) does not correspond to any particular procedure, we simplify the lattice as it is shown in the Figure 5.5.

To conclude, we presented in this section a first approach to guide to choice of a proof standard on the basis of a set of properties that describe it. Such properties were identified in the aim to provide a support to justify the use of an aggregation during the dialogue. Our approach is based

on the representation of procedures and their properties by a lattice of concepts inspired by the Formal Concept Analysis.

5.3 Generating arguments

We presented in Chapter 2 three approaches that addressed the problem of constructing explanations. These different approaches have stressed the importance of having an explanation that is: comprehensive, simple, intuitive and natural for a decision maker.

In this work, we propose to use the argument schemes presented in the previous chapter to construct explanations for the final recommendation of a decision problem. More precisely, for the best choice. Indeed, the construction of a recommendation depends on the problem statement. In our context, we assumed that the decision problem is a choice problem statement (see hypothesis 1.1 in Chapter 1), and therefore the recommendation corresponds to the determination of a choice set, noted $C(S)$. We assume, here, that such set exists and contains only a single element. There are several questions that the user may ask when confronted with a recommendation regarding such a choice, for instance:

(q_1) Why is this alternative x recommended?

(q_2) Why is this alternatives x *not* recommended?

or more complex requests involving the specific comparison with others alternatives:

(q'_1) Why is x recommended whereas this other one (y) is not?

(q'_2) Why is x *not* recommended whereas this other one (y) is?

Regarding the two last cases, the approach proposed by [Labreuche, 2005] (see subsection 2.2.5) allows fully to answer these two questions. In what follows, our ambition is to provide explanations for a more general case than comparing two options. It is clear that it is possible to construct an explanation by making explicit all pairwise comparisons of the winner with the other actions. However, it is possible to construct explanations that are more concise and *natural*.

Our system will seek to provide the most “simple”, “natural”, explanation for a given recommendation. This notion is difficult to grasp. Intuitively, the system will provide the *minimal evidence which allows to support the conclusion*. The idea is that among all possible arguments that can be constructed to justify a recommendation, the system should select a subset of arguments that are sufficient to construct such an explanation. The notion of minimality in our context, as we

shall see, will rely on the notion of “cost” of the explanation. Thus, the system will select arguments that make it possible to construct the least costly explanation. To account for that, we will distinguish two cases (remember we are in a choice problem statement case):

- ▷ there exists a *Condorcet winner*, which means that there exists an action $a \in A$ such that for any b different from a , a strict majority of criteria prefers a to b . In this case, letting $C(S) = a$ seems to be the only reasonable choice. In fact, by construction:
 - when there is a Condorcet winner, it is necessarily unique,
 - there is direct evidence that a is better to all other alternatives.
- ▷ there does not exist a Condorcet winner, but it is possible to find an action $a \in A$ that is better than all the others by using a specific exploitation technique (see Chapter 1). In this case, we give the intuition on the construction of the explanation by using the Copeland rule.

Then, we will discuss what kind of explanations the system can provide to the user depending if an action $a \in A$ is a Condorcet winner or not.

5.3.1 Computing minimal explanation for Condorcet-winner

Consider, for instance, that a decision maker has a choice problem among a set of alternatives $A = \{a, b, c, d\}$. Each alternative is evaluated on the following set of criteria $H = \{h_1, h_2, h_3, h_4\}$ and the results are summarized in the performance table of the Example 5.5.

Example 5.5. (*Performance Table*)

	h_1	h_2	h_3	h_4	h_5
a	6	4	8	4	7
b	7	6	2	3	5
c	3	2	5	2	3
d	7	7	2	0	2

Thereafter, on the basis of such information, the system computes the recommendation and presents it for discussion to the decision maker, as it is shown in the following dialogue:

Example 5.6. (*Example of dialogue*)

-
- 1. System:** I recommend a as being the best choice.
 - 2. User:** Why is that the case
 - 3. System:** a is globally better than all other options.
 - 4. User:** What does that mean?
 - 5. System:** because comparing a to any other options there is always a majority in favour of a
-

Let us briefly analyze this dialogue. In turn (1), the system suggests that the action a is the best one, without given further information. In turn (2), the user challenges this proposition and asks for a justification, which is given by the system in the turn (3). The justification explains that a is globally better than any other action (technically, this is a Condorcet Winner). Not satisfied with this explanation, the user asks for more clarification. In turn (5) the system explains that a beats all the other on the majority of criteria. Now, how to formalise such an explanation? Thus, in what follows, we discuss what kind of explanation can be provided by the system. First, we define an explanation as follows:

Definition 5.3. (*An explanation*)

*An explanation is a set of **statements** such that from this set it is possible to infer that a conclusion holds.*

In other terms, an explanation describes explicitly the reasoning steps to reach a conclusion. Here, a conclusion represents the fact that a certain action $a \in A$ is a Condorcet winner. Now, for the notion of minimality, we first refer to a natural definition. A minimal explanation, noted \subseteq –*minimal explanation*, represents a set of statements S , such that S contains all the *necessary* statements representing the comparison of the winner to the other alternatives of A . By necessary we mean that removing a statement makes the set not explanatory. This is a minimal requirement for an explanation, it ensures that it does not include any “useless” statements. Formally,

Definition 5.4. (\subseteq –*minimal explanation*)

A minimal explanation is a set $S = \{s_0, \dots, s_i\}$ of statements such that S explains the conclusion c and that $\forall s_i, S \setminus s_i$ does not explain c .

To make things more concrete let us now assume that statements are very basic, in the sense that are based on the structure of argument scheme for UNICRITERIA PAIRWISE EVALUATION (UC-PW-EV) (see Table 4.1). Formally,

Definition 5.5. (*Simple statements*)

Let $h_i \in H$ and $a \in A$ the Condorcet winner. A simple statement is of the form of $[h_i : a \succ x]$, such that $x \in A$, meaning:

“according to h_i , a is better than x ”.

Thus, an example of an explanation based on simple statement is illustrated through the Example 5.7.

Example 5.7. *If we consider the performance table of the example 5.5, we can construct the following ranking:*

$$h_0 : a \succ_0 b \succ_0 c \succ_0 d$$

$$h_1 : a \succ_1 b \succ_1 c \succ_1 d$$

$$h_2 : b \succ_2 c \succ_2 d$$

$$h_3 : c \succ_3 d \succ_3 a \succ_3 b$$

$$h_4 : a \succ_4 b \succ_4 c \succ_4 d$$

In this example the Condorcet winner is a and a possible explanation for that would be:

...
5. System: because, $[h_0 : a \succ b]$, $[h_1 : a \succ b]$, $[h_3 : a \succ b]$;
 $[h_0 : a \succ c]$, $[h_1 : a \succ c]$, $[h_2 : a \succ c]$;
 $[h_0 : a \succ d]$, $[h_1 : a \succ d]$ and $[h_2 : a \succ d]$

This example clearly shows that a more elaborated notion of minimality is required. Indeed, it would seem very natural here to simply state that a is top on three criteria. This means that:

- ▷ we need to allow different types of statements in the explanation corresponding to different argument schemes;
- ▷ we need to compare the “cost” of these different statements (to be able to meaningfully compare the cost of explanation).

Intuitively, we assume that a statement involving many alternatives or criteria is more difficult to cognitively process by the user. Thus, we identify first a new set, called the “*set of referents*”.

Definition 5.6. (*Set of referents*)

Let A a set of actions, H the set of criteria. We note $Ref = A \cup H$ the set of “referents”.

We denote by $Ref(s)$ the referents that appear in a statement.

We now define the cost of a statement:

Definition 5.7. (*Cost of statements*) *The cost of a statement, noted C , is given by a function which assigns to a statement a real number. We assume this function to be a monotone function from sets of referents to the reals, i.e $cost : 2^{Ref} \rightarrow \mathbb{R}$*

Note the two assumptions made on the cost function here. First, we assume that criteria and alternative referents play the same role, *i.e.* that the functions takes a set of referents without distinguishing their types and return a real. It is of course conceivable to have functions that would differentiate the cost associated to criteria and alternatives. The second assumptions is that the function is monotonic, *i.e.* a statement involving a superset of the referents used in another statement cannot be less costly. This seems to be a quite natural assumption to do.

In the rest of this document, we shall use for the sake of simplicity the cost function simply defined as returning the number of referents, in other words $cost(s) = |Ref(s)|$. When our results apply more generally, we explicitly state it.

Equipped with this notion we simply define the *cost of an explanation* as the sum of the costs of the statements involved in such an explanation. Thus, an explanation is said to be minimal *w.r.t* a given language when it is not possible to construct with this language an explanation which is less costly. Let us now investigate the consequence of this definition using different languages.

Minimal explanation with simple statements. Let us get back to our example 5.7, then, to the simple statement $[h_1 : a \succ b]$ is associated a cost of 3. Therefore, the whole explanation has an associated cost of 18.

It is easy to observe the following property :

Property 5.1. *A minimal explanation (w.r.t simple statements) is equivalent to a \subseteq -minimal explanation.*

Proof :

We observe that the cost of a simple statement is constant (it is 3). This means that minimizing the cost of the explanation corresponds to minimizing the number of statements, hence there will be minimal number of statements, namely: $(m - 1) \times \lfloor n/2 + 1 \rfloor$, such that m is the number of alternatives and n the number of criteria.

■

Technically, computing such an explanation is trivial: we just need to scan the performance table, and put in the explanation any statement supporting the winner against each action, until it gets the required number of supportive statements. Of course, there are several of such minimal explanation, and we would like to be more specific and concise. If we consider again the example 5.7, we observe that w.r.t simple statements we used the set $\{h_0, h_1, h_2, h_3\}$ to construct the explanation. However, we can use only three criteria if we were allowed to “factor” the statements and to group them regarding the same criteria.

Minimal explanation with “factored” statements. In this case, we will allow to “factor” the statements and thus to construct statements on the basis of argument scheme for UNICRITERIA GOOD RANKING (see Table 4.9). Of course the language obtained is strictly more expressive than the previous one (a simple statement being a special case of the factored statements).

Definition 5.8. (*Factored statement*)

Let $h_i \in H$, $a \in A$ the Condorcet winner. A factored statement is of the form $[h_i : a \succeq \{x, y, \dots\}]$, such that $x, y \in A$, meaning:

“on the criterion h_i , a is better than x, y, \dots ”

Example 5.8.

5. System: because, $[h_0 : a \succ \{b, c, d\}]; [h_1 : a \succ \{b, c, d\}]; [h_4 : a \succ \{b, c, d\}]$

In this case, the explanation has an associated cost of 15.

Property 5.2. A minimal explanation (w.r.t factored statements) implies minimizing the number of criteria.

Proof :

Observe that overall, the explanation will contain the minimal number of alternative referents needed (namely $(m - 1) \times \lfloor n/2 + 1 \rfloor$). This means that the only part of the explanation which makes a difference of cost is the number of criteria referents. Clearly, an explanation not minimizing the number of criteria cannot be minimal.

■

We draw attention of the reader to the fact that we can not take a minimal explanation w.r.t simple statements and turn it into a minimal explanation w.r.t factored statement. In fact, to compute such an explanation the system will consider coalition of criteria and check whether they are sufficient or not, in the sense of minimality. Thus, to account for that, the system will first build a binary matrix $M_{(m-1, n)}$, such that:

$$m_{(i,j)} = \begin{cases} 1 & \text{if } aPx, \\ 0 & \text{else.} \end{cases}$$

where a is the Condorcet winner and $x \in A$. The example 5.9 represents the binary matrix for the example 5.7.

Example 5.9. (Binary matrix for the Example 5.7)

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

After that, the explanation will include a minimal coalition of criteria such that there are at least $(n/2) + 1$ “one” in each line of the matrix (i.e. the winner beats each action at least by a majority of criteria). To find such a coalition, we the system can use the Algorithm 5.1. This algorithm is of course exponential in the number of criteria, but this wouldn’t be a problem as this number would typically not be high.

Algorithm 5.1 — Minimal explanation (Factored statements)

Input : a : a Condorcet winner; n : number of criteria;

Output : a minimal coalition of criteria;

Data : s : size of a coalition; L : a list to record the sufficient coalitions;

```

1 for  $s = (n/2) + 1$  to  $n$  do
2   For all possible coalitions of criteria of size  $s$ ,
3   check whether they are sufficient.
4   if yes then
5     |  $L \leftarrow L \cup \{\text{coalition}\}$ 
6   end
7 end
8 return a minimal coalition

```

If we get back to the example 5.8, we can note that it seems more natural to explain that a is the best action by stating that it is ranked first on the majority of criteria $\{h_0, h_1, h_4\}$. Moreover, if we apply algorithm 5.1 to example 5.10, the output will be the set $\{h_0, h_1, h_2, h_3\}$ which is sufficient in this case. But, we can observe that h_4 is not used in the explanation whereas a is ranked first on it. This seems contrary to our intuition.

Example 5.10.

$h_0 : a \succ_0 b \succ_0 c \succ_0 d$

$h_1 : d \succ_1 a \succ_1 b \succ_1 c$

$h_2 : b \succ_2 a \succ_2 c \succ_2 d$

$h_3 : c \succ_3 a \succ_3 b \succ_3 d$

$h_4 : a \succ_4 b \succ_4 c \succ_4 d$

Thus, we need to refine the language to take into account such requirements.

Minimal explanation with generic statements. The new language will allow us to define the new generic referent $*$ which stands for “all alternatives” and has a cost of 1 as any other referent. Thus, in this case, a statement is constructed on the basis of the argument scheme for the UNICRITERIA TOP RANKING (see Table 4.8). Moreover, we can use also “ $* \setminus \{x, y, \dots\}$ ”, which stands for “all alternatives except x, y, \dots ”.

Definition 5.9. (*Generic statement*)

Let h_i and $h_j \in H$, $a \in A$ the condorcet winner. A generic statement is of the form:

$[h_i : a \succeq *]$, meaning “ a is better than anyone else on h_i .” and

$[h_j : a \succeq * \setminus \{x, y, \dots\}]$ meaning “ a is better than every one except x, y, \dots on h_j .”

In the following we make the assumption that the generic statement “ a is better than anyone else on h_i .” is one with minimal cost. (This is the case here as we consider $*$ as having the same cost as any other referent).

With this language a possible explanation, for the example 5.10 is the one presented in the Example 5.11. It has an associated cost of 13, because we use the following statements: $[h_0 : a \succeq *]$ (cost=3); $[h_1 : a \succeq * \setminus \{d\}]$ (cost=4); $[h_4 : a \succeq *]$ (cost=3) and $[h_2 : a \succeq d]$ (cost=3).

Example 5.11.

...

5. System: because, a is better than every one except d on $\{h_0, h_1, h_4\}$
but a is better than d on h_2

In this case it is interesting to note that a minimal explanation will not always minimize the number of criteria involved in the explanation. In some cases, it may be interesting to use more criteria on which the alternative is ranked higher. More precisely, it can only be the case when more than 7 criteria and 6 alternatives are involved, as formally proved in the following proposition:

Property 5.3. *with generic statements, an explanation minimizing the cost does not imply minimizing the number of criteria involved in the explanation, unless there are less than 7 criteria or less than 6 alternatives, minimizing the cost of the explanation with generic statements involves minimizing the number of criteria.*

Proof : We seek to find the minimal number of criteria such that we can construct an example where an explication involving more criteria is less costly than another explanation involving less criteria. Consider the following rankings and let c be the Condorcet-winner:

- h_0 c is ranked first
- h_1 c is ranked first
- h_2 c is ranked first
- h_3 $x_1 \succ \dots \succ x_{m/2} \succ c \succ x_{m/2+1} \succ \dots \succ x_m$
- h_4 $x_{m/2+1} \succ \dots \succ x_m \succ c \succ x_1 \succ \dots \succ x_{m/2}$
- h_5 $x_1 \succ x_m \succ c \succ \dots$
- h_6 $\dots \succ c \succ x_1$

To explain that c is the best choice, we can construct two possible explanations, E_0 and E_1 , such that:

- $\triangleright E_0: [h_0 : c \succ *]; [h_1 : c \succ *]; [h_2 : c \succ *]; [h_3 : c \succ * \setminus \{x_1, \dots, x_{m/2}\}];$ and $[h_4 : c \succ \{x_1, \dots, x_{m/2}\}].$
- $\triangleright E_1: [h_0 : c \succ *]; [h_1 : c \succ *]; [h_2 : c \succ *]; [h_5 : c \succ * \setminus \{x_1, x_m\}]; [h_6 : c \succ x_1];$ and $[h_3 : c \succ x_m].$

The example requires two different candidate explanations to be compared, one (E_0) involving strictly less criteria than the other one (E_1). Thus, in terms of number of criteria (i) $|E_0| < |E_1|$, such that (ii) $cost(E_0) > cost(E_1)$. Observe that this last condition implies that (iii) $E_0 \not\subseteq E_1$, otherwise its cost would never be higher (due to the fact that the cost function is monotone). Finally, note that (iv) $|E_0| > \lfloor \frac{n}{2} + 1 \rfloor$. This is so because if the explanation E_0 involved exactly the number of criteria required to make a majority, this would imply that c should be ranked top on each of these criteria. But in this case the $cost(E_0)$ would clearly be minimal (due to the assumption that a “ c is top” statement is one with minimal cost).

Now, the minimal set of criteria meeting the different conditions is $n = 7$ elements, where one coalition of size 5 and one coalition of size 6 can be extracted. (Condition (iii) alone would require at least 5 elements). Now from this number of criteria it is easy that the number minimal of alternative is 6. In fact, the explanation E_0 can only become more costly when at least $Cost(E_0) = 14 + m > Cost(E_1)20$, i.e., $m > 6$. The case constituted of h_3 and h_4 is clearly a worst-case situation (where c lies right in the middle of both orders), hence maximizing the cost when using the criteria in the explanation.

■

Before we get further it is interesting to make some observations. First, note that in order to compute a minimal explanations w.r.t. generic statement, the greedy heuristic consisting in taking the criteria on which the winner c is ranked higher, and progressively going down, is not acceptable. To see this, observe that having included a statement “ c is better than everyone except a” in the explanation, it is not necessarily optimal to include a second similar statement that would

leave a uncovered. The only thing that can be guaranteed is that *all* (necessary) “ c is ranked top” statements must be included, as well as at least one criteria where the rank of c is maximal.

Here Proposition 5.3 can help in the sense that under the aforementioned assumptions, we can restrict our attention to explanations minimizing the number of criteria involved. We can then directly adapt the previous algorithm such that the cost of each minimal explanation is stored, and the algorithm would return the coalition minimizing the cost (and of course we can pre-process a bit the instance by using the previous remarks).

Now remains a question: is there a natural interpretation of what (beside the number of criteria) is being optimized when we use generic statements and fall under the assumptions of Prop. 5.3? The answer is not obvious. A notion that seems connected to the cost is the number of alternatives that are dominated overall by the winner c in a given coalition (for each criteria of the coalition, simply count the number of alternatives below c). This is equivalent to count the Borda score of c restricted to the coalition considered, or alternatively to the *average ranking* of c in this coalition. Contrary to what we may think at first sight, an explication with minimal cost is not necessarily an explication maximizing the average ranking among the minimal coalitions. This is so because there may be several minimal coalitions allowing a minimal cost explanation, and there is no reason to select necessarily the one maximizing the average ranking. Take for instance the following example:

Example 5.12.

$h_0 : c \succ \dots,$
 $h_1 : c \succ \dots,$
 $h_2 : b \succ c \succ \dots$
 $h_3 : a \succ c \succ \dots$
 $h_4 : \dots \succ c \succ b$

Here two explanations with the same minimal cost can be constructed

- $\triangleright E_0: [h_0 : c \succeq *], [h_1 : c \succeq *], [h_2 : c \succeq * \setminus \{b\}] \text{ and } [h_3 : c \succeq b];$
- $\triangleright E_1: [h_0 : c \succeq *], [h_1 : c \succeq *], [h_2 : c \succeq * \setminus \{b\}] \text{ and } [h_4 : c \succeq b]$

The first option may seem more desirable in the sense that c ranks higher on the last exhibited criteria.

The symmetric question may then naturally be asked: is it the case that from a minimal coalition of criteria where c maximizes its average ranking we can necessarily extract a minimal cost explanation. Were it to be the case we would have a further refinement of our concept of minimal explanation.

In our context and under the assumption that there are less than 7 criteria we show that this is the case.

Property 5.4. *When the number of criteria is less than 7, it is always possible to construct a minimal explanation from a coalition minimizing the number of criteria and maximizing the average ranking of the Condorcet winner.*

Proof : Assume that there exist two minimal coalitions E_1 and E_2 , and that we want to build an example such that from the one maximizing the average ranking of the Condorcet winner c (say E_1) we cannot extract a minimal cost explanation. The critical point to make is the coalitions differ necessarily on at least two criteria. Indeed suppose only one criteria differs: this means that c is ranked higher on a criteria h_1 (used in E_1) than on h_2 (used in E_2). But in that case the only alternatives required to be ranked below c in h_1 are those also ranked below c in criteria h_2 (because E_2 is indeed an explanation and because the two explanations coincide on all other criteria). But in this case the minimal explanation from E_2 could also be constructed from E_1 , hence contradicting the fact that no minimal cost explanation can be built from E_1 . We are almost done: now that we know that the coalitions differ on at least two criteria we know that there are already 4 criteria used in the example. Hence we need to x criteria to build an example, such that $x + 2 > \frac{x+4}{2} + 1$ (here again because the coalitions cannot just reach the majority otherwise c would be top on each criteria), i.e $x > 2$. Summing up with our 4 criteria this makes 7 criteria overall.

■

Using this property offers the additional advantage of providing an explanation with minimal cost where the average position of c is the highest possible, thus making it arguably “overall” more convincing. To compute such an explanation, we can adapt the exponential algorithm such that the number of zeros of each minimal coalition is stored and the algorithm would return the coalition minimizing the number of zeros. Such a number corresponds to the fact that on a given criterion, an action is not beaten. If we consider the matrix of the example 5.10 presented below, in the case w.r.t factored statements, the coalition that the system will present to the decision maker is $\{h_0, h_1, h_2, h_3\}$, where the number of zeros is three. However, w.r.t generic statements, the minimal coalition is $\{h_0, h_1, h_2, h_4\}$ and there are two zeros.

Example 5.13. *(Binary matrix for the example 5.10)*

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Considering the cost function and the property 5.4, the problem of computing the minimal explanation for an average ranking can be represented by an Integer Linear Programming (ILP) modelization. The ILP, \mathcal{P} , uses the following binary variables, such that: $h_i = 1$ if the criteria is involved in the explanation and 0 else; $x_{jk} = 1$ if c beats j according to a selected h_k and Ω is a big value. We also have the initial values of the binary matrix $m_{jk} = 1$ if c beats j according to h_k and 0 otherwise.

$$\mathcal{P} = \begin{cases} \min & \Omega \sum_{k=1}^n h_k - \sum_{k=1}^n \sum_{j=1}^m x_{jk} & (1) \\ \text{s.t.} & m_{jk} \geq x_{jk} & \forall j, \forall k \quad (2) \\ & h_k \geq x_{jk} & \forall j, \forall k \quad (3) \\ & \sum_{k=1}^n x_{jk} \geq n/2 + 1(\text{majority}) & \forall j \quad (4) \end{cases}$$

The objective function (1) makes sure that the number of criteria is first minimized ($\Omega \sum_{k=1}^n h_k$) and then that the number of “1s” is maximized. The condition (2) links the selected statements and their initial values in the matrix. The condition (3) links the variables in such a way that x_{jk} are only considered where the criteria belong to the coalition. Finally, the last constraint (4) checks that indeed c beats any other candidate with a strict majority in the selected coalition.

