FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

An argumentation model for deliberation in heterogeneous multiagent systems

Daniel José Marques Nora

U.PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Professor Ana Paula Rocha

Second Supervisor: Professor Rosaldo J. F. Rossetti

July 7, 2015

An argumentation model for deliberation in heterogeneous multiagent systems

Daniel José Marques Nora

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Professor Daniel Castro Silva
External Examiner: Professor Paulo Novais
Supervisor: Professor Ana Paula Rocha

July 7, 2015

Abstract

The perception of the society about the negative impact of transportation on quality of life and on the environment has greatly increased over the last few decades. To be capable of satisfying the transportation requirements of a region, planners must design efficient transit facilities while respecting environmental, social, economic and legal factors. This task is highly complex and error-prone due to the overwhelming number of issues to take into account, pertaining to widely different areas of expertise. Ideally, planners should be assisted by good decision support systems to efficiently discover the best solutions in each situation. This principle applies to many other complex design problems.

The aim of this thesis project is to develop an argumentation model for deliberation in heterogeneous multiagent systems. The model shall be generic enough to be easily adaptable to different application contexts, facilitating the construction of decision support systems based on argumentative agents. Such agents specialize in different areas of knowledge, and collaborate in search of solutions to solve a given problem as effectively as possible, while trying to respect the different quality parameters valued by each other. This approach should be particularly useful when the problems to solve are too complex to employ traditional optimization algorithms. Additionally, users will benefit from argumentation-based justifications for the system's suggestions.

Several key concepts pertaining to decision-making processes and deliberation dialogues are used as reference for the development of the model, which is split into three major components: an abstract argumentation framework for deliberation; a set of criteria for choosing decisions for individuals and groups, based on accepted arguments, priorities between goals and concepts from social choice theory; and a communication protocol enabling agents to interact and exploit the theory in connection with the previous components, allowing integration of innovating techniques, such as simulation, in order to take full advantage of the agents' analytical abilities. The application domain used for demonstrating the model's capabilities is that of urban transportation networks.

Resumo

Nas últimas décadas, a perceção da sociedade sobre os impactos negativos dos transportes na qualidade de vida e no meio ambiente tem vindo a aumentar. Para poder satisfazer as necessidades de transporte de uma região, os planificadores têm de desenhar infraestruturas de transporte eficientes, tendo em consideração fatores ambientais, sociais, económicos e legais. Esta tarefa é altamente complexa e suscetível a erros devido ao grande número de questões a ter em conta, relacionadas com áreas de conhecimento muito distintas. O ideal seria que os planificadores fossem sempre assistidos por bons sistemas de apoio à decisão para que pudessem descobrir eficazmente as melhores soluções em cada situação. Este princípio aplica-se a muitos outros problemas de *design* complexos.

Este projeto de dissertação tem como objetivo desenvolver um modelo de argumentação para deliberação em sistemas multiagente heterogéneos. O modelo deverá ser suficientemente genérico para que possa ser adaptado facilmente a outros domínios de aplicação, facilitando assim a construção de sistemas de apoio à decisão baseados em agentes argumentativos. Tais agentes são especialistas em diferentes áreas de conhecimento, e colaboram na procura de soluções para resolver determinado problema da forma mais eficaz possível, tentando respeitar os diferentes parâmetros de qualidade valorizados por cada um. Esta abordagem deverá ser especialmente útil quando os problemas a resolver forem demasiado complexos para aplicar algoritmos de otimização tradicionais. Mais ainda, os utilizadores beneficiarão da argumentação como mecanismo de justificação para as decisões propostas pelo sistema.

Diversos conceitos-chave relativos aos processos de tomada de decisão e diálogos de deliberação são utilizados como referência para o desenvolvimento do modelo, que está divido em três componentes principais: uma framework de argumentação abstrata para deliberação; um conjunto de critérios de escolha de decisões para indivíduos e grupos, baseados em argumentos aceites, prioridades entre objetivos e conceitos da teoria da escolha social; e um protocolo de comunicação que permite aos agentes interagir e tirar partido da teoria relativa às componentes anteriores, possibilitando a integração de técnicas inovadoras, tais como simulação, de modo a aproveitar em completo as capacidades analíticas dos agentes. O domínio de aplicação utilizado para demonstrar as aptidões do modelo será o das redes de transporte urbanas.