To sum up, we presented in this section the generation of minimal explanations for a best choice corresponding to a Condorcet winner. We have seen that an explanation is defined by a set of statements and has a specific cost depending on the language used to define a statement. The Table 5.3) summarizes the different cases:

Statements	Property	
<i>Simple</i>	\Leftrightarrow minimizing the number of statements	Property 5.1
<i>Factored</i>	\Rightarrow minimizing the number of criteria	Property 5.2
<i>Generic</i>	\Rightarrow minimizing the number of criteria when $n < 7$ or $m < 6$	Property 5.3
	\Leftarrow minimizing the number of criteria and maximizing the average ranking of the Condorcet winner when $n < 7$	Property 5.4

Table 5.3 — Properties of minimal cost explanations

Unfortunately, the existence of a Condorcet winner is not always obvious. If we take the Example 5.14, we can observe that no Condorcet winner exists.

Example 5.14.

$$\begin{aligned} h_0 : a \succ_0 e \succ_0 c \succ_0 d \succ_0 b \\ h_1 : b \succ_1 a \succ_1 e \succ_1 c \succ_1 d \\ h_2 : c \succ_2 d \succ_2 b \succ_2 a \succ_2 e \\ h_3 : e \succ_3 a \succ_3 d \succ_3 c \succ_3 b \\ h_4 : d \succ_4 b \succ_4 a \succ_4 c \succ_4 e \end{aligned}$$

We can observe in this example that there is no action who beats all other actions in pairwise comparisons. In this case, in order to provide a recommendation, the system should use a technique that allows to find a solution. A natural extension of the Condorcet winner is the *Copeland rule*. We propose in what follows, to give a glimpse of what can be an explanation under such a rule. We will not discuss in detail, as we have done previously, the minimality of the explanation or the opportunity to have different explanations.

5.3.2 Computing explanation for a non Condorcet winner

The Copeland rule provides an alternative way to find a winner when there is no Condorcet winner. In other terms, the winner is determined by finding the actions with the most pairwise victories or the action that defeats as many other action as possible, even by a very small margin (see example 5.15).

Example 5.15.

If we consider the rankings of the example 5.14, we can obtain the following table, which shows the number of wins and loses for each action.

	wins	except against
<i>a</i>	3	<i>b</i>
<i>b</i>	2	<i>c, d</i>
<i>c</i>	2	<i>a, e</i>
<i>d</i>	1	<i>a, b, e</i>
<i>e</i>	2	<i>a, b</i>

The action a has the greatest number of wins (3 out of 4) and is therefore the Copeland winner.

In what follows we propose to give an overview of what is an explanation in this case. Indeed, we will not discuss, as we done for the Condorcet winner, the minimality of the explanation.

Our aim is just to show that from one case to another, the necessary information to include in an explanation are different.

We note, that the the definition of an explanation (see Definition 5.3) remains the same in this case. However, we should should take into account, in the definition of the statement, the Copeland score of the action (see Definition 5.10).

Definition 5.10. (*Copeland statements*)

Let $a \in A$ a Copeland winner and s its Copeland score. A statement is of the form:

$[a_{(s)} \succeq * \setminus \{x\}] \dots$ meaning “with a Copeland score s , a is better than every one except $\{x\}$ ”

$[a_{(s)} \succeq * \setminus \{x, y, \dots\}] \dots$ meaning “with a Copeland score s , a is better than every one except $\{x, y, \dots\}$ ”

Thus, to explain a Copeland winner, the system should distinguish two cases: a winner with a Copeland score $s = m - 2$ (such that m is the number of actions) and a winner with a Copeland score $s \geq m - 3$. We note that a Copeland winner with a score equal to $m - 1$ is a Condorcet winner, therefore, the system will provide the explanation as it was described in the previous subsection.

The distinction between such two cases is essentially due to the content of the explanation. Indeed, in the first case, as it is illustrated by the example 5.16, to explain that an action is better than all other options except one, it is sufficient to provide evidence that the latter can not be better than the winner by putting forward its low score. In comparison with the Condorcet, we do not need to specify the criteria, it suffices to exhibit the global score.

Example 5.16. *System: with a score of 2, a is better than every one except b but b has a score of 0.*

However, in the second case, the things are a little bit different, in the sense that the explanation may not be concise. In fact, to explain that an action x is a Copeland winner that beats every one except two, we should not only show that these two action can not be better than the winner, by making explicit their scores, but we need also to show that it is not the case for the rest of the actions. In other words, we must show that every action that is not the winner, has a score smaller than the winner.

5.4 Generating counter-arguments

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 assumed by the system. The intervention of the decision maker can be considered in two ways either by *critical questions* associated to an argument scheme or *critical responses*.

We recall that critical questions represent attacks, challenges or criticisms that, if not answered adequately, falsify the argument fitting the scheme. Moreover, [Verheij, 2003] distinguished 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. For instance, a possible move in the Example 5.1, 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. 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 of his problem. Of course, the user should justify his proposition by providing an argument supporting that option. We find this type of this 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 considers, for some reasons, that one of the eliminated actions is better than the proposed choice. For instance, in the Example 5.1, confronted to the recommendation of the system, the user realizes that it is not the one he expected, because the system did not take into account the fact that the first criterion is sufficient to find a solution. The user expresses his rejection by indicating the alternative that he designated to be the best choice (Turn 4 in Example 5.1). More precisely, the response of the user will refer to particular critical question that allows to identify the counter argument. Such a questions corresponds to “Is there not a better alternative?”.

On the other hand, a question can be identified [Gordon *et al.*, 2007], depending on whether it refers to standard assumptions (**A**) of the scheme or to exceptional circumstances (**E**). This has in particular a significant difference on how the burden of proof is allocated. We now list some of the questions that can be attached to the different premises of our argument schemes. For instance for the *Argument Scheme for Multi-Criteria Pairwise Evaluation* the different type of questions is clear. The burden of proof lies on the proponent when it must provide supportive evidence (positive reasons) for the main claim. On the other hand, the opponent should be the one providing negative reasons to block the conclusion.

1. *actions* (A): is the action possible?
2. *list of criteria* (A): (i) Is this criteria relevant?, (ii) Should we introduce a new criteria?, (iii) Are these two criteria are in fact the same?
3. *supporting reasons* (A): (i) Are there enough supporting reasons to support the claim?

4. *opposing reasons* (E): Are there not enough reasons to block the claim?

The Table 5.4, illustrates different examples of critical questions that allows to attack the different argument schemes of our hierarchy. It should be noted that a negative answer to some of these questions leads to a conflict whose resolution requires sometimes the transition to a different stage of the decision aiding process. For instance, when you challenge whether the action is possible to start with, you are dealing with problem formulation, where the set of alternatives is defined. It is out of the scope of this thesis to discuss this problem. We will just mention 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.

If we get back to example 5.1, the response of the decision maker in Turn 6: “because it is better than any one else on h_1 ”, is certainly a counter argument against the previous proposal of the system, but it contains additional information that challenges the proof standard in use. We call such a response “*Critical response*”. We discussed in section 5.2 that it is possible to discuss the use of a proof standard during the interaction on the basis of the information provided by the user. Such information contradicts or satisfied new properties, which allows the system to navigate among the lattice and to choose a new proof. We draw the attention of the reader to the fact that the user is not aware about the fact that he challenged the proof. He only provides indirect information that is treated by the system as critical for the use of a given proof. In the Example 5.1, the critical response contradicts the property of “anonymity”. The Table 5.5 offers some examples of user’s responses, during the interaction, that helps to distinguish which property (of a given proof) has been satisfied or contradicted. It is clear that it is difficult to identify exhaustively or interpret clearly all possible responses of a decision maker in an interaction, and this is not the purpose of this work. Moreover, we distinguished for each property two types of responses, namely: **Positive Evidence (PE)** and **Negative Evidence (NE)**. The former allow to confirm or to reinforce th assumption that the property is satisfied. The latter, in contrary, indicates when the property is contradicted or not satisfied.

Finally, we note that is was difficult to find examples of critical responses for two cases. The first one concerns the positive evidence for the property of additivity w.r.t coalitions. The second one is the negative evidence against the existence of a veto. For the latter, in practice it seems not natural to try to contradicts a veto.

	Questions	A	E	Argument schemes
1	Is there not a better alternative?		×	MC-PW-EV, UC-Top, UC-GOOD, RECOMM-SCHEME
2	Is this criterion relevant?	×		UC-PW-EV, UC-IN-EV, UC-IN-ACC, UC-PW-ACC, UC-GOOD, UC-Top
3	Are these two criteria in fact the same?	×		MC-PW-EV, MAJO-SCHEME, LEXICO-SCHEME, WEIGHTEDM-SCHEME, WEIGHTEDM-SCHEME
4	Should we introduce a new criterion?	×		MC-PW-EV, MAJO-SCHEME, LEXICO-SCHEME, WEIGHTEDM-SCHEME, WEIGHTEDM-SCHEME
5	Is the action possible?	×		MC-PW-EV, UC-PW-EV, UC-IN-EV, UC-IN-ACC, UC-PW-ACC, UC-GOOD, UC-Top
6	Is the action acceptable?		×	MC-PW-EV, UC-PW-EV, UC-IN-EV, UC-IN-ACC, UC-PW-ACC, UC-GOOD, UC-Top
7	Is the performance correct?	×		UC-PW-EV, UC-IN-EV, UC-IN-ACC, UC-PW-ACC
8	Is the preference relation appropriate?	×		UC-PW-EV, UC-IN-EV, UC-IN-ACC, UC-PW-ACC
9	Are there enough positive reasons to support the claim?	×		MC-PW-EV
10	Are there not enough reasons to block the claim?		×	MC-PW-EV

Table 5.4 — Examples of Critical Questions

Property	Default value	Examples of critical responses: PE and NE
Ordinality	true	<p>Positive Evidence: the difference between x and y is not significant for me.</p> <p>Negative Evidence: the difference between x and y on the criterion h_i is too small, large, ...</p>
Anonymity	true	<p>Positive Evidence: you can also replace the criterion h_i by h_j, it is the same result</p> <p>Negative Evidence: the criterion h_i is more important than the criterion h_j x is better than y on the coalition of criteria $\{h_i, h_j\}$ if x is preferred to y on the criterion h_i it should be the same on the criterion h_j x is too bad (or better than any one else) on the criterion h_j</p>
Additivity w.r.t coalitions	true	<p>Positive Evidence: ?</p> <p>Negative Evidence: if x is preferred to y on the criterion h_i it should be the same on the criterion h_j I see why x is better than y on $\{h_i\}$ but I am not sure to understand why it remains better when we add h_j</p>
Additivity w.r.t values	false	<p>Positive Evidence: the difference between x and y on the criterion h_i is too small, large ...</p> <p>Negative Evidence: if x is preferred to y on the criterion h_i it should be the same on the criterion h_j</p>
Dependence between criteria	false	<p>Positive Evidence: if x is preferred to y on the criterion h_i it should be the same on the criterion h_j if x is preferred to y on h_i then w should be preferred to z on h_j</p> <p>Negative Evidence: the difference between x and y on the criterion h_i is too small</p>
Veto	false	<p>Positive Evidence: x is too bad</p> <p>Negative Evidence: comparing x to y on the criterion h_i, a is not satisfactory ?</p>

Table 5.5 — Examples of critical responses against the properties

5.5 Acceptability of arguments

Determining the acceptability of arguments built during an interaction, is an important step in an argumentation process. It will allow the system to check whether a recommendation is warranted or not on the basis of the different arguments and information exchanged during the dialogue. We have seen in Chapter 2 that acceptability of arguments can be determined, either by following the principles of Dung's framework [Dung, 1995], or by considering it, as in the case of Carneades model [Gordon *et al.*, 2007], where the acceptability relies on the determination of proposition acceptability.

In this work, we adopt the second approach. As it was mentioned earlier in this document, we will not assume the existence of a fixed knowledge base to construct arguments, which departs from the classical Dungian framework. Thus, argumentation in our context will be viewed as a process to produce justifications of the recommendations of a decision problem. Therefore, what is important is to determine whether recommendations are acceptable on the basis of arguments constructed in favour and against them. Moreover, the aim of the system, among others, is to inform the decision maker of the consequences of the changes done during the dialogue (e.g. adding a new information, revising a parameter, revising an aggregation procedure, etc). To do that, we map the dialogue between the system and a user during the interaction to a discussion graph with a tree-like structure. This graph will allow to *visualize and evaluate* the set of arguments exchanged during this dialogue. Thus, we seek to determine the proposition acceptability in a graph of arguments.

We adopt to our context the graph of the Carneades model presented in Chapter 2. We recall that, that argument graphs have two kinds of nodes, *statement nodes* and *argument nodes*. The statements are declarative sentences in some language. In our context the statements are constructed on the basis of preferential information provided by the user. The different statements considered in this work are presented in Section 5.6. On the other hand, the arguments nodes, in the Carneades model, are instantiations of argument schemes linking a set of premises to a conclusion. Formally,

Definition 5.11. (*Arguments*)[Gordon *et al.*, 2007]

An argument is a tuple $\langle c, d, p \rangle$, where c is a statement, $d \in \{pro, con\}$ and $p \in 2^{premise}$. If a is an argument $\langle c, d, p \rangle$, then $conclusion(a) = c$, $direction(a) = d$ and $premises(a) = p$.

We recall that the premises can be of three types: (i) ordinary premises, (ii) assumptions and (iii) exceptions. For our context, we adopt the same definition of an argument. More precisely, arguments are instances of the argument schemes identified in the hierarchy (see Figure 4.4).

Furthermore, argument graphs have different kind of edges that link up the premises and conclusions of the arguments. They represent the pro and con argument as well as the different kinds

of premises. For our graph, we keep all these links and we propose to add a new one. Indeed, we have proposed to represent explicitly the proof standard associated to each statement. To represent such an association we will use the following link.

Definition 5.12. (*“According To” link*)

An “According To” edge links a statement to its proof standard. This link translates the fact that arguments, built in favour and against a statement, must be aggregated following that proof standard.

Graphically, such a link is represented by a double line edge with no arrowhead (see Figure 5.6).

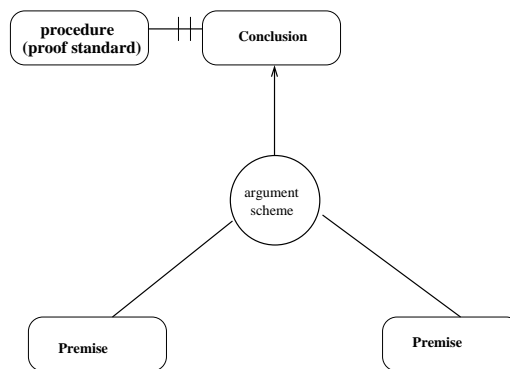


Figure 5.6 — Argument graph for a multiple criteria context

Now, the acceptability of a statement in the Carneades model, depends essentially on three elements: its *dialectical status* during the dialogue, its *proof standard* and its *premise type*. For our context, we follow the same idea but unlike the original version where the proof is accepted by default, in our context the proof standard can be discussed and challenged (see Section 5.2) which implies that it may have its own dialectical status during the dialogue and thus its own acceptability. Thus, first, we will consider the following *Argument Context*:

Definition 5.13. (*Argument Context*)

Let C_m , the argument context in a multiple criteria situation, be a tuple $\langle status, ps, status_{ps} \rangle$, where $status$ is a function of type $statement \rightarrow \{stated, questioned, accepted, rejected\}$, ps is a function of type $statement \rightarrow proof\text{-standard}$ and $status_{ps}$ is a function of type $proof \rightarrow \{accepted, rejected\}$.

We note that, unlike the Carneades model (see Definition 2.4), we will not assume, at the beginning, the existence of a partial order among the set of arguments. Since these arguments correspond to the position of criteria against some given actions, the existence of an order among arguments will result from the existence of an order among the criteria. In other terms, any preferential information among the criteria is taken into account by the structure of the arguments

corresponding to those criteria. Therefore, we propose to modify the acceptability function proposed in the Carneades model (see Function 2.1 in Chapter 2) to take into account the acceptability of the proof standard. Formally,

Definition 5.14. (*Acceptability of statements*)

Given an argument context C_m , a statement s is acceptable in an argument graph G if and only if:

- ▷ it satisfies its proof standard (ps), and
- ▷ its proof standard is acceptable

$$acceptable(s, G) = satisfies(s, ps(s), G) \wedge acceptable(ps, G) \quad (5.1)$$

We propose to distinguish two connected parts in the function. The first part, concerning the satisfaction of the proof standard by a statement, remains the same as in the earlier version of the function. In other terms, whether the proof standard is satisfied depends on the defensibility of the arguments pro and con the statement. The defensibility of an arguments depends on whether or not its premises holds. Finally, whether or not the premise holds depends on the dialectical status of the statement and the premise type.

We note that we can associate to any statements of the graph a proof standard (even the one identified in the Carneades model). In addition, for all statements that are not of type “supporting reasons” (SR dans la figure 5.7), we assume that the second part of Function 5.1, is by default true. Certainly, we have established in this work that the proof can be discussed or challenged and therefore the result depends on the proof used. However, by construction, there will always be **only one** argument pro such statements (e.g. conclusion, Simple majority, set of conditions, in the Figure 5.7). Thus, replacing a proof by another one will not affect the result. In other words, there is not really an aggregation operation which will affect the result to obtain, which is not the case at the aggregation level, where modifying the proof will affect not only the nature of supporting reasons but the final recommendation as well. Thus, statement acceptability, that is not SP, amounts to check if there exist sufficient evidence that supports the statements according to the used proof. In other words, we have to verify the first part of Function 5.1.

Now, to a statement at the aggregation level (i.e. supporting reasons), we will consider a multi criteria aggregation procedure as a proof standard (see discussion on the relation between proof standards and aggregation procedure in Section 4.2). Thus, determining the acceptability of the proof amounts to checking whether the use of that procedure is *justified or not*, depending on the arguments that support it. Thus, if at a step of a dialogue, the procedure becomes not justified, even if the statement satisfied this proof (in the sense of the Carneades), it is however not warranted to have an acceptable conclusion.

To be more concrete and for the sake of clarity, we will consider the example 5.17 to explain the acceptability of the proof standard “simple majority”. The example illustrates part of an argument graph constructed during a given interaction. We note that, the argument a_1 instantiates the argument scheme for the simple majority principle (see Table 4.11). The second argument a_2 is based on the set of properties that describes such a procedure (see Chapter 4 where each procedure is described by a combination of properties). Such properties provide a way to *support the use* of a procedure during the dialogue. We recall that if the set of properties are satisfied then the use of the procedure is justified or warranted (see Figure 4.6). Finally, the argument a_3 instantiates an argument scheme for MC-PW-EV and the arguments b_1 , b_2 and b_3 instantiate argument scheme for UC-PW-EV.

Example 5.17.

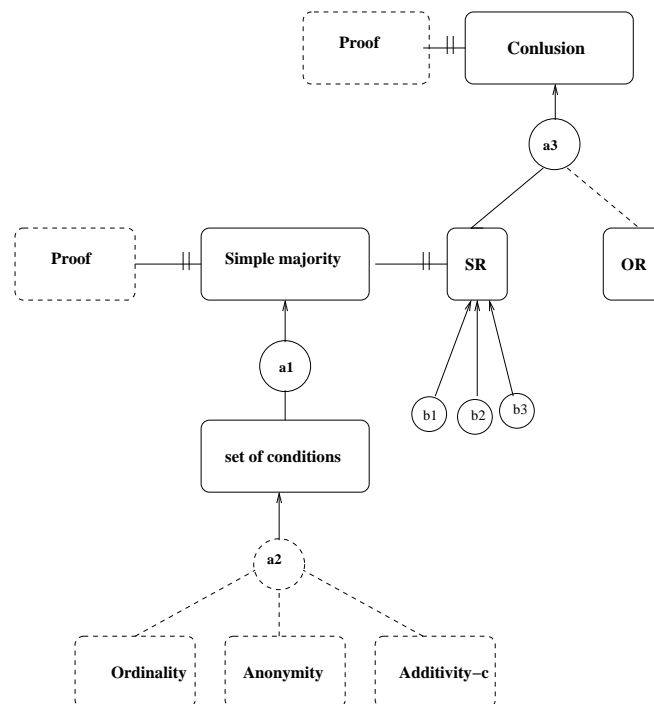


Figure 5.7 — Example of an argument graph

If we get back to the acceptability, we can say that the proof “simple majority” is acceptable if, like any statement in the graph, it satisfies its proof standard and this latter is acceptable. For a statement of type *proof standard*, as we explained before, the second part of Function 5.1 is assumed to be true by default, and the first part amounts to check if there exist a *defensible pro argument* supporting the procedure. In the example, it corresponds to the argument a_1 . Thus,

whether or not a_1 is defensible depends on whether or not the premise “set of conditions” holds. Whether or not this premise holds depends on its status and its type. Concerning the premise type, we consider that it corresponds to an *assumption premise*, because when the system chooses the procedure, it makes some assumptions and reason as if they are true until that a response of the user contradicts one of them. These assumptions reflect the fact that the conditions for using this procedure are met. For instance, for the majority method, an assumption is that *the criteria are of equal importance*.

According to the Carneades model, this kind of premise holds only if the dialectical status of the statement is *accepted* or *stated* (see Section 2.2.4). However, in our context this statement can neither be accepted or questioned directly by the user, it can only be stated or rejected by the system. The fact that the system considers such assumptions (or conditions) as true is due essentially to the fact that the properties describing the proof are considered satisfied. Thus, as these properties are satisfied the conditions are acceptable. Therefore, when choosing the procedure, the status of the premise “set of conditions” can only be *stated*, and becomes *rejected* if one of the properties of the procedure is challenged.

if p is an assumption representing “set of conditions”, then:

$$\text{holds}(p, G) = \begin{cases} \text{acceptable}(s, G) & \text{if } \text{status}(s) = \text{stated} \\ \text{false} & \text{if } \text{status}(s) = \text{rejected} \end{cases}$$

Consequently, we have to check the acceptability of this statement, by determining whether the argument pro this statement (here a_2) is defensible or not. Whether or not this argument is defensible depends on whether or not the premises holds. The premises correspond to the *set of properties* describing this procedure. In the example, the properties describing the majority are: ordinality, anonymity and additivity w.r.t coalitions. According to what we have discussed in Section 5.2, if the system chooses the majority principle as the proof, then it should assume that its properties are satisfied, and it moves to another procedure if one of its properties is challenged through a *critical response* provided by the decision maker (see Section 5.4). Thus, the justification of the use of a given procedure requires that the dialectical status of each property is “assumed true”, at the beginning of the process. Now, if the property is challenged, the status becomes “rejected” and the procedure is no longer warranted.

if p is an assumption and represents “a property”, then:

$$\text{holds}(p, G) = \begin{cases} \text{true} & \text{if } \text{status}(s) = \text{assumed_true} \\ \text{false} & \text{if } \text{status}(s) = \text{rejected} \end{cases}$$

To sum up, if the dialectical status of the premises corresponding to the properties is “assumed

true”, therefore, such premises holds in the argument a_2 . Thus, a_2 is a defensible pro argument supporting the conclusion “set of conditions”. As a result, its acceptability is true. Therefore, the premise of the argument a_1 holds and then a_1 is a defensible pro argument supporting the conclusion “simple majority”. Finally, the acceptability of the proof is true, because it satisfies its proof and this latter is by default acceptable.

Finally, as in the Carneades model, we can state the following:

Theorem 5.1. *In a multiple criteria context, with proof standard Scintilla of Evidence, Best Arguments, Dialectical Validity or a multicriteria aggregation procedure and for every argument in G : acceptable and holds are total functions.*

Proof :

The proof is an extension of the proof presented by [Gordon *et al.*, 2007] for the acceptable and holds function in the Carneades model. In other terms, we will consider that proof is with induction on the structure of argument graphs but we add the fact that a statement is linked to its proof standard which is an explicit statement of the graph. Moreover, our graph G is also finite and acyclic. If we consider the acceptability of any statement node s with no parents. A statement is acceptable iff it satisfies its proof stand and such proof is acceptable. Since G does not contain any argument pro s , with any proof standards s trivially is not acceptable, whatever the acceptability of the proof. We now examine the holds function for any premise p containing s . As we adopted the same distinction of premises for our context (i.e., ordinary, exception, and assumption) then, clearly all premises using s either hold or do not hold. However, we have a specific case where s is a particular statement with only two status: assumed true or rejected. In this case also, the premise either holds or does not hold, respectively when s is assumed true, or rejected.

Consider next any statement node s in G that is a parent node. Again s is acceptable iff it satisfies its proof standard and if the proof is acceptable. For the satisfaction of the proof, it is exactly as in the Carneades models. In other terms, all proof standards depend only on the arguments pro or con s that are in G . These arguments can be identified since by the induction hypothesis the theorem holds for all parents statements of s and all premises using them. Then, with all proofs it is clear that we can distinguish unambiguously either s satisfies or not its proof. Now, the proof is acceptable iff it satisfies its proof standard and in turn such proof is acceptable. In this case, we mentioned before that we can use any proof presented in the Carneades and therefore by default such a proof is acceptable. For the satisfaction side, a proof standard is considered as any statement in the graph, then its acceptability follows the same reasoning as previously (i.e., it depends only on the argument pro the proof). Thus, in the case where the proof is acceptable, then the acceptability of s depends only on the satisfaction of its proof. In all other cases, s is clearly not acceptable, whatever the arguments pro and con s . We finally examine the holds function for any premise using s . Since this function depends only on the

acceptability and dialectical status of s , clearly all premises using s either hold or do not hold.

■

5.6 Managing the dialogue

Dialogue games reflect interactions between different participants in dialogue or conversations. In such interactions, each participant intervenes by making utterances, according to a pre-defined set of rules. Typically, the rules define how the dialogue may or must start, what statements may or must be uttered in a given context and, how the dialogue may or must terminate.

In this section, we aim to formalise, through a dialogue game, the interaction between an analyst and a decision maker during the evaluation process. We have seen in Chapter 2 that there exist different types of dialogue, negotiation, persuasion, etc. (see Table 2.10). However, none of these dialogues seems to correspond to such type of interaction. Indeed, this interaction can not be treated as a persuasion or even a negotiation, because the initial situation is not a conflict of interest or opinion but a problem situation in which the decision maker has some difficulties in determining the best action to be undertaken. From this point of view we can assume that it can be closer to the deliberation dialogue where the aim is to reach an agreement on a plan, and the individual aims are to influence this agreement to their benefits. However, the individual aims in our dialogue are not the same. In fact, on the one side, the analyst has the aim to help and guide the client to construct a solution to his problem and to be convinced that it is a reasonable one. On the other side, the decision maker or the client, has to understand the problem and get convinced about the solution.

Type of dialogue	Initial Situation	Participant' Goal	Goal of Dialogue
Decision aiding process	“a problem situation”	<p>Analyst: help and guide the client to construct a solution to his problem and to be convinced that it is a reasonable one</p> <p>Client: understand the problem and get convinced about the action to undertake.</p>	to reach a consensus

Table 5.6 — Decision aiding process dialogue

Thus, as summarized in Table 5.6, we suggest to enrich the list of dialogues with the one of the decision aiding process. We note that in this kind of dialogue we can have several participants,

but in our work, we consider only two players an analyst and a decision maker. More precisely, between the system, which plays the role of the analyst, and the decision maker or the user. In what follows, we describe the details of the dialogue game used by our system, by presenting the locution rules, the commitment rules and a protocol (or dialogue rules). More precisely such a dialogue is based on recent extensions that incorporate argument schemes within dialogue [Reed and Walton, 2007]. Thus, to construct our dialogue game we adopt and extend the rules presented in [Reed and Walton, 2007].

5.6.1 Basic structures

Commitment stores To capture the dialogues between the players, we associate to each of them a *commitment store*, which holds the statements and the arguments to which a particular player is committed. For instance, following an assertion, the participant making that assertion will become committed to defending that assertion. Thus, in our context the commitment stores are noted CS_s and CS_u , respectively, for the system and the user.

Stack of Question Under Discussion (QUD) Also, we associate to each player a “*stack of question under discussion*” noted QUD_s and QUD_u , respectively, for the system and the user. It will contain the issues that have been raised and are currently under discussion in the dialogue.

5.6.2 Locution rules

The locution rules allow for making a claim or a statement, for challenging, conceding and retracting a claim, for supporting a claim with an argument, and for attacking arguments with counterarguments or by challenging their premises. In short, locutions rules indicate the moves that are permitted during the dialogue.

We define, first, the *Dialogue Statements* that we need to express the different types of information at the different stages of the evaluation process. More generally all types of statements required to instantiate any argument schemes. Such statements are set of declarative sentences constructed on the basis of preferential information provided by the decision maker and the performance table. Let, A is the set of actions; H is the set of criteria, $E = \{e_0, \dots, e_m\}$ is the set of evaluation scales associated to the criteria, and $\mathcal{P} = \{\text{ordinality, anonymity, additivity-c, additivity, dependence among criteria, veto}\}$ is the set of properties that allows to describe the proof standards. Thus, if s is a statement then it can corresponds to a:

- ▷ **Recommendation:** $recommended(x)$, where $x \in A$.
- ▷ **Preferential information:** decision maker’s preferences are represented by binary relations that reflect different situations, namely:

- Preference for an action that can be expressed either on a single criterion or on a coalition of criteria: $Pref_H(x, X)$, where: $x \in A$, $X \in 2^A$ and $H \in 2^A$;
 - Not Acceptability (resp. Acceptability) of an action on a given criterion: $\neg Acc_{h_i}(x)$ (resp. $Acc_{h_i}(x)$), where $x \in A$. The non acceptability of an action is the conclusion of the argument scheme UC-IN-ACC or UC-PW-ACC.
 - Preference among criteria: $Pref(h_i, h_j)$ where $h_i, h_j \in H$.
- ▷ **Performance information:** to each action we can associate a performance $Perf : h_i(x)$ where $x \in A, h_i \in H$.
- ▷ **Comparison among performances:** this comparison allows us to build a binary relation between two actions: $Comp(Val(X), Val(Y))$, where $X, Y \in Perf$; $Comp \in \{<, >, =, \geq, \dots\}$ and $Val : Perf \rightarrow e_i$. When there is no ambiguity we overload the notation and $h_i(x)$ refers to the value of x on the scale of h_i (e.g. $h_i(x) \geq h_i(y)$).
- ▷ **Definitional information:** $Defines(Comp(f_1(X), f_2(Y)), Conc)$, where $Conc \in \{\text{Preferential information}\}$, f_1 and f_2 are functions such that $f : e_i \rightarrow e_i$. (e.g. $h_i(x) < h_i(y) + 30 \Rightarrow \neg Acc_e(x)$).
- ▷ **Proof standard information:** we represent the proof standard associated to each statement as a particular statement in the argument graph. We recall that at the aggregation level the proof corresponds to the aggregation procedure. Thus to each aggregation we have a specific label *proof_id* which allows to recognize the selected procedure (eg. **Majority** for the simple majority procedure).
- ▷ **Scheme Information:** for each argument scheme we associate a label *scheme_id* which allows to recognize the used argument scheme. We consider all the argument schemes presented in the Chapter 4, e.g., MC-PW-EV, UC-PW-EV, ...
- ▷ **Property Evidence Information:** for each property we can associate a Positive Evidence, noted $PEvidence(p)$ and a Negative Evidence, noted $NEvidence(p)$, where $p \in \mathcal{P}$ (see Table 5.5).

These constitutes the set of all well defined formed dialogue statements \mathcal{DS} . They are summarized in the Table 5.7. Moreover, to each statement described above an implicit proof standard is associated.

The different locution used in our dialogue game are described below and summarised in the table 5.8 by distinguishing the explicit reply that we can have for each locution. For such purpose, we use the notions of replies (or attacks) and surrenders defined by [Prakken, 2005a]. Let $\phi, \varphi \in \mathcal{DS}$

Dialogue statements	
Recommendation	$recommended(x)$
Preferential information	$Pref_H(x, X), \neg Acc(x), Pref(h_i, h_j)$
Performance information	$Perf : h_i(x)$
Comparison among performances	$Comp(Val(X), Val(Y))$
Definitional information	$Defines(Comp(f_1(X), f_2(Y)), Conc)$
Proof standard information	$proof_id$
Scheme information	$scheme_id$
Property evidence information	$PEvidence(p), NEvidence(p)$

Table 5.7 — Dialogue statements (\mathcal{DS})

- ▷ **Assert**(ϕ). This allows to exchange information.
- ▷ **Accept**(ϕ). It allows to accept a claim.
- ▷ **Challenge**(ϕ). The challenge requests some statement that can serve as a basis for justifying or explaining ϕ .
- ▷ **Retract**(ϕ). This locution is for withdrawal (or retraction) a claim.
- ▷ **Argument**. To support a claim we can use the locution $argue(\phi, premises, \Pi(scheme_id))$, where Π allows to synthesize a hierarchy of schemes¹. We note that in some cases, it is possible to combine an argue with an Assert ($PEvidence(p)$ or $NEvidence(p)$). For instance, when the user provides a critical response against the recommendation.
- ▷ **Critical question**. To challenge an argument we can ask one of its associated critical question by using the locution $pose(C, type, conclusion, scheme_id)$, where:
 - C : represents one of the different questions associated to the argument $scheme_id$;
 - $type$: indicates if the question is an assumption or an exception;
 - $conclusion$: is the conclusion of the attacked argument.
- ▷ **Which proof?** It allows to reveal the proof in use, by the following: $according\ to(pf)$, where pf is a Proof Standard information.

Moreover, for each statement is associated a dialectical status during the dialogue. The basic status are *stated*, *questioned*, *accepted*, *rejected* or *assumed true*. Such statuses are determined, during the dialogue, by the precedents locutions (called also speech acts).

¹for instance, Π allows to provide a minimal explanation as detailed in Section 5.3

Moves	Replies	Surrenders	Status of the claim
assert (ϕ)	challenge (ϕ) which proof? pose(C , type, ϕ , scheme_id) (if $\phi = \text{recommended}(x)$ and C concerns the acceptability)	accept (ϕ)	stated
challenge (ϕ)	argue (ϕ , premises, $\Pi(\text{scheme_id})$)	retract ϕ	questioned
argue (ϕ , premises, scheme_id)	challenge (premise) pose(C , type, ϕ , scheme_id) assert(ϕ) which proof?	accept ϕ accept(premise)	accepted
pose(C , type, ϕ , scheme_id)	challenge($\text{not} - C$) assert (C) assert ($\text{not} - C$)	retract(ϕ)	rejected
accept ϕ			accepted
retract ϕ			rejected
which proof?	<i>according to (pf)</i>		

Table 5.8 — Locutions rules for a decision aiding dialogue

Figure 5.8 gives a simple state transition diagram showing the the types of moves that the players can make and the choice of move which is then available in the new state. It also shows the moves that lead to the roles of system and user being switched and how the game can terminate. The system opens the dialogue by stating (asserting) a statement, which is in reality the recommendation for the decision problem posed by the user. After a statement, the user can either accept it, ask for a justification (challenge). When a justification is required, it is asked to provide sufficient supporting evidence for the claim challenged (argue). Such evidence can be accepted or discussed by asking a critical question.

However, such a graph is binding because it imposes a fixed sequential reasoning and does not offers a flexibility in the responses or the moves. In practice, for instance a user may wish to intervene at any time of the interaction to pose a question or to contradict previous assumptions of the system. For instance, in the graph the only moment where it is permitted to pose a critical question is after an argue. However, in our context, an example of a critical question is “is the performance correct?”and the user may use it at any time of the dialogue even if the argument scheme concerned was not presented at the last move. In fact, the user may realize belatedly, after a justification, that the value of an action is not correct and therefore proceeds immediately to change it. Therefore, in order to represent simply the behaviour of the user, we propose to distinguish the following commitment rules and dialogue rules. We emphasize that contrary to many dialogue games the rules described here give different opportunities to the decision-maker

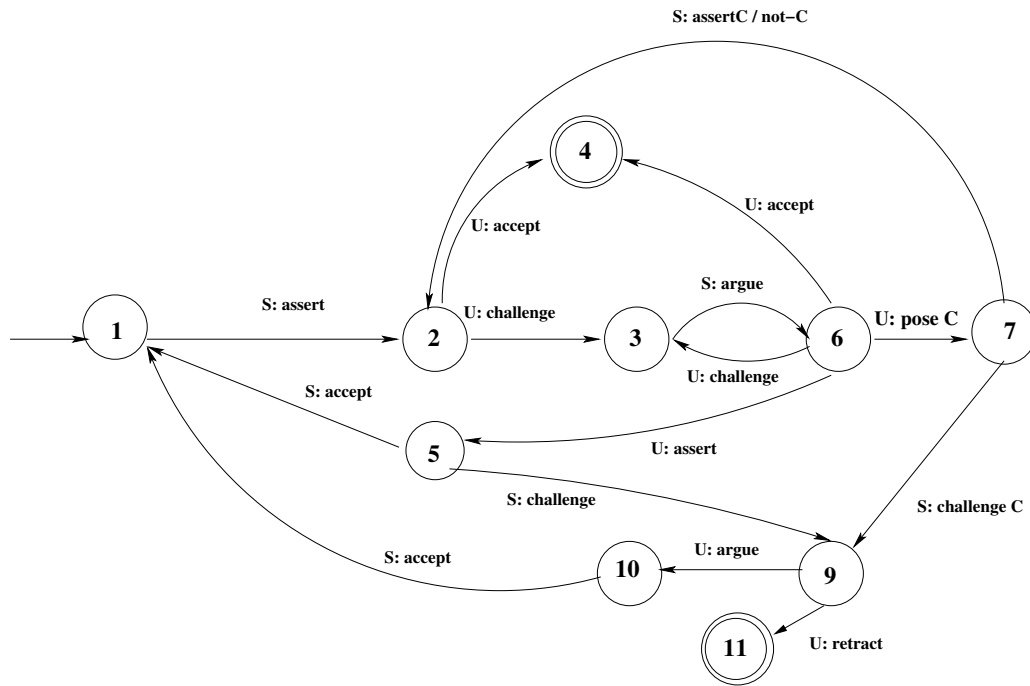


Figure 5.8 — State Transition Diagram

and the system, illustrating the asymmetry of the situation.

5.6.3 Commitment rules

They are rules which define the circumstances under which participants express commitment to a statement (or a claim). They determine the consequences of each move in dialogue on the commitment stores of each participant. The initial state of the commitment store is given by the information provided by the decision maker. For instance if he expressed that there exist differences among criteria, therefore such an information should be included to the $CS = \{Pref(h_i, h_j)\}$. In what follows we present the different rules adopted for our dialogue game.

R_0 After a player states a statement, it is included in his commitment store

R_1 After the withdrawal (the retraction) of the statement, it is deleted from the CS. Only NEvidence and PEvidence cannot be retracted².

R_2 the challenge of a statement, places the statement in the hearer's commitment store unless it is already there or unless the hearer immediately retracts his commitment to this issue.

²The reason is that we do not want, at this stage, to give the DM the ability to explicitly discuss the properties. We leave this open for future research

- R_3 the challenge of a statement, places statement in the hearer's stack of questions under discussion (QUD);
- R_4 After an argue, the argument scheme used, the premisses and the assumption of the scheme are included in the commitment store of the speaker. Indeed, unlike in the traditional dialogue game where only the claim is stored in the CS , we suggest to add the whole argument scheme used to justify the claim. This allows to distinguish the different critical questions that is possible to ask. Moreover, it allows "backtracking" moves where the critical questions can be posed against statements made earlier in the dialogue (to have a more flexible dialogue [Prakken, 2006]).

5.6.4 Dialogue rules

They are rules for regulating the moves. They can be separated into three different categories, depending on the kind of the moves: *updating*, *attacking and responsive moves* [Prakken, 2005a]. The rules for attacking move, represented by the $[R_6 - R_{10}]$, are used to prohibit the non relevant moves. On the other hand, the rules for responsive moves, represented by $[R_{11} - R_{18}]$, are more strict because they impose an immediate reply.

Rules for updating.

- R_5 At any time, at his turn, the user is allowed to update (retract + assert) a value statement of the CS_u .

Rules for attacking moves.

- R_6 It is not allowed to challenge an issue already under challenge, or upon which a speaker is already committed. In other terms, to challenge an issue, it should not be in the speaker's stack of QUD.
- R_7 It is permitted to challenge any statement the other participant is committed to, unless it does contradict R_1 ;
- R_8 It is not permitted to the speaker to state something that he is already committed to, or which contradicts something he is committed to.
- R_9 It is allowed to issue a statement which contradicts a commitment of the other player, unless it does contradict R_3 .
- R_{10} It is permitted to ask any of the critical questions attached to an argument scheme upon which the other agent is committed, as long as this question is not already under discussion.

Rules for responsive moves.

- R_{11} After an update, the system must inform of the consequences ($\text{assert}(\text{recommended}(x))$);
- R_{12} After an $\text{assert}(\text{recommended}(x))$, it is allowed to ask only a critical question on the acceptability of x (i.e. critical question number 6 in the Table 5.4);
- R_{13} Each speaker takes his turn to move by advancing once locution at each;
- R_{14} it is an obligation to respond to a challenge by either:
- (i) retracting the issue, only in the case of the user, or
 - (ii) presenting an argument for that issue.
- R_{15} after an accept, inform of the consequences on the current state of the recommendation;
- R_{16} after an assert (from the user), if there is a contradiction, inform of the consequences by making explicit that contradiction.
- R_{17} after a statement has been offered in response to a challenge then if the response is a substitution instance of some argument scheme of the game, the locution Pose C is a legal move, where C is a critical question of that scheme appropriately instantiated.
- R_{18} it is an obligation to respond to a critical question C by:
- (i) if C is an assumption of its argumentation scheme, the move is followed by:
 - ▷ assert C ;
 - ▷ assert $\text{not} - C$;
 - ▷ retract C .
 - (ii) If C is an exception to its argumentation scheme,
 - ▷ assert C ;
 - ▷ assert $\text{not} - C$;
 - ▷ retract C ;
 - ▷ challenge $\text{not} - C$;

The two parts of the rule R_{10} allows to take into account the fact that the critical questions can be of two types, assumption and exception. For example, an assumption of the scheme MULTI CRITERIA PAIRWISE EVALUATION (see Chapter 4) is that the performance of an action is indeed correct (see Table 5.4). If this questioned the speaker must state that it is the case, or that it is not, or withdraw commitment to it. Similarly the question “is the action acceptable?” for the same scheme allows a critic to probe the action’s acceptability. If this question is posed, the proponent can respond with not only statement or withdrawal moves but also a challenge, ‘Why is the action not acceptable?’.

5.7 Properties of the dialogue game

Finally, according to [Prakken, 2006], in general it is difficult to prove formal properties of dialogue without making strong assumption of agents behaviour (and strategies). For example in our case, we may not prevent the decision maker to change endlessly his view and modify different parameters of the decision problem. In what follows, we prove a formal property of our system in the case where the decision maker *cannot update* the information given.

Property 5.5. *If the decision maker does not update the information of the decision problem, then the dialogue terminates.*

Proof :

If updates are not possible, then the rule R_5 can be applied. Now different cases are possible:

Regarding a recommendation given by the system, the user can either accept, therefore the dialogue terminates, or challenge the recommendation. Each time the user makes a challenge (i.e. why?), the graph (representing the argument exchanged during the dialogue) is unfolded and our graph is finite (it is composed of three levels).

To the first challenge, the system provides the different steps of reasoning by revealing the mutli criteria level that supports the conclusion at the highest level. As a result, the different statements are placed in the CS_s and QUD_s (rules R_2 and R_3). The system, now, should reply either by an argue or a retract (rule R_{14}). In the case of a retract, the dialogue terminates. However, the user can reply to the argue by a challenge. Therefore, the system unfolds completely the graph (i.e. all the levels are explicitly given). In this case, the user can no longer use a “why?” because, according the rule R_6 he is not allowed to challenge an issue upon he is already committed to. In fact, the lower level of the graph contains all the information provided by the user at the beginning of the process. Such information represent the initial state of the commitment store.

Therefore, the user can either accept, then the dialogue terminates, or propose a new action as the recommendation. We note that he cannot use the critical questions because the rule R_5 is not allowed.

In the case, the user proposed a new action as the recommendation (knowing that the set of actions is finite), the system either accept then the dialogue terminates or challenge the user to provide evidences supporting his recommendation. According to the rule R_{14} , the user can retract its proposition and then the dialogue terminates or give an argument. In the case where the user provides an argument, he will probably “challenge” the current proof used by the system. Technically, the challenge provides information that discusses indirectly the properties of the proof (see Table 5.5). More precisely, the user provides a *critical response*, i.e. *argue + assert(PEvidence or NEvidence)*. The critical responses allow the system to navigate among the lattice in order to choose a

new proof and to propose a new recommendation. The user can always challenge the recommendation, but on the one hand, he cannot do it indefinitely (see the discussion above), on the other hand two things are not allowed. First, he is not allowed to retract a PEvidence or a NEvidence, and therefore to back to a node previously visited in the lattice. Second, he is not allowed to state a NEvidence when he is committed to a PEvidence and vice versa (rule R_8). With such two conditions and the fact that the lattice is finite, we are sure that at some point the dialogue, the system and the user will agree on a particular recommendation and therefore the dialogue terminates.

■

5.8 Conclusion

We presented in this chapter different tools and concepts to enhance the decision capabilities of a decision aiding tool. First, we proposed a first contribution to tackle the problem of choosing an aggregation function during the evaluation process. Our claim is that it is not obvious to engage with the decision maker in a long process of questions/replies to select the appropriate aggregation. Thus, we propose to construct a Concept lattice such that the nodes correspond to the procedures and the links to the fact that a property was satisfied or contradicted depending on the response of the user. Such a lattice will allow the system to navigate among the set of candidate of different proof standards during the interaction. Second, we addressed the problem of building explanations in the case where the recommendation of the system is a Condorcet winner. We have seen that different building blocks provide different explanations. Such explanations were discussed on basis of their minimality and cost. After that, we discussed the aspect of the evaluation of the arguments exchanged during the dialogue. On the one hand, we presented different ways that allow the decision maker (or the user) to discuss, to challenge or to correct the proposition and assumptions of the system namely, critical questions and critical responses. Such tools allow, on the one side the user to point the elements that are wrong or missing in the model, and on the other side the system to make revisions and updates. On the other hand, we adapted an acceptability function to the context of multi criteria evaluation, where the proof standard is subject to challenge and discussion. Such a function offers a way to evaluate the arguments in favour and against a given conclusion, in an argument graph. Finally, we proposed different rules to represent the interaction between a system and a user, through a formal dialogue game, based on different types of rules and loctions.

In the next chapter we propose to show how such different features can be taken into account in an implemented system.

6 ArgDec: a prototype for an Argumentation based Decision aiding tool

The aim of this chapter is to show how the different elements and concepts presented in this work, can be used in a “real” application. To present this example, we implemented a system realised under the form of a simple GUI (Graphical User Interface) in Java. Such a system allows to simulate a dialogue between an analyst (here played by the system) and a decision maker in a decision aiding situation. Thus, the system will allow to: structure the decision problem, to calculate the recommendation and to generate automatically the minima explanation that allows to justify it, select the proof standard on the basis of the rules and properties described in the previous chapters; and finally to inform the user of changes during the interaction by both a textual and graphical representation.

6.1 Introduction

We present in this chapter a prototype for a decision aiding tool, called ArgDec (Arguing and Deciding). This prototype is implemented under the form of GUI in Java. The idea is to show how the different features described in this work are taken into account through an implemented system. More precisely, we aim to show how the tools of argumentation and decision aiding can be integrated into an automatic tool.

Although there was until recently very little attention in the decision analysis literature 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, studying for instance the impact on the efficiency of recommendations

when greater interaction with the user is allowed, or when explanations are provided [Pu and Chen, 2007]. Due to 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 options as opposed to another one).

With the system presented in this chapter, we seek to enhance the capability of the decision support to provide explanation by providing methods for justifying their advice or recommendation in an intuitive manner and on the other hand, to offer the opportunity to the user to observe step by step the effects of the interaction on the final result.

The chapter is organized into two sections. The Section 6.2 presents a general description of the tool. The Section 6.3 is devoted to present an example of an interaction between the system and the user.

6.2 ArgDec: an overview

We propose in this work to implement a simple and basic prototype which aims: to calculate the recommendation of the decision problem, to explain the recommendation and to inform of the consequences of changes made by the user. Our system is based on the various proposals presented in this document. We used Java for the implementation of different graphical interfaces and functions. The introductory screen of the system is illustrated in the Figure 6.1, where the user is invited to enter the different data and information in order to specify his decision problem. The user should give information on three basic elements: *problem statement*, *set of criteria*, and *set of alternatives*.

During the interaction the system constructs an arguments graph to visualize the exchanged information. The main elements of the arguments graph are illustrated in the figure 6.2. First we recall that an arguments graph is a set of nodes and links. The nodes can be arguments or statements. The arguments are represented by circles and the statements by boxes. There are two kinds of arguments pro and con. Second to each statement is associated a proof standard that allows to aggregate the arguments constructed in favour and against that statement. Such a link is called according to (see Definition 5.12). Finally, the argument graph allows to evaluate the arguments and the proposition. Thus, the reader will note that the boxes of the different statements do not have the same colour. Different colours correspond to different acceptability status of a statement: *green* for acceptable (true) *red* for non-acceptable (false).

Finally, the set of information provided by the user is recorded in a specific file, as it is shown in the figure 6.3. In fact the system generates an XML (eXtensible Markup Language) file that includes on the one hand, the data provided by the user at the beginning of the process, namely: the problem statement, the number of alternatives, the number o criteria and the performance



Figure 6.1 — Introductory screen

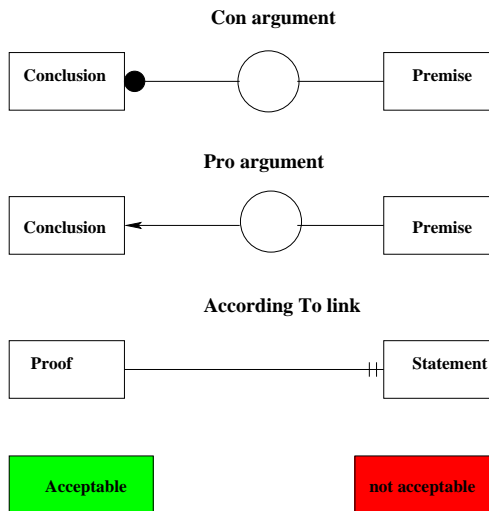


Figure 6.2 — Main elements of the arguments graph

table. Of course, as we shall see, other information can be added; on the other hand, the file will contain all the solutions or recommendations computed by the system at each step of the dialogue, such as: the recommendation of the problem, the different possible coalitions to explain the recommendation, and the minimal coalition to get a minimal explanation. Of course if the user updates an information and a new recommendation is provided, such an information is

recorded in the file. The file allows us to get at the end of the interaction a report summarizing all important data.

```

- <Alternatives>
  <Alternative_Nombre Nombre="4"/>
  <Alternative0 id="a0" Performance0="7" Performance1="6" Performance2="2" Performance3="3"
  Performance4="5"/>
  <Alternative1 id="a1" Performance0="6" Performance1="4" Performance2="8" Performance3="4"
  Performance4="7"/>
  <Alternative2 id="a2" Performance0="3" Performance1="2" Performance2="5" Performance3="5"
  Performance4="3"/>
  <Alternative3 id="a3" Performance0="7" Performance1="7" Performance2="2" Performance3="0"
  Performance4="2"/>
</Alternatives>
- <Criteria>
  <CriteriaRelation/>
  <Criteria_Nombre Nombre="5"/>
  <Criterion0 id="h0"/>
  <Criterion1 id="h1"/>
  <Criterion2 id="h2"/>
  <Criterion3 id="h3"/>
  <Criterion4 id="h4"/>
</Criteria>
- <Solution0>
  <solution Gagnant="a1" Indice="1"/>
</Solution0>
- <Accepted_coalition0>
  <coalition0 indice0="2" indice1="3" indice2="4" Nbr_zero="0" Taille="3"/>
  <coalition1 indice0="0" indice1="2" indice2="3" indice3="4" Nbr_zero="2" Taille="4"/>
  <coalition2 indice0="1" indice1="2" indice2="3" indice3="4" Nbr_zero="2" Taille="4"/>
  <coalition3 indice0="0" indice1="1" indice2="2" indice3="3" indice4="4" Nbr_zero="4" Taille="5"/>
</Accepted_coalition0>
- <Minimal_coalition0>
  <Coalition0 indice0="2" indice1="3" indice2="4" Nb_zero="0" Taille="3"/>
Terminé

```

Figure 6.3 — Screen shot of the XML file

In what follows, we present an example of dialogue between the system and a user. At each step of the dialogue (presented in a specific screen shot), we discuss the result by relying on the concepts presented in previous chapters.

6.3 Illustrative example

In this section we detail an example of an interaction between the system and a user who has a decision problem. First, the Figure 6.4 allows to specify the *decision problem statement* of the user. To do that, the user has to reply to three questions constructed on the fact that we can distinguish problem statements on the basis of partitioning of the set of action A (see section 1.3 in Chapter 1). However, due to the hypothesis 1.1 considered in this work, the problem statement is predefined and corresponds to the *choice problem*. In other terms, the check boxes, *No*, *yes* and *Two*, corresponding respectively, to the first, second and the third questions, are checked in advance and the user did not really need to specify the problem statement.

Second, the user provides information on criteria. The Figure 6.5 allows to enter different types

Figure 6.4 — Problem statement screen

of information : the number of criteria, labels to describe each criterion, information on scales and threshold if any, etc.

Figure 6.5 — Information on criteria screen

Figure 6.6 — Information on alternatives screen

The only information which is really necessary is the number of criteria. The other ones are optional but if they are filled, the system will take them into account when it computes the recommendation. For the moment, the user specifies only the number of criteria, which in the example is five, then the system will consider the set $H = \{h_0, h_1, h_2, h_3, h_4\}$. After that, the user fills the fields concerning the set of alternatives (see Figure 6.6). In this case, two information are of primary importance; the number of alternatives and the performance table. Concerning the number of alternatives, in the example he chooses four alternatives, i.e., $A = \{a_0, a_1, a_2, a_3\}$ and for the performance table, he provides the Table 6.1.

Finally, the user can ask the system to resolve his problem by pressing the button “resolve”

	h_0	h_1	h_2	h_3	h_4
a_0	7	6	2	3	5
a_1	6	4	8	4	7
a_2	3	2	5	2	3
a_3	7	7	2	0	2

Table 6.1 — Performance Table

(see figure 6.1). The system computes, therefore, the recommendation and displays three frames on the screen, as it is illustrated in the figure 6.7. The first frame, named *Dialogue Box*, is the interface through which the user will interact with the system. On this frame, the system suggests that: “ a_1 is the best choice” for his decision problem. The second frame, named *Data*, summarizes the set of data provided by the user. In this example, it represents the performance Table. Finally, the last screen is for displaying the arguments graph constructed during the interaction. At the first turn, the only thing that is showed is the recommendation.

After that, according to the different rules of our dialogue game, presented in Section 5.6, the user can either *accept* the recommendation or *ask* for justifications. To do that, the user will use the menu *Replay*, where the items: “**Accept**” are “**Why?**” are available.

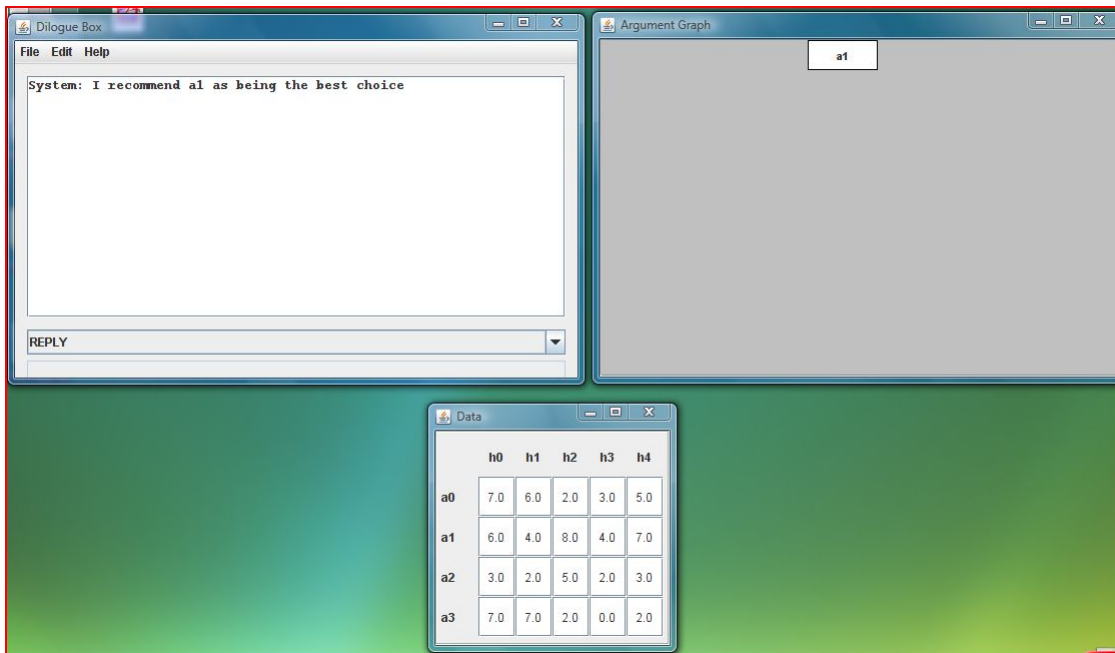


Figure 6.7 — Stating a recommendation

In this example, the user challenges the recommendation, and the system presents a first expla-

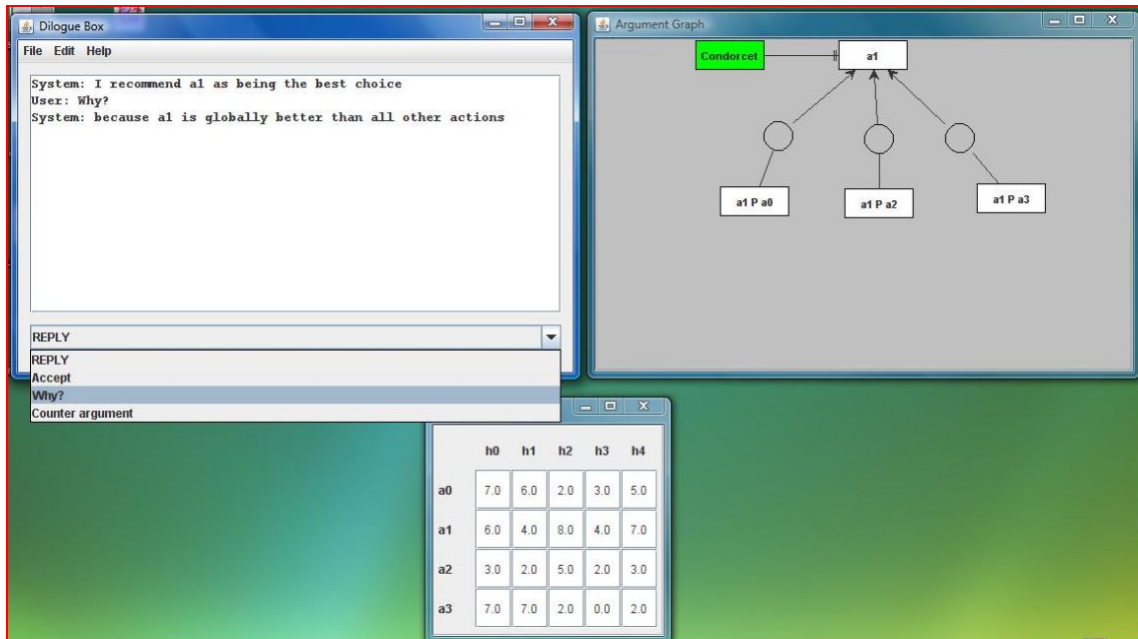


Figure 6.8 — A first level of explanation

nation, as it is illustrated in the Figure 6.8. It suggests that that: “ a_1 is globally better than all the other actions” (technically is the Condorcet winner). At the same time, we can see, on the graph, that different arguments pro the final conclusion were constructed (see the right frame in the figure 6.8). Such arguments represent the supporting reasons that allow to conclude that a_1 is the best choice. The premises of such arguments represent the pairwise comparison of the winner with the other actions. More specifically, they are conclusions of different argument schemes for MULTICRITERIA PAIRWISE EVALUATION (MC-PW-EV) (see Table 4.17). Moreover, the graph makes explicit the proof standard used at this level to conclude that a_1 is the best choice. It corresponds to the “Condorcet rule”.

On the other hand as we specified at the beginning of this chapter the system computes the acceptability status of each statement, according to Function 5.1 (see Chapter 5). We note that at this stage of the dialogue, the final conclusion is not acceptable because, even if the proof is acceptable, the conclusion does not satisfy its proof. In fact, the arguments supporting the conclusion are not defensible. We recall that the defensibility of argument depends on whether or not its premises hold or not. In our example, as the premises are of *ordinary type* and their dialectical status correspond to *stated*, then to conclude that they effectively hold we need to check if they are acceptable or not (see Section 2.2.4).

To sum up, we have two different statuses that we can associate to a given statement: the dialectical status and the acceptability one. The former is determined by the speech acts of the dialogue

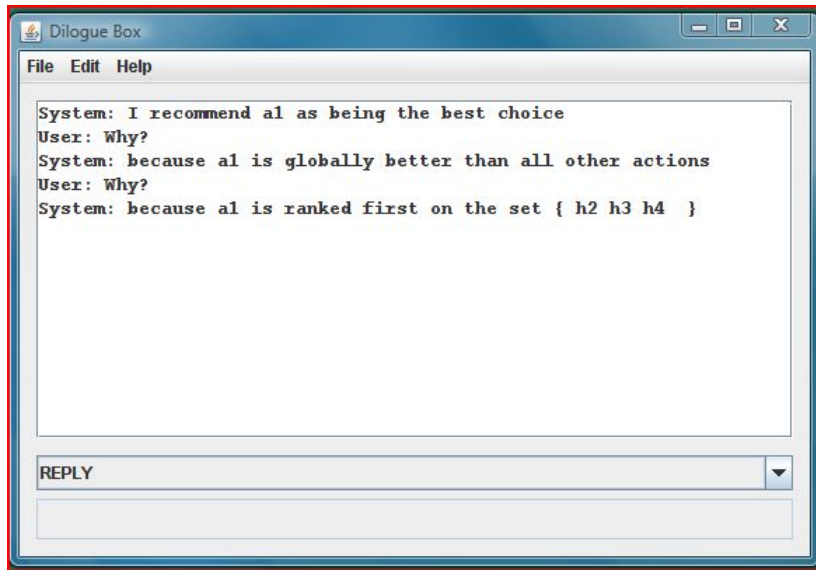


Figure 6.9 — Minimal explanation

but is not visible on the screen. For instance, in the Figure 6.7, the dialectical status of the recommendation is *stated*, which is also the case for the different premises a_1Pa_0 , a_1Pa_2 and a_1Pa_3 . But, when the user asks for explanation (i.e. why?), the status becomes *questioned*. Such statuses are used in the computation of the acceptability of the statement (see Section 5.5). The Annex E summarizes the dialogue game corresponding to the example of this chapter by presenting the different locutions used in the game and dialectical status of the different statements.

In we get back to the dialogue, the user now asks form more clarification because it is not satisfied with the first explanation. We draw the attention of the reader to the fact that the user was able to play another move such as accepting or stating a counter-argument (see figure 6.8).

In response to the request of the user, the system automatically generates a minimal explanation according to the principle described in Chapter 5. In this case, it is sufficient to explain that: “ a_1 is ranked first on criteria h_2 , h_3 , and h_4 ”, there is no need to mention criteria h_0 and h_1 (see Figure 6.9). Of course, to such an explanation, a graphical representation is associated (see Figure 6.10). First, we note that the graph is constructed on the basis of the hierarchical structure of argument presented in Chapter 4. Thus, we can recognize, the elementary level, the aggregation level and the multi criteria level. Second, the graph displays the different informations, provided in the explanation, to support the final conclusion. In other terms, for each pairwise comparison between the winner and an action of the set A , a subgraph, corresponding to the minimal coalition of criteria, is constructed.

Let us give the details of the subgraph of the conclusion a_1Pa_0 . This graph allows to conclude that

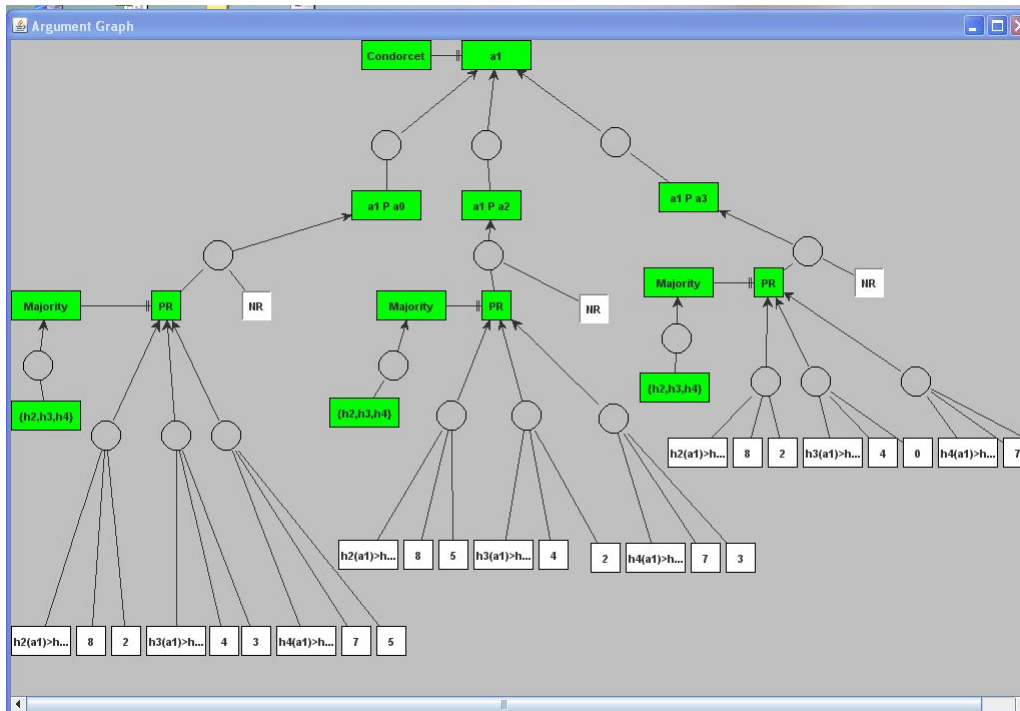


Figure 6.10 — Graph for a minimal explanation

this conclusion is warranted, because it is supported by a set of supporting reasons and there are no opposing reasons. This corresponds to the argument scheme on the top of our hierarchy, i.e. MC-PW-EV. As we mentioned in Chapter 4, the supporting reasons are constructed according to a specific proof standard. In the graph, the system presents explicitly such a proof and corresponds to the “simple majority method”. More concretely, the choice of the proof was done by the system at the beginning of the process, immediately after that the user provided the whole specification of his decision problem.

The problem of choosing a proof standard was addressed in this work and it constitutes an important step in the evaluation process. It depends on the information available at each step of the interaction. To facilitate the process of choosing an aggregation procedure, we proposed to base the system’s reasoning on the fact that it has to choose the most “simple” and “intuitive” procedure. Therefore, for an initial set of information $I_0 = \{\text{performance table}\}$, the system associates the proof $r_0 = \{\text{simple majority}\}$. At the same time, the system assumes that the properties describing such a proof are *satisfied*, such that their dialectical status correspond to “assumed true”. In other terms, according to the lattice presented in the previous chapter (see Figure 5.4), the system chooses the node (Simple majority, 111000). We draw the attention of the reader on the fact that such information is not visible to the user and only concern the system. It is useful in order to guide the selection of a procedure during the interaction (i.e. to navigate among the lattice).

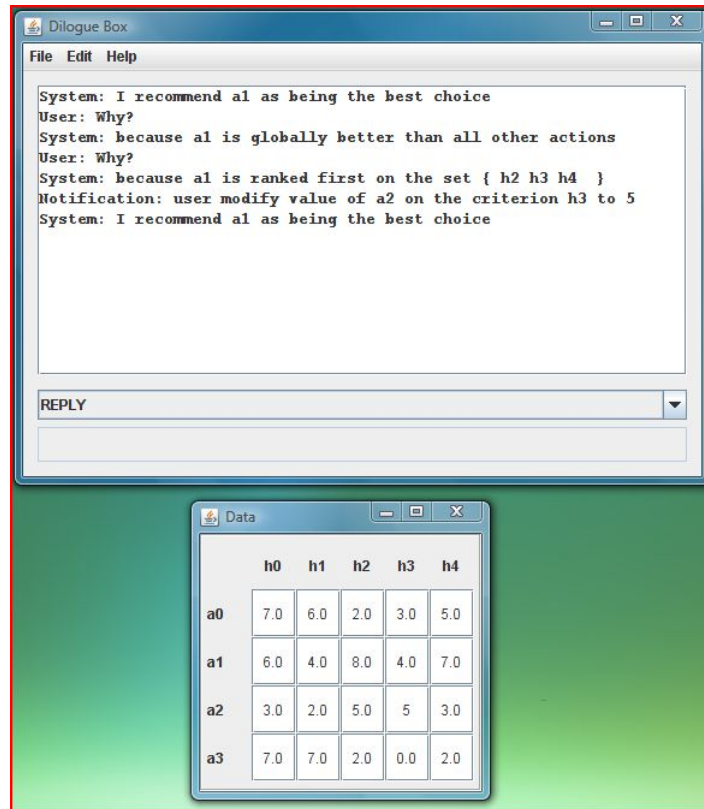


Figure 6.11 — Example of update

The only thing that the user can see on the screen is that the procedure is supported by their own argument scheme (for instance for the majority, see Table 4.11).

On the other hand, the arguments that support the conclusion $a_0 P a_1$ are depicted in the figure. They correspond to the minimal coalition of criteria exhibited in the explanation, i.e., $\{h_2, h_3, h_4\}$. Thus, for each criterion, and argument scheme for the pairwise comparison in constructed (i.e. UC-PW-EV). We note that for the sake of clarity, the conclusions of the elementary level, corresponding for instance to “ a_1 is at least as good as a_0 on the criterion h_2 ”, are not represented in the graph. Moreover, the premises of that arguments represent the different information provided by the user at the beginning of the process.

Now, confronted with that explanation, the user noticed that the value assigned to the alternative a_2 on the criterion h_3 is wrong. Therefore, he updates and corrects the value directly on the screen (see Figure 6.11). In fact, the tool allows to change the value on the performance table automatically. Any correction is taken into account by the system and it is indicated by the message “Notification:” on the Dialogue Box. Moreover, the system should provide a new recommendation as it is shown in the Figure 6.11. In fact, according to the rule R_{11} of our dialogue

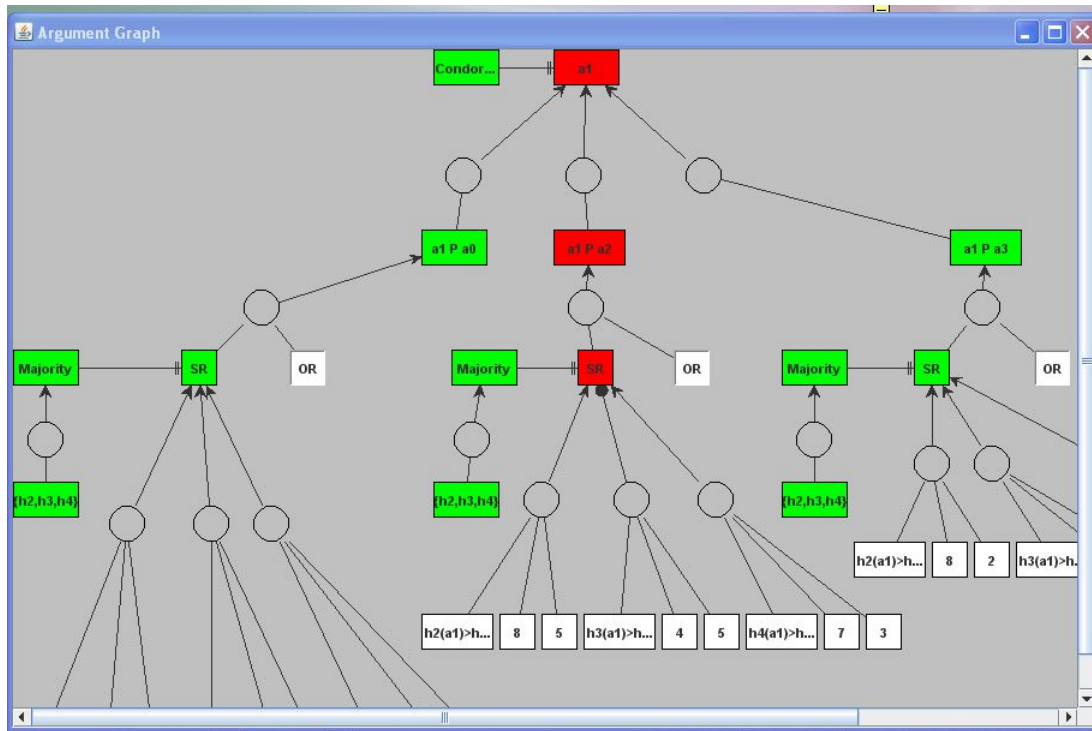


Figure 6.12 — Graph after the update

game, the system should inform of the consequence of the modification by computing a new recommendation on the basis of the new information.

More concretely, such an update is interpreted by the system as an attack or a critic against the argument on the criterion h_3 . Such an attack is built through the critical question number 7: “Is the performance correct?” (see Table 5.4 in Section 5.4). Thus, as we can observe in the Figure 6.12, the fact that the user changes a value, the argument h_3 is no longer a support for the conclusion $a_1 P a_2$ but becomes a counter argument. Thus, the statement “SR” does not satisfy its proof (majority) and therefore becomes not acceptable (see Section 5.5). In fact, in the previous chapter we presented a new version of the acceptability function. It suggest that a statement is acceptable if and only if its satisfies its proof and this latter is acceptable. At this stage of the dialogue, the proof is acceptable but the statement does not satisfies the proof because the number of defensible pro arguments does not correspond to the majority on the set of criteria $\{h_2, h_3, h_4\}$. As a result, the final conclusion a_1 is no longer warranted and at the same time not acceptable.

Consequently, the system takes into account the new information and provides the corresponding recommendation, which is, in this example, the same, i.e. a_1 . The system justifies that choice by presenting a new minimal explanation. It suggests that: “ a_1 remains better that anyone else on h_2 and h_4 and even if a_2 is better than a_1 on h_3 , a_1 beats a_2 on h_0 ” (it could have chosen h_1 as

well) (see Figure 6.13). Such a minimal explanation is constructed on the basis of the principle presented in Section 5.3.

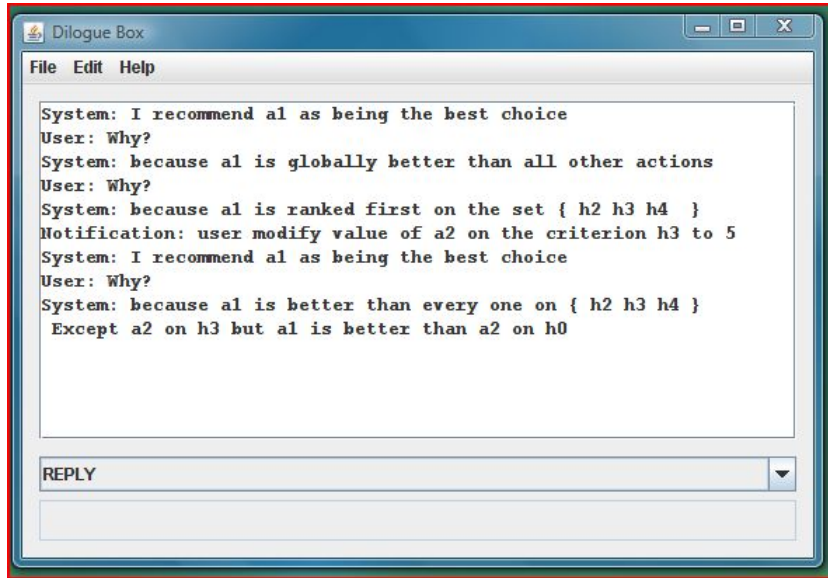


Figure 6.13 — New minimal explanation

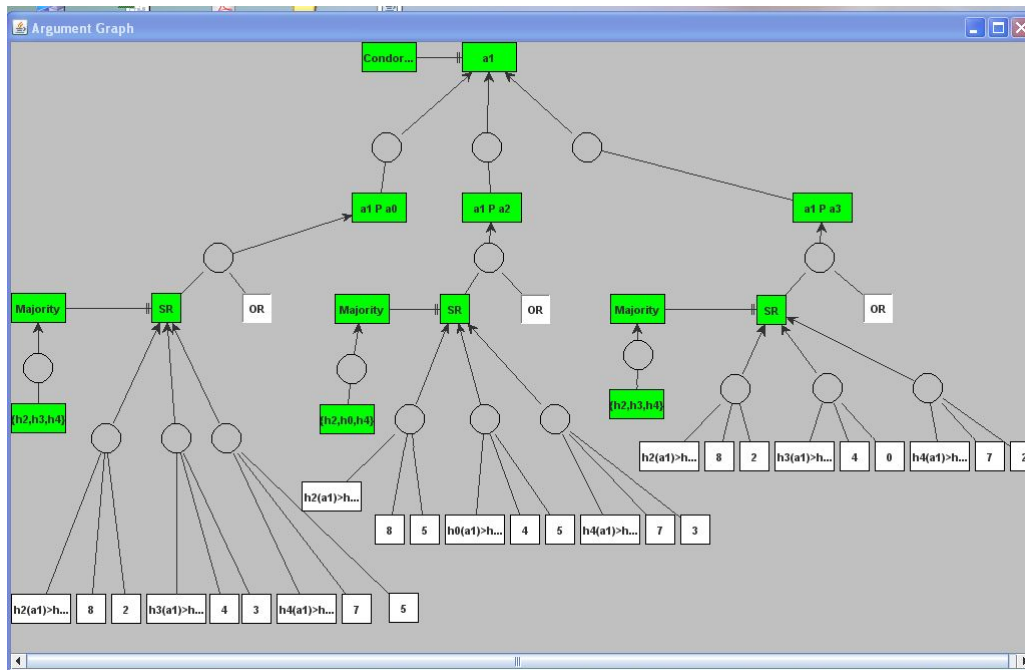


Figure 6.14 — Graph with the new minimal explanation

To represent such an explanation, the system updates the graph of arguments, by replacing the

argument on the criterion h_3 by the one on the criterion h_0 (see Figure 6.14). Thus, the final conclusion is again acceptable and warranted.

However, the user still not satisfied refutes the recommendation by a counter argument (see Figure 6.15). He states that he prefers another action, a_0 , as the best choice (see Figure 6.17). Such an information attacks the previous proposition of the system and therefore the solution is no longer warranted. The attack is the result of the critical question number $N^{\circ}1$: “Is there not a better alternative?”. In this case, the burden of proof shifts from the system to the user to provide the evidences for its proposition. This is due to the type of critical question, which is an exception.

The user replies that from his point of view, the action a_0 is better than a_1 according to the coalition $\{h_0 \text{ and } h_1\}$ (see Figure 6.18). Such an information causes many changes, as it is illustrated in the Figure 6.19. First, the user specifies explicitly the reasons that allow to support his choice. We note, that such reasons constructed concern only the comparison between a_0 and a_1 , and therefore only the graph corresponding to such a comparison is explicit. Indeed, the user reacts only on the comparison of these two alternatives and did not mention anything for the others. Second, we can remark that the user takes into account only two criteria in order to compare the two actions. This was not the case at the beginning of the process where the system used the majority principle. As a consequence, this information provides a way to contradict the use of the current proof.



Figure 6.15 — Example of a counter-argument

In other terms, *this not only attacks the previous recommendation of the system, but also attacks the proof-standard currently used*. Given this new information, the current proof-standard becomes a positive coalition consisting of $\{h_1, h_2\}$, enforcing the system to use this proof-standard. Hence, although a_1 continues to satisfy the *majority* proof-standard, the proof-standard itself is

not acceptable (it is colored in red in the Figure 6.19¹).

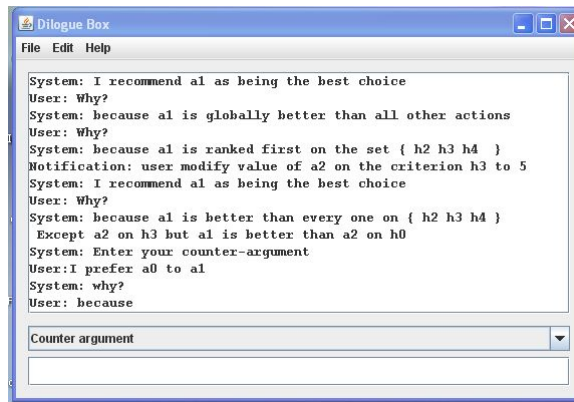


Figure 6.16 — Example of user’s attack

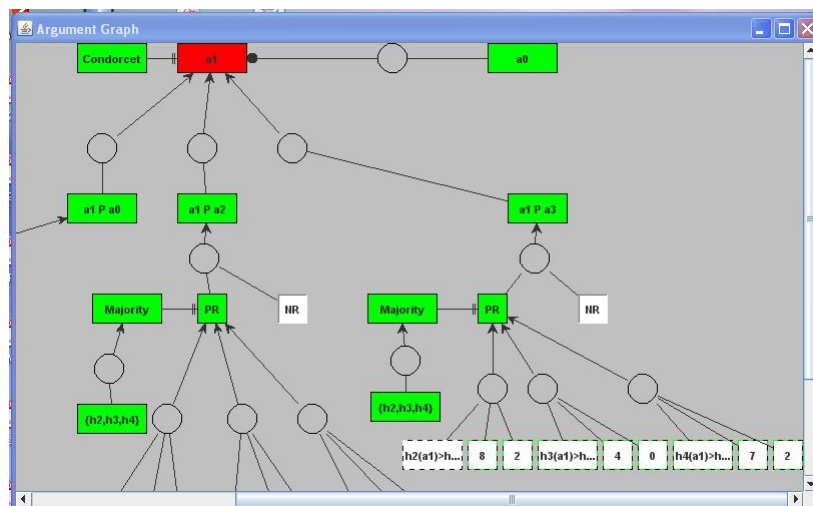


Figure 6.17 — User’s attack (graph)

In fact, such an information corresponds to a critical response providing a negative evidence against the property “anonymity” of the proof simple majority (see Table 5.5 in Section 5.4). Thus, the dialectical status of this property becomes rejected, and therefore does not support the use of the proof simple majority. According to our acceptability function, when the property is challenged, the proof is not acceptable. As a consequence, the system should change the proof standard considered until now. To account for that, the system will use the lattice (see Figure 5.4), in order to move from the current proof to the new one. Invalidating the anonymity leads to

¹We note that the elementary level in this figure is not deleted. For sake of clarity we represent just the part of the graph that we need to explain the dialogue.

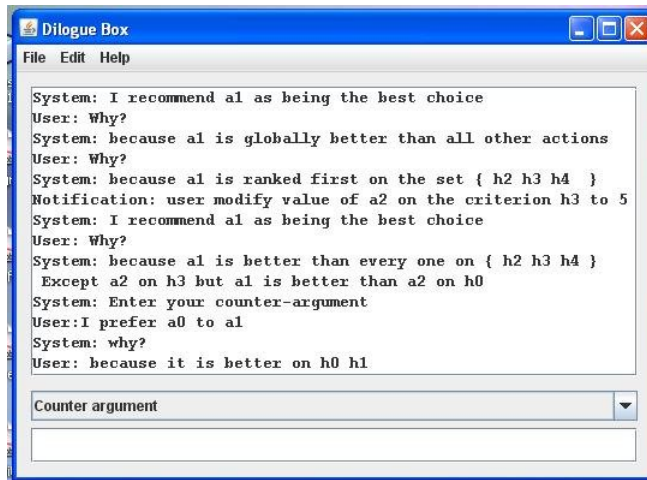


Figure 6.18 — User's critical response

the following combination (101000), which corresponds either to the lexicographical method or a weighted majority.

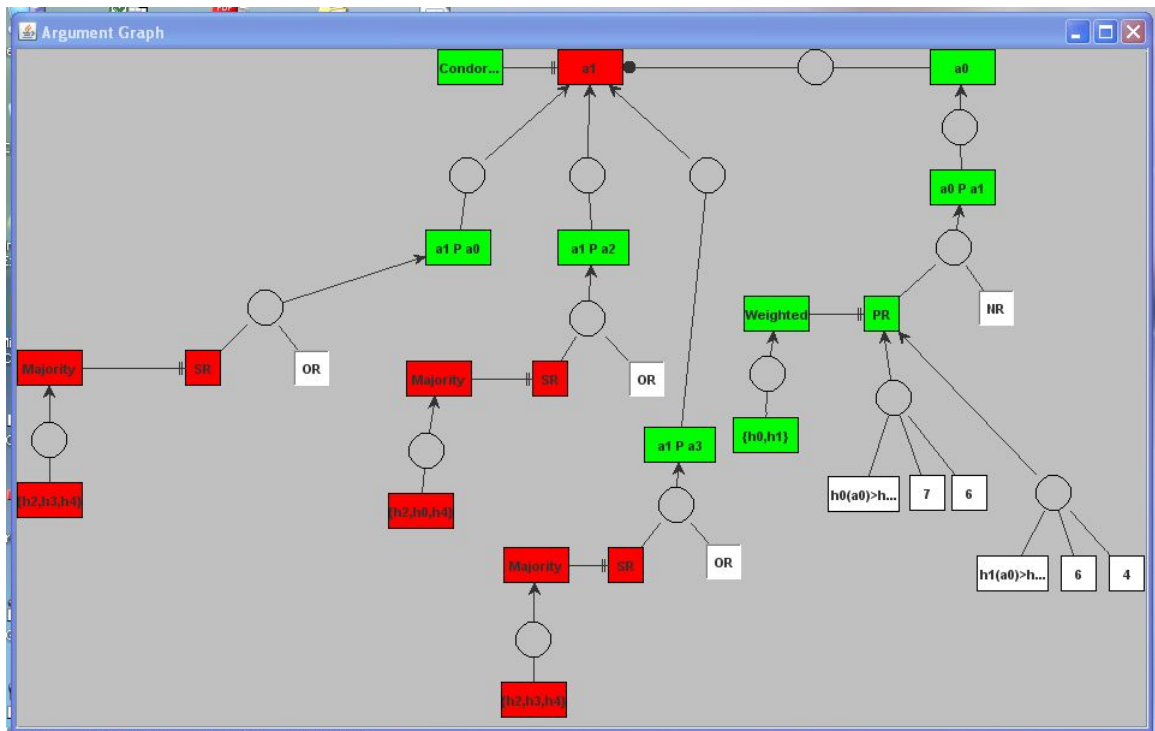


Figure 6.19 — Challenging the proof

However, in this case, the system cannot use the lexicographical method because there is no

explicit information on the existence of a linear order among the criteria which is the primary assumption to use such a procedure. Finally, the Table 6.2 presents examples of statements acceptability. For each statement we specify the proof associated, the premise type and the value of the holds function when the statement is a premise of an argument. We note that the premise “ a_0Pa_1 ” does not holds because the dialectical status of this statement is rejected, and for a rejected statement which is an ordinary premise the holds function takes the value false. In fact, the user specifies clearly that he prefers a_1 to a_0 , which is not the case for the other actions. However, the acceptability of such a statement is true because it depends on the defensibility of the argument pro, with the premises SR and OR . on the one hand, the SR premise has a dialectical status “accepted” and it is an assumption, then holds= true; and on the other hand there are not strong opposing reasons, then it is not taken into account in the acceptability computation. We note that the OR is depicted in the graph to represent the argument scheme for MC-PW-EV, but the corresponding boxe is coloured white to specify that in reality the system or the decision maker does not specify any negative reason.

statement	Proof	Premise type	argument's conclusion	holds	Acceptability
$\{h_2, h_3, h_4\}$	–	assumption	Majority	False	False
Majority	–	–	a_1Pa_0	–	False
SR	Majority	assumption	a_1Pa_0	True	False
Majority	–	–	a_1Pa_2	–	False
SR	Majority	assumption	a_1Pa_2	True	False
Majority	–	–	a_1Pa_3	–	False
SR	Majority	assumption	a_1Pa_3	True	False
Weighted	–	–	a_0Pa_1	–	True
SR	Weighted	assumption	a_0Pa_1	True	True
a_1Pa_0	–	ordinary	a_1	True	True
a_1Pa_2	–	ordinary	a_1	True	True
a_1Pa_3	–	ordinary	a_1	True	True
a_0Pa_1	–	ordinary	a_0	True	True
a_0	–	ordinary	a_1	True	True
a_1	Condorcet	–	–	–	False

Table 6.2 — Examples of statements acceptability (Figure 6.19)

At the next move of the dialogue, the system takes into account such changes and presents a new recommendation a_3 (see Figure 6.20). It is based on the fact that it beats all the other actions, and especially the solution suggested by the user, on the criteria h_0 and h_1 (see Figure 6.21). However, the user strongly rejects this solution. He justifies his rejection by the fact that a_3 has a very bad evaluation on the criterion h_3 (see Figure 6.20). In other terms, the user points the

existence of a strong negative reason against this action (see Figure 6.22). We specified that a conclusion of an argument scheme MC-PW-EV can be rejected, if it is possible to identify a strong opposing reasons that blocks such a conclusion. In this case, the negative reason corresponds to an absolute veto (argument scheme for UNICRITERIA INTRINSIC ACCEPTABILITY).

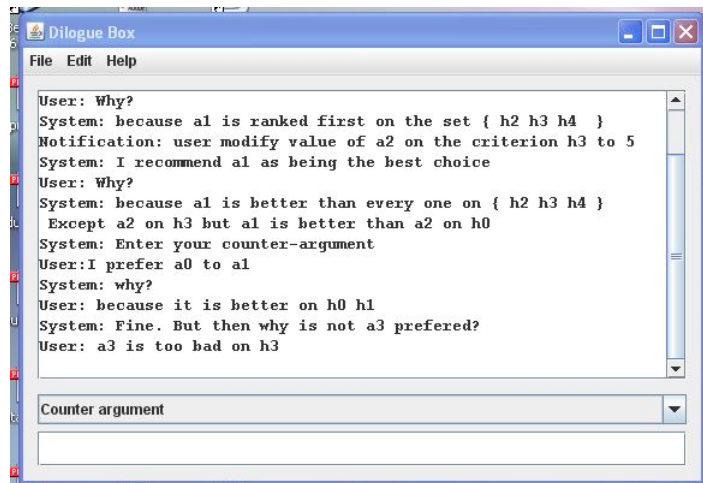


Figure 6.20 — Presenting a new recommendation

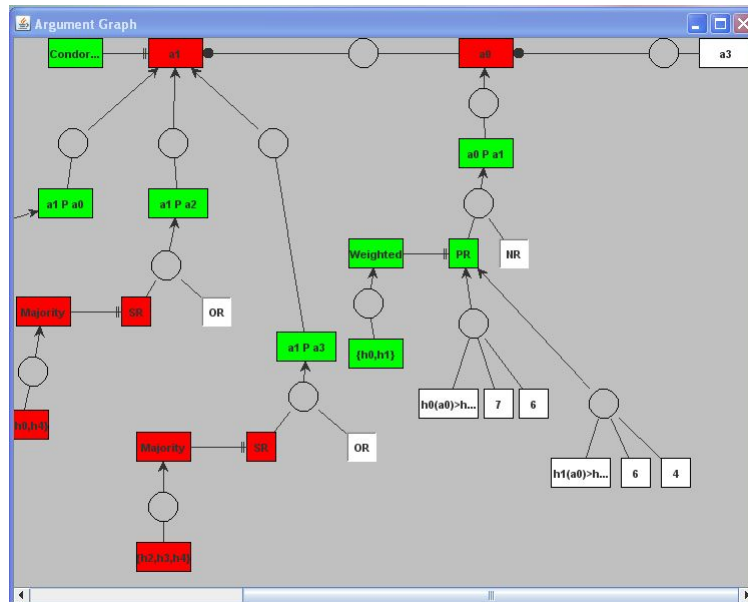


Figure 6.21 — Attack against the user's proposition

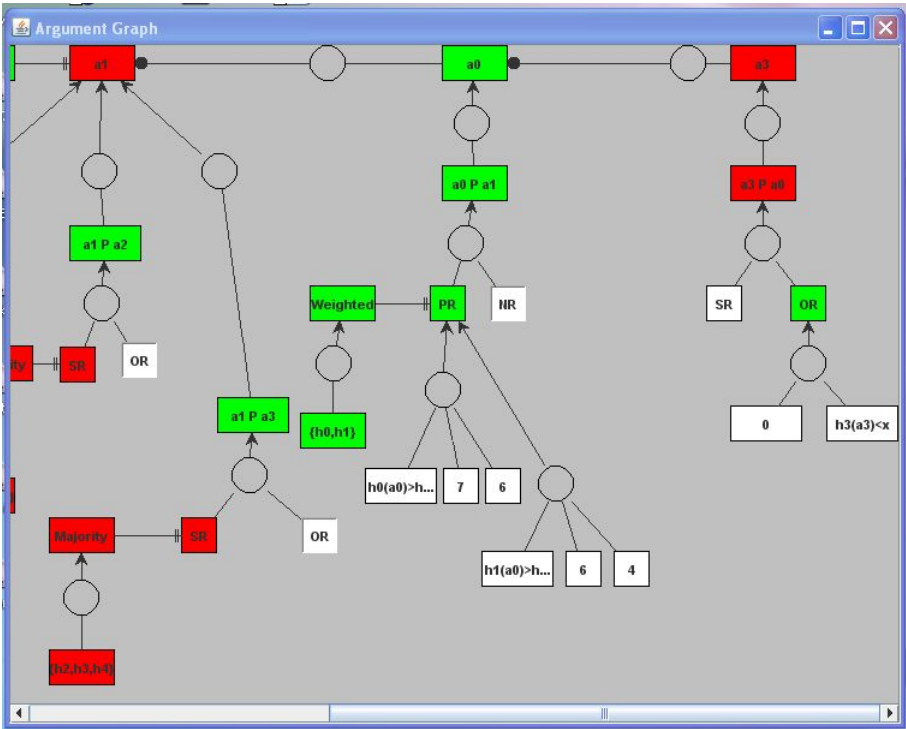


Figure 6.22 — representation of a strong negative reason

CONCLUSION

We started this thesis with the following questions: does it make sense to use argumentation within a decision aiding process? What conditions are necessary in order to get a model that is meaningful for a decision maker and for a decision analyst in a decision aiding situation? Does this idea present some advantages?

Following these motivations, we investigated the different ways to use argumentation in a decision context. Different approaches in the literature have contributed to greatly extend our understanding of the subject. However, such approaches do not necessarily meet all our expectations when we deal with decision aiding processes.

We proposed, therefore, to build our own model based on elements from argumentation theory and decision aiding methodology. Our contributions in this work are manifold, from the knowledge modeling exercise to the implementation of a prototype exhibiting the different features discussed in this thesis. More specifically, in the first part of the work we propose a hierarchical structure of argument schemes in order to represent the reasoning steps of a multiple criteria evaluation process. Such a structure 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. In the second part of the work, we initiated a systematic study of the use of argumentation in decision aiding tools. We introduced and studied a number of concepts in order to enhance the capability of a decision support to provide formal explanations and to inform the user of the consequences of modifications that occur during the interaction. To account for that we proposed to:

- ▷ choose and update a proof standard during the interaction by using a Concept Lattice. Such a lattice provides a mean to guide the system and to navigate among the different candidate proof standards, depending on the properties that are currently satisfied or contradicted.
- ▷ generate automatically minimal explanations to justify the recommendation of a choice problem statement (more specifically, we studied situations where a Condorcet winner exists). Different types of explanations were identified depending on the language used.

Such explanations are based on the different argument schemes constructed in this work.

- ▷ inform the user as to how the modifications that occur during the interaction affect the recommendation and the decision to be made. Such updates or revisions are the results of the intervention of the decision maker, through either a critical question or a critical response, in order to correct, refine or contradict previous assumptions. To handle such updates or revisions the system is endowed with different tools, namely an acceptability function to evaluate the exchanged arguments and an arguments graph to visualize such arguments.
- ▷ manage the dialogue between the system and the user, by constructing different rules describing a formal dialogue game. Such a dialogue game is based on recent extensions that incorporate argument schemes within dialogue games. Contrary to many dialogues games the rulee described in this work give different opportunities to the user and the system, illustrating the asymmetry of the situation.

However, there are still many open questions and problems which deserve further attention.

Further levels or structures of argument schemes?

We presented in this work, a hierarchical structure of argument schemes to represent explicitly the reasoning steps and the assumptions considered by the system during the evaluation process. For the sake of simplicity we did not consider the two first steps of the process, namely problem situation and problem formulation, in the construction of such a structure.

However, the construction of an evaluation model depends on the problem formulation which depends in turn on the situation problem. The following Example, which is a continuation of the example 1.15, illustrates an example of such a relation, between an evaluation model and a formulation problem.

Example 1. (*Example 1.15 Cont.*)

A possible reaction to this suggestion could be the following: “meanwhile we had a strategic discussion and the company considers that in reality the issue is to find a supplier with whom to establish a strategic partnerships considering the expansion of our activities.” Clearly, not only does the evaluation model makes no sense but the problem formulation has also to be revised. We now have:

- ▷ \mathbb{A} are potential suppliers;
- ▷ \mathbb{V} concern the suppliers reliability, market share, availability to strategic partnerships, quality record, etc.
- ▷ \mathbb{II} will now become a classification problem statement, the issues being to find out whether each supplier fits the company’s strategy.

A new evaluation model has to be built now in such a way that:

- ▷ *A are potential suppliers;*
- ▷ *D are the dimensions under which the suppliers are analysed (market share, quality certification, history of past supplies management structure, etc.);*
- ▷ *H are the criteria the client agrees to use in order to represent his preferences;*
- ▷ *there is no uncertainty;*
- ▷ *R could be a multiple criteria classification procedure.*

We can observe through the example above, that the intervention of the decision maker modifies not only the evaluation model but also the problem formulation. Thus, the revision process may concern one or several models of the decision aiding process. Moreover, we can note that in this case the explanation for challenging *R* is based on the fact that the formulation problem has changed. Currently, our model does not take into account such type of changes but it would be interesting to investigate how to handle revisions and updates at each step of the decision aiding process. Is it then possible to rely on argumentation to represent explicitly the link between the different models? In other terms, under what conditions can we generalize the model in order to manage, to represent and to explain the changes and interactions between the different outcomes and phases of the decision aiding process? A preliminary work in this direction is the one of [Dimopoulos *et al.*, 2004].

At the same time, we considered in this work the context of a multiple criteria and a specific type of problem which is the choice problem statement. Therefore, the hierarchical structure was constructed on the basis of such restrictions. There are already different ways or means to model a problem situation (linear programming for an optimisation problem, for instance). Finally, as already mentioned, there exist several other types of problem statements (e.g. sorting, clustering, etc.). Thus, our approach should be used or adapted to meet different problem situations.

Generating and evaluating explanations

We have seen through different examples presented in this document that the aim of an explanation is not just to justify a statement or a recommendation, it is also used to help the decision maker to identify relevant questions and possible critics. Indeed, confronted with an explanation, a decision maker can materialize the reasoning followed by the system to get the result and identify the elements that are wrong or missing. Moreover, the construction of an explanation is not just the process of constructing a set of arguments and to present them to the decision maker. An explanation should be intuitive, simple and natural in order to be convincing and accepted by the

decision maker. Thus, we believe that two aspects are important in developing an explanation: (i) how to build an explanation (i.e., what are the elements that should be explicitly mentioned in the explanation and how to present them?) and (ii) how to evaluate the impact of an explanation on the decision maker?

In this work we addressed the first issue by discussing the problem of constructing an explanation for a global recommendation for a choice problem statement, i.e. a Condorcet winner. We presented three different categories of explanations based on three different building blocks (simple, factored and generic one), which relates to different argument schemes. However, in practice, one can suppose different situations:

- ▷ there is not a Condorcet winner but it is possible to identify an action that is better than another one according to some other procedure;
- ▷ there is not a single recommendation but a set of actions to be recommended;
- ▷ the problem statement is not a choice problem.

When a Condorcet winner does not exist we should use a different procedure to get a result. For instance, as we briefly presented in this document, the Copeland rule, which is an extension of the Condorcet winner, allowing to select the action with the largest Copeland score. There are various procedures presented in the literature, such as procedures based on the Kernel, procedure based on the covering relation, etc. Thus, what kind of explanations can we have in such cases? Is it possible to define any notion of minimality?

Let us consider, briefly, the example of the procedures based on the covering relation. In short, an alternative can be recommended either because it is not outranked², or because even if it is outranked, it can be saved because against some third alternative it is arguably better than the alternative that outranks it. Thus, we can distinguish the existence of *two types of evidences*, direct evidence and indirect one. Direct evidence is easier to provide and explain, so it should be favoured: in other words, when possible, the explanation should rely on direct evidence, and refer to indirect evidence only when necessary.

With these covering procedures, an alternative is not recommended because there exists another alternative outranking it. So answering the question “Why is this alternative x not recommended?” should be easy: it suffices to provide the example of an alternative that is better than x : there necessarily exists such an alternative. However, being directly outranked is not a sufficient condition, so there is an implicit part in the reply.

“ x is not recommended because y outranks x ”

(*implicit: and there no indirect evidence of x being preferred to y*).

²see subsection 1.4.1.1

The question “why is this alternative x recommended?” is slightly more difficult, since there are two possible cases: either x is not outranked and the response is obvious

“ x is recommended because no y outranks x ”

but it can also be that x is outranked and that x is nevertheless considered because of its comparison to third alternatives. In this case, the explanation *must* provide the indirect evidence

“ x is recommended because even though y outranks x

x is strictly preferred to z , while y is indifferent or incomparable to z .

So relatively to z , x is arguably better than y .

The principle underlying the covering method suggests that this idea can be extended, in spirit to argumentation games. Very broadly speaking one can see the direct and indirect evidence as different types of arguments (supporting argument, counter argument) exchanged in a sort of a dialectical game. What do we get then at the end? Is it interesting?

Second, such procedures do not necessarily provide a unique choice (or a unique recommendation). In fact, the recommendation of a choice problem may be a set of actions that are equivalent and difficult to distinguish between them. Therefore, what kind of explanation should we provide in this case? In other terms, how to answer such questions: “Why is this set recommended” ? or “Why is this action x not in the recommended set?” etc. Finally, the problem statement is not always a choice problem. It can be for instance a ranking problem, where the result is a weak order among the set of alternatives. What should we explain in this case: all the ranking and the position of each action or just the best actions at the top of the ranking?

On the other hand, the construction of the explanations, in our context, is based on information provided by the user. In our case, we considered a simple numerical representation under the form of performance table and very simple preference structures. However, in practice, it is not always possible to represent the decision maker’s preferences under such a form. Thus, in general, is it possible to identify categories of explanations according to the initial set of information provided by the user?

Concerning the aspect of evaluating the impact of explanations it would be interesting to conduct experiments with real users to see what kind of explanations they prefer. We have proposed in this work different types of justifications depending on different languages and based on notions of minimality on an intuitive idea. But we have no way to decide if actually a user will prefer an explanation with a generic statement because it is concise or a detailed explanation (as with simple statements) because he needs the details to understand the solution. In other terms, understanding the behaviour of a user faced to a given justification, will help in developing strategies to

construct formal explanations. Thus, we can imagine to have an implemented system, realised in the form of an interactive web-based discussion forum, where the user is asked to give his preferences on different types of explanations generated automatically for a given issue. The idea is to measure the satisfaction degrees of the user and try to understand and to analyse what kind of relation we can have between the content and the shape of the explanation and such a degree. We can also imagine that such a study will allow to identify some criteria or indicators that can be useful to classify the explanations. Of course, ideally an argument should be constructed with respect to a given audience. [Hunter, 2004] discussed the impact of arguments by showing how we can use empathy and antipathy to define a pre-ordering relation over arguments in order to capture how an argument is “more believable” than another for a given audience.

Choosing an aggregation procedure

We discussed in this work the problem of choosing an aggregation procedure in order to resolve a decision problem. We proposed to use a Concept Lattice where the nodes represent the different procedures considered at the beginning of the process and the links the fact that a property of a given procedure is satisfied or contradicted.

However, in our proposal we have considered a limited and specified number of procedures. Thus, the lattice does not take into account other classes of methods or procedures such as sorting method, clustering method, etc. Moreover, we propose to navigate among the different nodes on the basis of the responses if the decision maker during the interaction. For such a purpose we presented some examples of kind of responses, called critical responses. Such responses offer a way to identify what property is challenged and / or which property should be taken into account. Such examples were constructed by respecting the theory and concepts of decision aiding methodology. However, we believe that an experimental study aiming at analyzing the behavior of the decision maker in a situation of decision support, would probably confirm such responses and / or allows to identify other types of reactions more realistic and practical. Such a study could also serve as a validation of the properties identified in this work and to identify other natural features of decision maker’s preferences which we have not thought of.

Towards real applications

We presented in this thesis a first contribution towards the implementation of a system with the aim to (i) present a recommendation that can be explicitly justified; (ii) revise any piece of reasoning involved in this process, and be informed of the consequences of such moves; and (iii) stimulate the user by representing naturally (sometimes visually). Our main goal was to show that decision aiding is more than simply solving a complex decision model more or less faithful of the decision maker’s values and preferences. It involves explaining, convincing, revising and

updating the outcomes of what we call a decision aiding process.

To show this, we presented an example of dialogue between the system and the decision-maker, considering different reactions and situations of disagreement. Subsequently, we believe that it would be interesting to test our system with real users and different problem situations. The idea is to have their views on the usefulness, benefits and / or disadvantages of having a system that is able to justify solutions and take into account their responses to update information and solutions.

The Decision Deck project offers a good opportunity to make a significant step towards the integration of our concepts to existing system. This project aims at developing a generic decision aid platform composed of modular and interconnected software components. These software components implement the common functionalities of a large range of multiple criteria decision aid methods. Its purpose is to provide effective tools for three types of “users”: (i) practitioners who use MCDA tools to support actual decision makers involved in real world decision problems; (ii) teachers who present MCDA methods in courses, for didactic purposes; and (iii) researchers who want to test and compare methods or to develop new ones. Thus, this projet will offer us a first way to present our approach and to test it by different users categories [Bisdorff *et al.*]. Being confronted to real users will allow us to improve our system and to take into account other aspects and fonctionnalités, such as processing and analysing natural language.

Finally, our system is based on a dialogue game that is more flexible than most dialogue games presented in the literature but not enough to simulate a real interaction between a decision analyst and a decision maker. Indeed, it is very difficult to imagine all the user’s reactions and the different moves that we can allow without experimentation. Thus, on the basis of experiments, we believe that the dialogue game can be improved with new rules more flexible and more general, thereby covering a wide range of dialogues within decision aiding situations. At the same time, such a dialogue can be extended to account for situations involving several decision makers.

Bibliography

- L. Amgoud and C. Cayrol. Inferring from inconsistency in preference-based argumentation frameworks . *International Journal of Automated Reasoning*, 29(2):125–169, 2002.
- L. Amgoud and H. Prade. Explaining qualitative decision under uncertainty by argumentation. In G. Yolanda and R. Moo, editors, *Proceedings of the 21st Conference on Artificial Intelligence (AAAI'06)*, pages 219–224, 2006.
- L. Amgoud and H. Prade. Using arguments for making and explaining decisions. *Journal of Artificial Intelligence*, 173:413–436, 2009.
- L. Amgoud, N. Maudet, and S. Parsons. Modelling dialogues using argumentation. In E. Durfee, editor, *Proceedings of the 4th International Conference on Multi-Agent Systems (ICMAS'00)*, pages 31–38. IEEE Press, 2000.
- L. Amgoud, S. Parsons, and N. Maudet. Arguments, dialogue, and negotiation. In W. Horn, editor, *Proceedings of the European Conference on Artificial Intelligence (ECAI'00)*, pages 338–342. IOS Press, 2000.
- L. Amgoud, C. Cayrol, and M.-C. Lagasquie Schiex. On the bipolarity in argumentation frameworks. In J.P. Delgrande and T. Schaub, editors, *Proceedings of 10th International Workshop on Non-Monotonic Reasoning (NMR'04)*, pages 1–9, 2004.
- L. Amgoud, J.-F. Bonnefon, and H. Prade. An argumentation-based approach to multiple criteria decision. In L. Godo, editor, *Proceedings of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'05)*, pages 269–280, 2005.
- L. Amgoud. Argumentation for decision making. In I. Rahwan and G. Simari, editors, *Argumentation in Artificial Intelligence*. Springer, 2009. To appear.
- Aristotle. *On Sophistical Refutations*. Harvard University Press, Loeb Classical Library, Cambridge, Mass., 1928.

- Aristotle. *Topics*. Harvard University Press, Loeb Classical Library, Cambridge, Mass., 1939. trans. John Henry Freese.
- K. Atkinson, T. Bench-capon, and P. Mcburney. Justifying practical reasoning. In C. Reed F. Grasso and G. Carenini, editors, *Proceedings of the 4th International Workshop on Computational Models of Natural Argument (CMNA'04)*, pages 87–90, 2004.
- K. Atkinson, T. Bench-Capon, and P. Mcburney. A dialogue game protocol for multi-agent argument over proposals for action. *Autonomous Agents and Multi-Agent Systems*, 11(2):153–171, 2005.
- K. Atkinson, T.J.M. Bench-Capon, and S. Modgil. Argumentation for decision support. In S. Bressan, J. Küng, and R. Wagner, editors, *Proceedings of the 17th International Conference on Database and Expert Systems Applications*, pages 822–831, 2006.
- K. Atkinson. *What Should We Do? : Computational Representation of Persuasive argument in practical reasoning*. PhD thesis, Department of Computer Science, 2005.
- K. Atkinson. Value-based argumentation for democratic support. In P.E Dunne and T.J.M. Bench-Capon, editors, *Proceedings of the 1st International Conference on Computational Models of Natural Argument (COMMA'06)*, pages 47–58, 2006.
- C.A. Bana e Costa and J.C.. Vansnick. Macbeth - an interactive path towards the construction of cardinal value fonctions. *International Transactions in Operational Research*, 1:489–500, 1994.
- C. Bana e Costa. Les problématiques de l'aide à la décision : vers l'enrichissement de la trilogie choix-tri-rangement. *Recherche Operationnelle*, 30(2):191–216, 1996.
- V. Belton and J. Pictet. Talking about the practice of MCDA. In D. Bouyssou, E. Jacquet-Lagrèze, P. Perny, R. Slowinski, D. Vanderpooten, and Ph. Vincke, editors, *Aiding Decisions with Multiple Criteria: Essays in Honour of Bernard Roy*, pages 71–88. Kluwer Academic, Dordrecht, 2001.
- V. Belton and T. Stewart. *Muliple Criteria Decision Analysis: An Integrated Approach*. Kluwer Academic, Dordrecht, 2002.
- T.J.M Bench-Capon and H. Prakken. Argumentation. In A.R. Lodder and A. Oskamp, editors, *Information Technology & Lawyers : Advanced technology in the legal domain from challenges to darlyroutine*, pages 61–80. Springer Verlag, 2005.
- T.J.M. Bench-Capon and H. Prakken. Justifying Actions by Accruing Arguments. In P. E. Dunne and T. J. M. Bench-Capon, editors, *Proceedings of the 1st International Conference on Computational Models of Argument (COMMA'06)*, pages 247–258, 2006.

- S. Benferhat, D. Dubois, and H. Prade. Argumentative inference in uncertain and inconsistent knowledge base. In D. Heckerman and E. H. Mamdani, editors, *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence (UAI'93)*, pages 411–419, 1993.
- R. Bisdorff, V. Mousseau, and P. Meyer. Decision deck project. <http://www.decision-deck.org/>.
- A. Blum, J. Jackson, T. Sandholm, and M. Zinkevich. Preference elicitation and query learning. *Journal of Machine Learning Research*, 5:649–667, 2004.
- B. Bonet and H. Geffner. Arguing for Decisions: A Qualitative Model of Decision Making. In E. Horvitz and F. V. Jensen, editors, *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence (UAI'96)*, pages 98–105, 1996.
- J.-F. Bonnefon and H. Fargier. Comparing sets of positive and negative arguments : Empirical assessment of seven qualitative rules. In G. Brewka, editor, *Proceedings of 17th European Conference on Artificial Intelligence (ECAI'06)*, pages 16–20, 2006.
- D. Bouyssou and Ph. Vincke. Ranking alternatives on the basis of preference relations: a progress report with special emphasis on outranking relations. *Journal of Multiple Criteria Decision Analysis*, 6:77–85, 1997.
- D. Bouyssou, Th. Marchant, M. Pirlot, P. Perny, A. Tsoukiàs, and Ph. Vincke. *Evaluation and decision models: a critical perspective*. Kluwer Academic, Dordrecht, 2000.
- D. Bouyssou, Th. Marchant, M. Pirlot, A. Tsoukiàs, and Ph. Vincke. *Evaluation and decision models with multiple criteria: Stepping stones for the analyst*. Springer Verlag, Boston, 2006.
- D. Bouyssou. Building criteria: A prerequisite for MCDA. In C.A. Bana e Costa, editor, *Readings in Multiple Criteria Decision Aid*, pages 58–80. Springer-Verlag, 1990.
- R. Brafman and C. Domshlak. Representing, eliciting, and reasoning with preferences aaai-07 tutorial forum, 2007. <http://www.cs.bgu.ac.il/brafman/tutorial-slides.pdf>.
- R.V. Brown. Toward a prescriptive science and technology of decision aiding. *Annals of Operations Research*, 19:467–483, 1989.
- S. Carberry, J. Chu-Carroll, and S. Elzer. Constructing and utilizing a model of user preferences in collaborative consultation dialogues. *Computational Intelligence*, 15(3):185–217, 1999.
- G. Carenini and J. D. Moore. A strategy for generating evaluative arguments. In *Proceedings of the 1st international conference on Natural language generation (INLG'00)*, pages 47–54, 2000.

- G. Carenini and J. D. Moore. Generating and evaluating evaluative arguments. *Artificial Intelligence*, 170(11):925–952, 2006.
- L. Carlson. *Dialogue Games: An Approach to Discourse Analysis*. Reidel, 1983.
- C. Cayrol and M.-C. Lagasque Schiex. On the acceptability of arguments in bipolar argumentation frameworks. In L. Godo, editor, *Proceedings of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'05)*, pages 378–389, 2005.
- L. Chen and P. Pu. Preference-based organization interfaces: Aiding user critiques in recommender systems. In C. Conati, K. F. McCoy, and G. Paliouras, editors, *Proceedings of 11th International Conference on User Modeling (UM'07)*, pages 77–86, 2007.
- C.I. Chesñevar, A.G. Maguitman, and R.P. Loui. Logical models of argument. *ACM Computing Surveys*, 32(4):337–383, 2000.
- M. Dastani, J. Hulstijn, and L.W.N. van der Torre. How to decide what to do? *European Journal of Operational Research*, 160(3):762–784, 2005.
- G. Debreu. Topological methods in cardinal utility theory. In K.J. Arrow, S. Karlin, and P. Suppes, editors, *Mathematical Methods in the Social Sciences*, pages 16–26. Stanford University Press, 1960.
- Luis C. Dias and Vincent Mousseau. Inferring electre's veto-related parameters from outranking examples. *European Journal of Operational Research*, 170:172–191, 2006.
- Y. Dimopoulos, P. Moraitis, and A. Tsoukiàs. Argumentation based modeling of decision aiding for autonomous agents. In *Proceedings of IEEE/WIC/ACM International Conference on the Intelligent Agent Technology*, pages 99–105. IEEE press, 2004.
- J. Doyle and R.H Thomason. Background to qualitative decision theory. *AI Magazine*, 20:55–68, 1999.
- P. M. Dung. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.
- W. Edwards. Social utilities. *Engineering Economist*, 6:119–129, 1971.
- J. Figueira, S. Greco, and M. Ehrgott. *Multiple Criteria Decision Analysis: State of the Art Surveys*. Springer Verlag, 2005.

- J. Fox and S. Parsons. On Using Arguments for Reasoning about Actions and Values. In *Proceedings of the AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning*, pages 55–63. AAAI Press, 1997.
- J. Fox and S. Parsons. Arguing about beliefs and actions. In A. Hunter and S. Parsons, editors, *Applications of Uncertainty Formalisms*, pages 266–302, 1998.
- J. Fox, D. Barber, and K. D. Bardhan. Alternatives to Bayes? A quantitative comparison with rule-based diagnostic inference. *Methods of Information in Medicine*, 19(4):210–215, 1980.
- L.A. Franco, D. Shaw, and M. Westcombe. Special issue: Problem structuring methods. *Journal of the Operational Research Society*, 57:757 – 883, 2006.
- B. Ganter and R. Willer. *Formal concept Analysis*. Springer, 1999.
- T.F. Gordon and D. Walton. The Carneades argumentation framework. using presumptions and exceptions to model critical questions. In P.E Dunne and T.J.M. Bench-Capon, editors, *Proceedings of the 1st International Conference on Computational Models of Natural Argument (COMMA'06)*, pages 195–207, 2006.
- T.F. Gordon and D. Walton. Proof burdens and standards. In I. Rahwan and G. Simari, editors, *Argumentation in Artificial Intelligence*. Springer-verlag, 2009. To appear.
- T. Gordon, H. Prakken, and D. Walton. The Carneades model of argument and burden of proof. *Artificial Intelligence*, 171(4):875–896, 2007.
- S. Greco, V. Mousseau, and R. Slowinski. Ordinal regression revisited: multiple criteria ranking using a set of additive value functions. *European Journal of Operational Research*, 191:415–435, 2008.
- J.B. Grize. Matériaux pour une logique naturelles. Travaux de recherche du Centre de recherche Sociologique. Université de Neuchatel, Suisse, 1976.
- V. Ha and P. Haddawy. Similarity of personal preferences: theoretical foundations and empirical analysis. *Artificial Intelligence*, 146:149 – 173, 2003.
- P. Haddaway, V. Ha, A. Restificar, B. Geisler, and J. Miyamoto. Preference elicitation via theory refinement. *Journal of Machine Learning Research*, 4:317 – 337, 2003.
- A.C. Hastings. *A Reformulation of the Modes of Reasoning in Argumentation*. PhD thesis, Evanston, Illinois, 1963.
- D. Henriet. The Copeland choice function: an axiomatic characterization. *Social choice and Welfare*, 2:49–64, 1985.

- A. Hunter. Making argumentation more believable. In *Proceedings of the 19th national conference on Artificial Intelligence (AAAI'04)*, pages 269–274, 2004.
- E. Jacquet-Lagrèze and Y. Siskos. Preference disaggregation: 20 years of MCDA experience. *European Journal of Operational Research*, 130:233–245, 2001.
- A. C. Kakas and P. Moraitis. Argumentation based decision making for autonomous agents. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'03)*, pages 883–890. ACM, 2003.
- A. C. Kakas, P. Mancarella, and Phan Minh Dung. The acceptability semantics for logic programs. In P. Van Hentenryck, editor, *Proceedings of the 11th international conference on Logic programming*, pages 504–519, 1994.
- N. I. Karacapilidis and D. Papadias. A group decision and negotiation support system for argumentation based reasoning. In G. Antoniou, A. Ghose, and M. Truszczynski, editors, *Proceedings of the Workshop on Reasoning with Incomplete and Changing Information and on Inducing Complex Representations (PRICAI'96)*, pages 188–205, 1998.
- N. I. Karacapilidis and D. Papadias. Hermes: Supporting argumentative discourse in multi-agent decision making. In *Proceedings of the 15th National Conference on Artificial Intelligence and 10th Innovative Applications of Artificial Intelligence Conference*, pages 827–832. AAAI Press, 1998.
- N. I. Karacapilidis and D. Papadias. Computer supported argumentation and collaborative decision making: the HERMES system. *Information System*, 26:259–277, 2001.
- H. Katsuno and A. O. Mendelzon. On the difference between updating a knowledge base and revising it. In J.F. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pages 387–394, 1991.
- J. Katzav and C. Reed. On argumentation schemes and the natural classification of arguments. *Argumentation*, 18(2):239–259, 2004.
- R.L. Keeney and H. Raiffa. *Decisions with multiple objectives: Preferences and value tradeoffs*. J. Wiley, New York, 1976.
- M. Kienpointer. Toward a typology of argument schemes. In *Proceedings of the international Conference of the Society of the Study of Argument (ISSA'86)*. Amsterdam University Press, 1986.

- P.A. Kirschner, J. Simon, and S. Chad, editors. *Visualizing argumentation: software tools for collaborative and educational sense-making*. Springer-Verlag, 2003.
- D. A. Klein. *Decision-analytic intelligent systems: automated explanation and knowledge acquisition*. L. Erlbaum Associates Inc., 1994.
- D.H. Krantz, R.D. Luce, P. Suppes, and A. Tversky. *Foundations of measurement*. Academic Press, New York, 1971.
- C. Labreuche. Argumentation of the results of a multi-criteria evaluation model in individual and group decision aiding. In E. Montseny and P. Sobrevilla, editors, *Proceedings of the 4th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT)*, pages 482–487, 2005.
- C. Labreuche. Argumentation of the decision made by several aggregation operators based on weights. In *Proceedings of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'06)*, pages 683–690, 2006.
- J.-F. Laslier. *Tournament Solutions and Majority Voting*. Springer Verlag, Berlin, 1997.
- R.P. Loui. Hart's critics on defeasible concepts and ascriptivism. In *Proceedings of the 5th international conference on Artificial intelligence and law (ICAIL'95)*, pages 21–30. ACM, 1995.
- R.D. Luce and J.W. Tukey. Simultaneous conjoint measurement: A new type of fundamental measurement. *Journal of Mathematical Psychology*, 1:1–27, 1964.
- R.D. Luce. Semiorders and a theory of utility discrimination. *Econometrica*, 24:178–191, 1956.
- M.I. Madakkatel, I. Rahwan, J.F. Bonnefon, R. N. Awan, and S. Abdallah. Formal argumentation and human reasoning: The case of reinstatement. In *Proceedings of the AAI Symposium on the Uses of Computational Argumentation*, 2009.
- L.Y. Maystre, J. Pictet, and J. Simos. *Méthodes multicritères ELECTRE. Description, conseils pratiques et cas d'application à la gestion environnementale*. Presses Polytechniques et Universitaires Romandes, Lausanne, 1994.
- P. McBurney and S. Parsons. Dialogue game protocols. *Agent Communication Languages*, 2650:269–283, 2003.
- P. McBurney and S. Parsons. Dialogue games for agent argumentation. In I. Rahwan and G. Simari, editors, *Argumentation in Artificial Intelligence*. Springer-Verlag, 2009. To appear.

- J.L. McCarthy. Epistemological problems of artificial intelligence. In R. Reddy, editor, *Proceedings of the 5th International Joint Conference on Artificial Intelligence(IJCAI'77)*, pages 1038–1044, 1977.
- D. McDermott and J. Doyle. Nonmonotonic logic 1. *Artificial Intelligence*, 13:41–72, 1980.
- R.C. Moore. Semantical consideration on nonmonotonic logic. *Artificial Intelligence*, 25:75–94, 1985.
- M. Morge. The hedgehog and the fox. an argumentation-based decision support system. In *Proceedings of the 4th International Workshop on Argumentation in Multi-Agent Systems (ArgMas'07)*, pages 55–68, 2007.
- K. Morik. User models and conversational settings: Modeling the user's wants. In A. Kobsa and W. Wahlster, editors, *User Models in Dialog Systems*, pages 364–385. Springer, 1989.
- H. Moulin. *Axioms of Cooperative Decision Making (Econometric Society Monographs)*. Cambridge University Press, 1991.
- T.J. Norman, D.V. Carbogim, E.C. Krabb, and D. Walton. Argument and Multi-Agent Systems. In C. Reed and T.J. Norman, editors, *Argumentation Machines: New Frontiers in Argument and Computation*, chapter 2, pages 15–54. Kluwer Academic, 2003.
- W. Ouerdane, N. Maudet, and A. Tsoukiàs. Arguing over actions that involve multiple criteria: A critical review. In Khaled Mellouli, editor, *Proceedings of the 9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'07)*, pages 308–319, 2007.
- W. Ouerdane, N. Maudet, and A. Tsoukiàs. Argument schemes and critical questions for decision aiding process. In Ph. Besnard, S. Doutre, and A. Hunter, editors, *Proceedings of the 2nd International Conference on Computational Models of Argument(COMMA'08)*, pages 285–296, 2008.
- W. Ouerdane, N. Maudet, and A. Tsoukiàs. Argumentation Theory and Decision Aiding. In J. Figueira, S. Greco, and M. Ehrgott, editors, *New Trends in MCDA*. Springer-Verlag, 2009.
- W. Ouerdane, N. Maudet, and A. Tsoukiàs. Discussing proof-standards in the context of multiple criteria decision-aiding. The IJCAI-09 workshop on Computational Models of Natural Argument (CMNA'09), 2009.
- W. Ouerdane. Représentation des schémas d'argument dans un contexte d'aide multicritère à la décision. In N. Maudet, P. Schobbens, and M. Guyomrd, editors, *Proceedings of the Cinquième Journées Francophones Modèles Formels d'Interaction*, pages 255–265, 2009.

- M. Oztürk, A. Tsoukiàs, and Ph. Vincke. Preference modelling. In J. Figueira, S. Greco, and M. Ehrgott, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 27–72. Springer Verlag, Boston, Dordrecht, London, 2005.
- S. Parsons and N. Jennings. Argumentation and multi-agent decision making. In *Proceedings of the AAAI Spring Symposium on Interactive and Mixed-Initiative Decision Making*, pages 89–91, 1998.
- S. Parsons and P. McBurney. Argumentation-based dialogue for argment coordination. *Group Decision and Negotiation*, 12:415–439, 2003.
- S. Parsons, C. Sierra, and N. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.
- C. Perelman and L. Olbrechts-Tyteca. *The new Rhetoric: A treatise on argumentation*. University of Notre Dame Press, Notre Dame, London, 1969.
- J.L. Pollock. Defeasible reasoning. *Cognitive Science*, 11:481–518, 1987.
- J.L. Pollock. *Cognitive Carpentry : A Blueprint for how to Buid a Person*. The MIT Press, Cambridge (Massachusetts), 1995.
- H. Prakken and G. Sartor. The role of logic in computational models of legal argument: A critical survey. In A. C. Kakas and F. Sadri, editors, *Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski, Part II*, pages 342–381, 2002.
- H. Prakken and G. Vreeswijk. Logical systems for defeasible argumentation. In D. Gabbay and F. Guentner, editors, *Handbook of Philosophical Logic, second edition*, volume 4, pages 219–318, 2002.
- Henry Prakken. Relating protocols for dynamic dispute with logics for defeasible argumentation. *Synthese*, 127:187–219, 2001.
- H. Prakken. Analysing reasoning about evidence with formal models of argumentation. *Law, Probability and Risk*, 3:33–50, 2004.
- H. Prakken. Coherence and flexibility in dialogue games for argumentation. *Journal of Logic and Computation*, 15(6):1009–1040, 2005.
- H. Prakken. A study of accrual of arguments, with applications to evidential reasoning. In *Proceedings of the 10th International Conference on Artificial intelligence and law (ICAIL'05)*, pages 85–94. ACM, 2005.

- H. Prakken. Formal systems for persuasion dialogue. *Knowledge Engineering Review*, 21(2):163–188, 2006.
- P. Pu and L. Chen. Trust-inspiring explanation interfaces for recommender systems. *Knowledge-Based Systems*, 20(6):542–556, 2007.
- H. Raiffa. Preferences for multi-attributed consequences. Technical report, RM-5868-DOT, The RAND Corporation, Santa Monica, California, 1969.
- C. Reed and T.J. Norman. *Argumentation Machines : New Frontiers in Argument and Computation*. Kluwer Academic, 2003.
- C. Reed and G. Rowe. Araucaria: Software for argument analysis, diagramming and representation. *International Journal of AI Tools*, 14:2004, 2004.
- C. Reed and D. Walton. Application of argumentation schemes. In H.V. Hnasen, C.W. Tindale, J.A. Blair, and R.H. Johnson, editors, *Conference of the Ontario Society for the Study of Argument (OSSA2001)*, 2001. CD-ROM.
- C. Reed and D. Walton. Towards a formal and implemented model of argumentation schemes in agent communication. *Autonomous Agents and Multi-Agent Systems*, 11(2):173–188, 2005.
- C. Reed and D. Walton. Argument schemes in dialogue. In H.V. Hansen, C.W. Tindale, R.H. Johnson, and J.A. Blair, editors, *Dissensus and the search for common ground*, 2007. CD-ROM.
- R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 67(1,2):81–132, 1980.
- R. Reiter. Nonmonotonic reasoning. *Annual Review of Computer Science*, 2:147–186, 1987.
- N. Rescher. *Introduction to value theory*. Prentice Hall, Englewood Cliffs, 1969.
- R. J. Rienks, D. K. J. Heylen, and E. van der Weijden. Argument diagramming of meeting conversations. In A. Vinciarelli and J-M. Odobez, editors, *In proceedings of the Multimodal Multiparty Meeting Processing Workshop at the 7th International Conference on Multimodal Interfaces*, pages 85–92, 2005.
- F.S. Roberts. *Measurement theory, with applications to Decision Making, Utility and the Social Sciences*. Addison-Wesley, Boston, 1979.
- K. W. S. Roberts. Interpersonal comparability and social choice theory. *Review of Economic Studies*, 47(2):421–39, 1980.
- J. Rosenhead. *Rational analysis of a problematic world*. J. Wiley, New York, 1989.

- M. Roubens and Ph. Vincke. *Preference Modeling*. Springer Verlag, 1985.
- B. Roy and D. Bouyssou. *Aide Multicritère à la Décision : Méthode et Cas*. Economica, Paris, 1993.
- B. Roy. Classement et choix en présence de points de vue multiples: La méthode ELECTRE. *Revue Française d'Informatique et de Recherche Opérationnelle*, 8:57–75, 1968.
- B. Roy. Problems and methods with multiple objective functions. *Mathematical Programming*, 1(2):239–266, 1971.
- B. Roy. The outranking approach and the foundation of electre methods. *Theory and Decision*, 31:49–73, 1991.
- B. Roy. Decision science or decision-aid science? *European Journal of Operational Research*, 66:184–203, 1993.
- B. Roy. *Multicriteria Methodology for Decision Aiding*. Kluwer Academic, Dordrecht, 1996.
- A.A. Salo and R.P. Hamalainen. Preference ratio in multiattribute evaluation (PRIME) - elicitation and decision procedures under incomplete information. *IEEE Transactions on Systems, Man and Cybernetics: Part A*, 31:533–545, 2001.
- T. Sandholm and C. Boutilier. Preference elicitation in combinatorial auctions. In P. Cramton, Y. Shoham, and R. Steinberg, editors, *Combinatorial Auctions*, pages 233 – 264. MIT Press, Boston, 2006.
- A. Schärli. *Décider sur plusieurs critères, panorama de l'aide à la décision multicritère*. Presses Polytechniques Romandes, Lausanne, 1985.
- D. Shaw, L.A. Franco, and M. Westcombe. Special issue: Problem structuring methods ii. *Journal of the Operational Research Society*, 58:545–700, 2007.
- I. Stamelos and A. Tsoukiàs. Software evaluation problem situations. *European Journal of Operational Research*, 145:273–286, 2003.
- S. Toulmin. *The uses of Arguments*. Cambridge University Press, 1958.
- A. Tsoukiàs and Ph. Vincke. "extended preference structures in mcda". In J. Climaco, editor, *Multicriteria Analysis*, pages 37–50, 1997.
- A. Tsoukiàs, P. Perny, and Ph. Vincke. From concordance/discordance to the modelling of positive and negative reasons in decision aiding. In D. Bouyssou, E. Jacquet-Lagrèze, P. Perny, R. Slowinski, D. Vanderpooten, and Ph. Vincke, editors, *Aiding Decisions with Multiple Criteria: Essays in Honour of Bernard Roy*, pages 147–174. Kluwer Academic, Dordrecht, 2002.

- A. Tsoukiàs. Preference modelling as a reasoning process: a new way to face uncertainty in multiple criteria decision support systems. *European Journal of Operational Research*, 55:309–318, 1991.
- A. Tsoukiàs. On the concept of decision aiding process. *Annals of Operations Research*, 154(1):3–27, 2007.
- A. Tsoukiàs. From decision theory to decision aiding methodology. *European Journal of Operational Research*, 187:138–161, 2008.
- S.W. van den Braak, H. van den Oostendorp, H. Prakken, and G. Vreeswijk. A critical review of argument visualization tools: Do users become better reasoners? In *Proceedings of the 6th International Workshop on Computational Models of Natural Argument (CMNA'06)*, pages 67–75, 2006.
- S.W. van den Braak, G. Vreeswijk, and H. Prakken. "aver"s: An argument visualization tool for representing stories about evidence. In *Proceedings of the 11th International Conference on Artificial Intelligence and Law (ICAAIL'07)*, pages 11–15. ACM Press, 2007.
- B. Verheij. *Rules, Reasons, Arguments : formal studies of argumentation and defeat*. PhD thesis, Maastricht University, Holland, 1996.
- B. Verheij. On the existence and multiplicity of extensions in dialectical argumentation. In S. Benferhat and E. Giunchiglia, editors, *Proceedings of the 9th International Workshop on Non-Monotonic Reasoning (NMR'02)*, pages 416–425, 2002.
- B. Verheij. Dialectical argumentation with argumentation schemes: an approach to legal logic. *Artificial Intelligence and Law*, 11(2-3):167–195, 2003.
- Ph. Vincke. *Multicriteria Decision-Aid*. J. Wiley, New York, 1992.
- Ph. Vincke. Preferences and numbers. In A. Colorni, M. Paruccini, and B. Roy, editors, *A-MCD-A - Aide Multi Critère à la Décision - Multiple Criteria Decision Aiding*, Joint Research Center, pages 343–354. The European Commission, 2001.
- D.N. Walton and T. F. Gordon. Critical questions in computational models of legal argument. In P.E. Dunne and T. Bench-Capon, editors, *International Workshop on Argumentation in Artificial Intelligence and Law*, pages 103–111, 2005.
- D.N. Walton and E.C.W. Krabbe. *Commitment in Dialogue : Basic conceptions of Interpersonal Reasoning*. State University of New York Press, 1995.

- D. Walton and C. Reed. Argumentation schemes and defeasible inferences. In G. Carenini, F. Grasso, and C. Reed, editors, *Workshop on Computational Models of Natural Argument (CMNA'02)*, 2002.
- D.N. Walton. *Logical dialogues games and fallacies*. University press of America, 1984.
- D. Walton. What is reasoning? what is an argument? *Journal of Philosophy*, 87:399–419, 1990.
- D. Walton. *Argumentation schemes for Presumptive Reasoning*. Mahwah, N. J., Erlbaum, 1996.
- D.N. Walton. *Appeal to Expert Opinion*. Penn State Press, 1997.
- D.N. Walton. Justification of argument schemes. *Australasian journal of logic*, 3:1–13, 2005.
- D.N. Walton. Visualization tools, argumentation schemes and expert opinion evidence in law. *Law, Probability and Risk Advance Access*, 6(1,4):1–22, 2007.
- J. Wang. Artificial neural networks versus natural neural networks: a connectionist paradigm for preference assessment. *Decision Support Systems*, 11:415–429, 1994.
- A.N Whitehead and B. Russel. *Principia Mathematica*. Cambridge University Press, Cambridge, 1910.
- R. Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. *Ordered sets*, pages 445–470, 1982.
- M.J. Wooldrdige. *Reasoning about rational agents*. MIT Press, Cambridge, MA, USA, 2000.

Part

Appendix

A Binary relations

In what follows some basic properties about binary relations. Let A be a finite set of elements a, b, \dots , a binary relation S on the set A is a subset of the cartesian product $A \times A$, that is, a set of ordered pairs (a, b) such that a and b belong to A , i.e., $S \subseteq A \times A$. If the ordered pair (a, b) is in S , we write aSb . Otherwise, we write $a\neg Sb$.

A binary relation S on the set A is

$$\left\{ \begin{array}{ll} \text{reflexive} & \text{iff } aSa, \forall a \in A, \\ \text{irreflexive} & \text{iff } a\neg Sa \forall a \in A, \\ \text{symetric} & \text{iff } aSb \Rightarrow bSa, \forall a, b \in A \\ \text{asymetric} & \text{iff } aSb \Rightarrow b\neg Sa, \forall a, b \in A \\ \text{transitive} & \text{iff } aSb, bSc \Rightarrow aSc \forall a, b, c \in A, \\ \text{complete} & \text{iff } aSb \text{ or } bSa, \forall a, b \in A, \end{array} \right.$$

Given a binary relation S on the set A we respectively denote by P and I the asymmetric part and the symmetric parts of S :

$$\left\{ \begin{array}{ll} aPb & \Leftrightarrow aSb \text{ and } b\neg Sa \\ aIb & \Leftrightarrow aSb \text{ and } bSa \end{array} \right.$$

Let S be a binary relation on the set A , $S = \langle P, I \rangle$ is a linear order if:

$$\left\{ \begin{array}{ll} I & = \{(a, a), \forall a \in A\} \\ P & \text{is transitive} \\ P \cup I & \text{is reflexive and complete.} \end{array} \right.$$

Let S be a binary relation on the set A , $S = \langle P, I \rangle$ is a weak order if:

$$\left\{ \begin{array}{ll} I & \text{is transitive} \\ P & \text{is transitive} \\ P \cup I & \text{is reflexive and complete.} \end{array} \right.$$

B Toulmin's scheme with the Carneades Software

The Figure B.1 illustrates the representation of Toulmin's scheme (see Table 2.6) under the Carneades software. Such a software provides tools supporting a variety of argumentation tasks, including: argument mapping and visualization, argument evaluation, applying proof standards and respecting the distribution of the burden of proof, etc.

We can observe in the Figure B.1, that the conclusion "Harry is a British subject" (top of the tree) has the dialectal status *accepted* and the *Scintilla of Evidence* as a proof standard. However, the acceptability of this conclusion is false. In fact, according to Function 2.1, the statement does not satisfy its proof standard because it is not supported by a defensible pro argument. We recall that whether or not an argument is defensible depends on whether or not its premises hold. In this case, the premise "Harry has become an American citizen" does not hold, because it is an exception and its status is accepted. Therefore, the conclusion cannot be acceptable. The following table summarizes information on the acceptability of each statement of the Toulmin's scheme. We supposed that all the statements have the same proof standard (Scintilla of Evidence (SE)) and the same dialectical status "accepted". On the software, it is possible to consider different proof standards for the statements at the same time.

Statement	Premise type	Proof standard	holds	acceptability
Harry is a British subject	–	SE	–	False
Harry was born in Bermuda	Assumption	SE	True	False
A man born in bermuda will...	Ordinary	SE	True	True
Civil code section	assumption	SE	True	False
Harry has become an American citizen	Exception	SE	False	True

Conversely, the acceptability of this premise is true (it is indicated on the right), because it is supported by a defensible pro argument. Thus, it satisfies its proof standard which is also the scintilla of Evidence (see Figure B.2).

Now, if we imagine that there is no argument supporting the premise “Harry has become an American citizen”, and that its status is stated. Thus, the conclusion “Harry is a British subject” become acceptable (see Figure B.3).

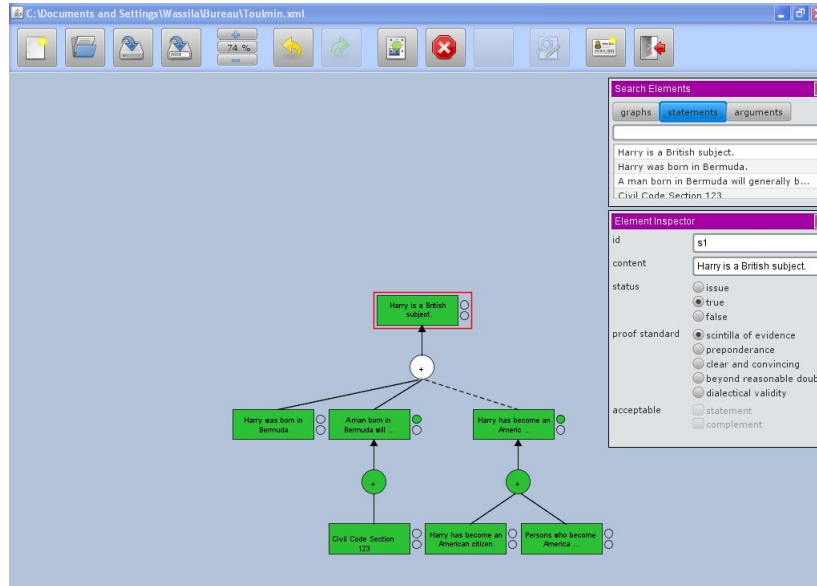


Figure B.1 — Toulmin under Carneades(1)

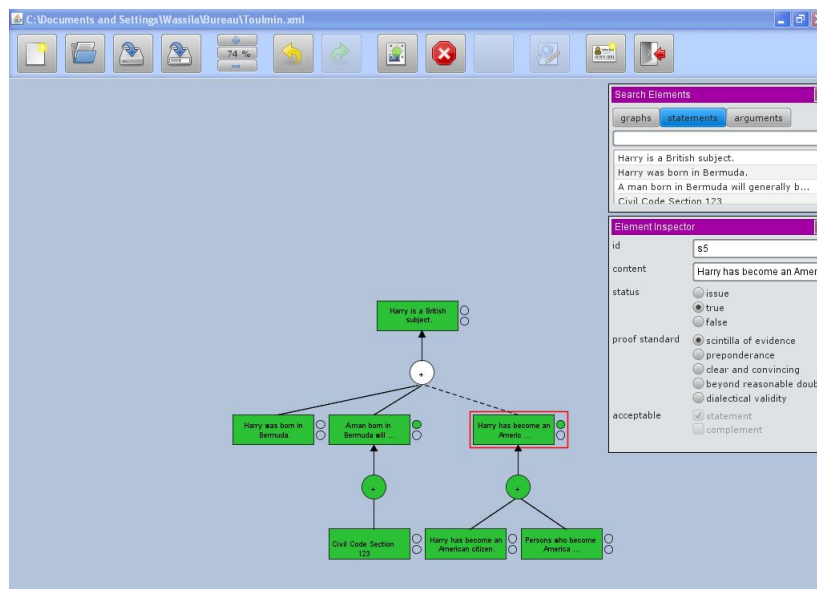


Figure B.2 — Toulmin under Carneades(2)

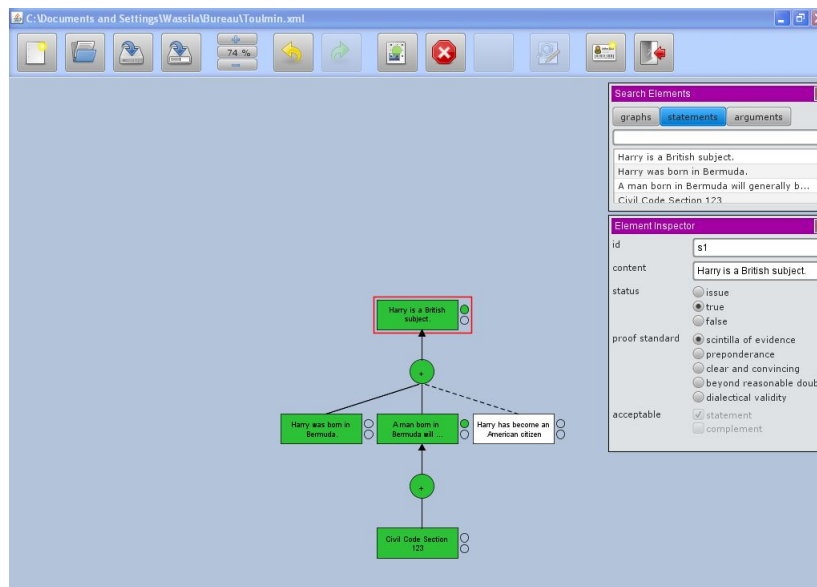


Figure B.3 — Toulmin under Carneades (3)

C --- Argumentative Dialogue

A **persuasion dialogue** is initiated from position of conflict in which an agent believes p and the other believes $\neg p$, and both try to persuade the other to change their mind. The dialogue continues until the dispute is resolved. It involves an attempt by one participant to have another participant endorse some proposition or statement. The statement at issue may concern the beliefs of the participants or proposals for action⁵, and the dialogue may or may not involve conflict between the participants. If the participants are guided only by the force of argument, then whichever participant has the more convincing argument, taking into account the burden of proof, should be able to persuade the other to endorse the statement at issue.

The negotiation dialogue is argumentative in a different kind of way. Negotiation is a form of interest-based bargaining where the goal is to “get the best deal”. It occurs when two or more parties attempt to jointly divide some resource (which may include the participants’ own time or their respective capabilities to act), where the competing claims of the participants potentially cannot all be satisfied simultaneously. Here, co-operation is required by both parties in order to engage in the negotiation dialogue, but, at the same time, each participant is assumed to be seeking to achieve the best possible deal for him or herself.

The inquiry is an essentially cumulative type of dialogue, meaning that retraction of commitment is not generally permitted. The inquiry is a collaborative investigation that seeks to prove something, or alternatively to show that the existing evidence is insufficient to prove it. The inquiry is a hierarchical procedure of reasoning where the premises are supposed to be better known or established than the conclusion to be proved from them. It occurs when two or more participants, each being ignorant of the answer to some question, and each believing the others to be ignorant also, jointly seek to determine the answer. These dialogues do not start from a position of conflict, as no participant has taken a particular position on the question at issue; they are trying to find out some knowledge, and no one need resile from their existing beliefs. Aircraft disaster investigations may be seen as examples of Inquiry dialogues.

An Eristic dialogue is one where the participants vent perceived grievances, as in a quarrel, and the dialogue may act as a substitute for physical fighting.

The information seeking dialogues are similar to inquiries, but differ in their initial conditions. An information seeking dialogue is initiated when there is an asymmetry between the agent in the sense that one is thought by the other to have more information in regard to p . The interview is one type of information-seeking dialogue, for example. Another type is the advice-solicitation dialogue, where the goal of the one party is to seek advice, in order to carry out an action or solve a problem, by consulting another party who is in a special position to offer. It occurs when one party does not know the answer to some question, and believes (perhaps erroneously) that another party does so. The first party seeks to elicit the answer from the second by means of the dialogue. Expert consultation is a common important subtype of this type of dialogue. Note, when the information sought concerns an action or course of action, I term this type of dialogue, a Plan-Seeking dialogue.

A deliberation dialogue represents the process of formalising a plan of action, and thus is also concerned with intentions. The joint aim of deliberation is to reach an agreement on a plan, and the individual aims are to influence this agreement to their benefit. It occurs when two or more parties attempt to agree on an action, or a course of action, in some situation. The action may be performed by one or more the parties in the dialogue or by others not present. Here the participants share a responsibility to decide the action(s) to be undertaken in the circumstances, or, at least, they share a willingness to discuss whether they have such a shared responsibility.

D Minimal explanations w.r.t Generic statements

We discussed in chapter 5 the construction of minimal explanation w.r.t generic statements, for a condorcet winner. In what follows, we provide a detailed description of how the minimal cost explanations are constructed, from a given minimal length coalition. The problem we face is formally described as follows:

- ▷ **Input:** n the number of criteria, m the number of candidates, c a Condorcet winner, and $\{h_0, \dots, h_k\}$ a minimal length coalition ($k \geq n/2$).
- ▷ **Output:** An explanation e such that $e = \operatorname{argmin}_{x \in \mathcal{E}} \operatorname{cost}(x)$, where \mathcal{E} is the set of all explanations (ie, an explanation minimizing the cost)

Let us denote by $\operatorname{rank}_{h_i}(c)$ the rank of a candidate c on a criteria h_i (where 1 is the top rank), and by $X_{h_i}^\uparrow$ (resp. $X_{h_i}^\downarrow$) the set of candidates ranked above (resp. below) c in h_i .

In short the problem amounts, for each criteria, to select whether the statement on h_i should be of the form :

- (i) “ c beats everyone except X ” where $X = X_{h_i}^\uparrow$ (note that “ c beats everyone” is a special case of except statement where $X = \emptyset$), or
- (ii) “ c beats Y ”, where $Y \subseteq X_{h_i}^\downarrow$ (a *subset* (not necessarily strict) of the candidates ranked below c in h_i).

The subtlety with (ii) is that the candidates which need to be exhibited will depend on the ones already “covered” by the other criteria. More precisely, let us denote by *ToCover* a *multiset* containing as many instances of the candidates as required to complete the explanation. Initially, we have $\operatorname{ToCover} = \{x_0, x_0, \dots, x_0, \dots, x_m, x_m, \dots, x_m\}$, where each candidate name is duplicated $n/2 + 1$ times as each other candidate except c must be “covered” $n/2 + 1$ times. This multiset

will be updated during the process when new criteria are selected for the explanation, so in fact we will note $ToCover_{e_t}$ to indicate that this is relative to the currently constructed explanation e_t .

The cost of the respective statements is as follows:

- (i) the cost is $2 + rank_{h_i}(c)$
- (ii) the cost is $2 + |\{x | x \in ToCover_{e_t} \cap X_{h_i}^\perp\}|$

Before going any further we make some initial observations:

1. Not all the criteria can be used for (i), in fact only those criteria where where c is not below $m/2$, otherwise it will always be more costly to use the (i) statement. We denote by $CandCriteria = \{h_i | rank_{h_i}(c) \geq m/2\}$
2. All the criteria where c is ranked can be used for (i) (meaning that they can be removed from $CandCriteria$), but not necessarily all of them if there are more than $n/2$ such criteria.

Principle of the algorithm. In short, for each criterion we must decide whether to use a statement of the form (i) or (ii). As previously noticed this decision depends on the current state of the multiset (hence of the previous choices), as the cost of statement (ii) depends on how many alternatives are actually covered by the statement. Simple heuristics consisting in considering criteria in the order of the (decreasing) rank of c (or of the number of covered alternatives) and in taking the less costly statement available do not always work (although they work very well in practice). Of course this does not rule out the possibility of such a greedy algorithm existing based on a slightly more sophisticated heuristics. In the absence of firm evidence, we give an informed search algorithm (more specifically, A^*) based on a simple admissible heuristics.

The search tree is a binary tree composed of nodes that are candidate criteria ($CandCriteria$), introduced in decreasing order of the ranking of the identified Condorcet winner. A node is expanded with two sons, one where this criteria is used as a (i) statement (“except”), and one where it is used as (a collection of) simple statements (ii). The cost associated to a node is classically decomposed as a cost and heuristic function: $g(x)$ is the current cost of the partial solution (for instance, if we take h_3 where c is third the cost is 5); and $h(x)$ is a heuristics which (under-) evaluates the cost for the remaining alternatives to be covered. The proposed heuristics is to suppose that the elements of $ToCover$ will be optimally covered by the remaining candidate criteria, that is, *disregarding the exact nature of the elements of $ToCover$* , we assume that “except” statements can be used as long as there are enough candidates to cover to make it cheaper than simple statements, but bearing in mind that the coalition is minimal (hence all the criteria must indeed be used). It is clearly an admissible heuristics, as this would minimize the cost involved to cover the candidates required to be covered. Algorithm D.1 gives the details of

how to compute $h(x)$. We assume that at the beginning the criteria are such that $h[0]$ is the one where c is ranked higher, etc.

Algorithme D.1 — Function Heuristics(e_t)

Input : s : the list of ordered current candidate criteria *OrdCandCriteria*, and the number of criteria selected for single statements *Single*

Data : z : the multiset *ToCover*;

```

1  $z \leftarrow |ToCover\_et|$ ;
2  $s \leftarrow |OrdCandCriteria| + Single$ ; //s number of criteria potentially available
  for single statements
3  $i \leftarrow 1$ ;
4 while ( $z - (s - 1) > rank(h[i])$ ) do
5    $s \leftarrow s - 1$ ;
6    $z \leftarrow z - (\min(m - rank(h[i]), z - s))$ ; //how many alternatives are covered at
  best by this criterion
7    $h \leftarrow h + 2 + rank(h[i])$ ; //compute the cost
8    $i \leftarrow i + 1$ ; //move to the next best criteria
9 end
10  $h \leftarrow h + z + 2 * s$ ; //add the cost of the remaining alternatives to be explained
  by single statements
11 return  $h$ 

```

The algorithm then chooses the more promising candidate node to be expanded. When a node is expanded the *ToCover* multiset is updated, and the algorithm computes the sons of this node. If there are none, then the node is a leaf and the final cost for this solution can be computed (by counting the cost induced by the alternatives uncovered in the proposed solution).

Illustrative example. Consider the following rankings, where we have 5 criteria $H = \{h_0, h_1, h_2, h_3, h_4\}$ and 10 alternatives $A = \{a, b, c, d, e, g, h, i, j\}$.

$$\begin{aligned}
 h_0 : & a \succ b \succ i \succ c \succ d \succ e \succ f \succ g \succ h \succ j \\
 h_1 : & c \succ a \succ b \succ d \succ e \succ f \succ g \succ h \succ i \succ j \\
 h_2 : & d \succ e \succ j \succ c \succ a \succ b \succ f \succ g \succ h \succ i \\
 h_3 : & a \succ b \succ d \succ c \succ f \succ g \succ e \succ h \succ i \succ j \\
 h_4 : & e \succ f \succ g \succ h \succ c \succ a \succ b \succ d \succ i \succ j
 \end{aligned}$$

In this example the condorcet winner is c and the minimal coalition to support it is the set H . In what follows we construct the minimal explanation by using the heuristics presented in the Algorithm D.1. The candidate criteria set is $CandCriteria = \{h_0, h_2, h_3, h_4\}$. We note that on the criterion h_1 , the action c is ranked first then it is not necessary to include it to the candidate set, since it will be used in the explanation. We will consider two sets *Exept* the set of criteria that

are selected such that the statement is under the form of (i) and *Single* the set of criteria that are selected such that the statement is under the form of (ii). The idea, now, is to decide for each criterion h_i of the candidate set whether it belongs to *Except* or *Single*. The initial *ToCover* set is as follows:

$$\begin{array}{cccccccccc} a & b & d & e & f & g & h & i & j & \\ a & b & d & e & f & g & h & i & j & \end{array}$$

Therefore, $|ToCover| = 18$. We note that each action must be covered $n/2+1$ times, but as h_1 is necessary in the explanation, then it covers all the actions once. Moreover, for each criterion i we compute $f_i(x) = g_i(x) + h_i(x)$, such that: $g_i(x)$ is the cost of the current partial solution and $h_i(x)$ is the heuristics which evaluates the remaining criteria. It is computed by the algorithm D.1.

The figure D.1 illustrates the search tree after the application of the Algorithm D.1 on the Candidate set. The red lines indicate the nodes that should be taken in the explanation. The details at each node are given in the Table below. At the end, in this example, the criteria h_0, h_2 and h_4 will be used as a (i) statement and h_3 as a (ii) statements. Thus, the explanation will include the following statements: $[h_0 : c \succ * \setminus \{a, b, i\}]$, $[h_1 : c \succ *]$, $[h_2 : c \succ * \setminus \{d, e, j\}]$, $[h_4 : c \succ * \setminus e]$ and $[h_3 : c \succ \{e\}]$.

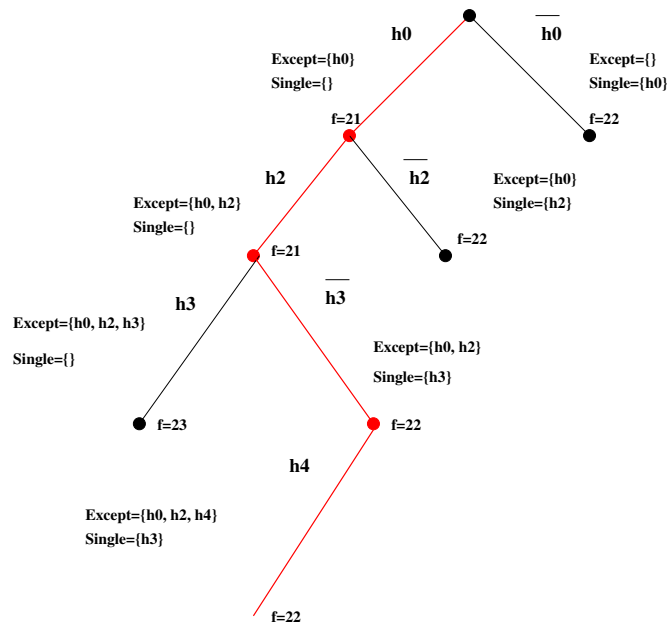


Figure D.1 — The search tree

critierion	Parent node	$ ToCover $	$ToCover$	s	$g_i(x)$	$h_i(x)$	$f_i(x)$
h_0	-	12	$a b d e f g h i j$ $a b d e f g h i j$	$\{h_2, h_3, h_4\}$	6	$6+6+3$	21
$\overline{h_0}$	-	18	$a b d e f g h i j$ $a b d e f g h i j$	$\{h_2, h_3, h_4\} + \{\overline{h_0}\}$	0	$6+6+6+4$	22
h_2	h_0	6	$a b d e f g h i j$ $a b d e f g h i j$	$\{h_3, h_4\}$	12	$6+3$	21
$\overline{h_2}$	h_0	12	$a b d e f g h i j$ $a b d e f g h i j$	$\{h_3, h_4\} + \{\overline{h_2}\}$	6	$6+6+4$	22
h_3	h_2	3	$a b d e f g h i j$ $a b d e f g h i j$	$\{h_4\}$	18	5	23
$\overline{h_3}$	h_2	6	$a b d e f g h i j$ $a b d e f g h i j$	$\{h_4\} + \{\overline{h_3}\}$	12	$6+4$	22
h_4	$\overline{h_3}$	1	$a b d e f g h i j$ $a b d e f g h i j$	$\{\overline{h_3}\}$	19	3	22

E An example of dialogue

We presented in Chapter 6 an example of a dialogue (see below) between the system ArgDec and a user. There, we show how the different elements and concepts presented in this work, can be used in a “real” application. The aim in this annex is to give more details on the the dialogue game by indicating at each turn: the locutions used by the two players (i.e the system and the user), the statements and their dialectical status. Thus, the Table E.2 presents the different locutions and the Table E.1 presents different examples of statements and their corresponding dialectical status. Moreover, the Table E.2 is based on the different notions and rules presented in Section 5.6.

User: [specify its evaluation model: four actions $A = \{a_0, a_1, a_2, a_3\}$, five criteria $H = \{h_0, h_1, h_2, h_3, h_4\}$. The performance table is as follows:]

	h_0	h_1	h_2	h_3	h_4
a_0	7	6	2	3	5
a_1	6	4	8	4	7
a_2	3	2	5	2	3
a_3	7	7	2	0	2

- (1) S: I recommend a_1 as being the best choice
- (2) U: Why is that the case?
- (3) S: because a_1 is globally better than all other alternatives
- (4) U: Why?
- (5) S: a_1 is ranked first on a strict majority of criteria (h_2, h_3, h_4).
- (6) U: mmmm. I think I changed my mind: a_2 should have value 5 on h_3 .
- (7) S: OK. I still recommend a_1 as the best choice.
- (8) U: Why is it still the case?
- (9) S: Because a_1 is still preferred to each other alternatives by a majority of criteria: it is still ranked first on h_2 and h_4 , and is only beaten by a_2 on h_3 . But b beats c on h_0 .
- (10) U: I see, but I would prefer a_0 to a_1
- (11) S: Why?

- (12) U: Because a_0 is better on $\{h_0, h_1\}$
(13) S: Fine. But then why is not d preferred?
(14) U: No. a_3 is too bad on h_3 . This would not be justifiable.
(15) S: OK.

We note that in Turn 5, 9 and 12, for sake of simplicity, we have not detailed in the table E.2 the premises of the arguments. However, we give examples in the Table E.1. We draw the attention of the reader to the fact that the statements representing the premises belong to arguments of different levels as it is shown on the argument graphs presented in Chapter 6.

Turn	Statement	Dialectical status
1	recommended(a_1)	stated
2	recommended(a_1)	questioned
3	recommended(a_1) $Pref_H(a_1, X)$	accepted stated
4	recommended(a_1) $Pref_H(a_1, X)$	rejected questioned
5	recommended(a_1) $Pref_H(a_1, X)$ Majority Defines(Comp($h_2(a_1), h_2(a_0)$), $Pref_{h_2}(a_1, a_0)$)	accepted accepted accepted stated
6	$Pref_{h_3}(a_1, a_2)$ $Pref_H(a_1, a_2)$ $Pref_H(a_1, X)$	rejected rejected rejected
7	recommended(a_1)	stated
8	recommended(a_1)	questioned
9	recommended(a_1)	accepted
10	recommended(a_1) recommended(a_0)	rejected stated
11	$pref_H(a_0, a_1)$	questioned
12	$pref_H(a_0, a_1)$ WEIGHTEDM Defines(Comp($h_0(a_0), h_0(a_1)$), $Pref_{h_0}(a_0, a_1)$)	accepted accepted stated
13	recommended(a_3) recommended(a_0) recommended(a_1)	stated rejected rejected
14	recommended(a_3)	rejected

Table E.1 — Examples of statements and their dialectical status

Turn	Player	Locution
1	S	assert(recommended(a_1))
2	U	challenge(recommended(a_1))
3	S	argue(recommended(a_1), $Pref_H(a_1, X)$, $\Pi(\text{RECOMM-SCHEME})$)
4	U	challenge($Pref_H(a_1, X)$)
5	S	argue($Pref_H(a_1, X)$, premises, $\Pi(\text{UC-TOP, MC-PW-EV, MAJO-SCHEME, UC-PW-EV})$)
6	U	update= retract($h_3(a_2) = 2$) + assert($h_3(a_2) = 5$) pose(CQ N°7, assumption, $Pref_{h_3}(a_1, a_2)$, UC-PW-EV)
7	S	assert(recommended(a_1))
8	U	challenge(recommended(a_1))
9	S	argue(recommended(a_1), premises, $\Pi(\text{UC-GOOD, MC-PW-EV, MAJO-SCHEME, UC-PW-EV})$)
10	U	pose($Pref_H(a_0, a_1)$, exception, recommended(a_1), MC-PW-EV)
11	S	challenge($Pref_H(a_0, a_1)$)
12	U	argue($Pref_H(a_0, a_1)$, premises, $\Pi(\text{MC-PW-EV, WEIGHTEDM-SCHEME, UC-PW-EV})$ + assert(NEvidence(anonymity))
13	S	assert(recommended(a_3))
14	U	pose(CQ N°6, exception, $Pref_{h_3}(a_3Pa_0)$, $\Pi(\text{UC-PW-EV})$)
15	S	accept + assert(recommended(a_0))

Table E.2 — Locutions

Vu:
Le Président
M.

Vu:
Les Suffrageants
M.

Vu et permis d'imprimer:

Le Vice-Président du Conseil Scientifique Chargé de la Recherche de l'Université Paris-Dauphine

Résumé

Nous proposons dans cette thèse d'utiliser certains concepts et outils de la théorie de l'argumentation dans le contexte d'un processus d'aide à la décision, i.e. une interaction entre un expert et un décideur, avec pour principal objectif d'arriver à un consensus. Notre ambition à travers ce travail est double : (i) renforcer les capacités d'aide à la décision d'un expert en lui offrant un moyen formel pour représenter de façon explicite et transparente les raisons pour lesquelles il recommande (ou non) une solution (le cas échéant), et (ii) renforcer les capacités d'un outil (semi) automatique pour lui permettre de gérer le dialogue avec un décideur, dans le but de l'aider à résoudre son problème de décision. Dans la première partie de notre travail, nous proposons de combler l'absence d'explications formelles durant un processus d'évaluation multicritère, en construisant une structure hiérarchique de trois niveaux de schémas d'argument. Ces schémas vont permettre de représenter de façon explicite et claire les hypothèses de chaque étape du processus. Dans la seconde partie, nous souhaitons montrer que l'utilisation des outils de l'argumentation, tel que la fonction d'acceptabilité, les questions critiques, vont permettre de faciliter le processus de révision et/ou de mise à jour durant l'évaluation multicritère. Pour cela, nous proposons de construire un système qui va permettre, entre autres, (i) de présenter une recommandation qui peut être explicitement justifiée, (ii) de réviser n'importe quelle étape du raisonnement durant le processus et d'informer des conséquences des évolutions en utilisant, par exemple, une représentation graphique des arguments échangés.

Mots-clés: Processus d'aide à la décision, Evaluation multi critère, Théorie de l'argumentation.

Abstract

We propose in this thesis to use tools and concepts from argumentation theory in a decision aiding process, i.e. an interaction between, at least, an expert and a decision maker, where the main objective of this process is to reach a consensus between the two participants. Our ambitions through this work is twofold: (i) enhance decision support capabilities of the analyst representing explicitly and accountably the reasons for which he recommend (or not) a solution (if any); and (ii) enhance decision support capabilities of an (semi) automatic device to handle (at least partially) the dialogue with the user. We first propose to specify in an argumentative terms the steps involved in a multiple criteria evaluation process, a phase of the decision aiding process. To do that, we construct a hierarchical structure where we identify three levels of argument schemes that are embedded. The objective is to make explicit assumptions that are hidden in such a process, hence allowing meaningful and natural explanations. Secondly, we propose to show that by relying on notions of argumentation such as acceptability function and critical questions, we facilitate the revision/update occurring during such a process. The kind of system that we sketch here would allow, among others,: (i) to present a recommendation that can be explicitly justified; (ii) to revise any piece of reasoning involved in this process, and inform of the consequences of such modifications by presenting (for instance graphically) the exchanged arguments.

Keywords: Decision aiding process, Multiple criteria evaluation, Argumentation theory.