Acknowledgements

First and foremost, I would like to express my feelings of great gratitude and debt towards my parents, José Nora and Luciana Marques, whose love gives meaning to all things I have done in life, and plenty of motivation for all things I will continue to do. All those moments when I felt discouraged or frustrated become insignificant before the sympathy and support they always provided me with. There are no two people for whom I could be more grateful in my whole life.

Secondly, I would like to thank all other members of my family, grandparents, uncles, aunts and cousins, who were sympathetic during my episodes of absence, both physical and psychological, during those times of greatest hardship. To be in your company cures all.

Thirdly, I would like to stress out the invaluable support that I received from my supervisors, Professor Ana Paula Rocha and Professor Rosaldo Rossetti, towards whom I feel great friendship, respect and admiration. I thank them for always guiding me in the right path, for wisely helping to manage my expectations, and for all the enjoyable discussions we shared together. I wish them all the personal and professional realizations they rightfully deserve. I would also like to extend these wishes to Professor João Pascoal Faria, also from Faculdade de Engenharia da Universidade do Porto, towards whom I feel special appreciation for his sincere commitment and excellent attitude towards his colleagues and students. Furthermore, I acknowledge the precious help that I received in the form of a research grant, awarded by the FCT (Fundação para a Ciência e a Tecnologia).

Lastly, I would like to acknowledge the pertinent remarks by Professor Paulo Novais from Departamento de Informática da Universidade do Minho, by my friends and course colleagues and by all other people who in some manner contributed to my work. I would also like to thank all authors referenced throughout this thesis, which would not have been possible without studying their precious articles. Finally, I humbly thank all those who, in the great and arduous cause of the progress of Science, dedicate themselves to reading my small contribution.

Agradecimentos

Em primeiro lugar, gostaria de expressar o meu enorme sentimento de gratidão e dívida para com os meus pais, José Nora e Luciana Marques, cujo amor dá significado a todas as coisas que fiz na vida, e motivação bastante para tudo o que continuarei a fazer. Todos os momentos em que senti desânimo ou frustração se tornam insignificantes perante a compreensão e apoio que sempre me ofereceram. Não há duas pessoas pelas quais possa ser mais grato em toda a minha vida.

Em segundo lugar, gostaria de agradecer a todos os outros membros da minha família, avós, tios e primos, que foram compreensivos perante os meus episódios de ausência, quer física quer psicológica, durante os períodos de maior esforço. Nada que a vossa companhia não cure.

Em terceiro lugar, gostaria de salientar o apoio valioso que recebi dos meus supervisores, a Professora Ana Paula Rocha e o Professor Rosaldo Rossetti, para com os quais sinto a maior amizade, respeito e admiração. Agradeço por me terem sempre guiado no caminho certo, por ajudarem sabiamente a gerir as minhas expectativas, e por todas as discussões agradáveis que tivemos. Desejo-lhes toda a realização pessoal e profissional que merecem. Gostaria também de estender estes votos ao Professor João Pascoal Faria, também da Faculdade de Engenharia da Universidade do Porto, por quem sinto especial apreço pelo seu empenho sincero e excelente tratamento de todos os colegas de trabalho e alunos. Reconheço ainda a preciosa ajuda que me foi concedida na forma de uma bolsa de investigação da FCT (Fundação para a Ciência e a Tecnologia).

Por último, gostaria de reconhecer as observações pertinentes do Professor Paulo Novais do Departamento de Informática da Universidade do Minho, dos meus amigos e colegas de curso e de todas as outras pessoas que de alguma forma contribuíram para o meu trabalho. Queria ainda deixar o meu agradecimento a todos os autores referenciados ao longo desta tese, a qual não teria sido possível sem a leitura dos seus preciosos artigos. Para terminar, agradeço com humildade a todos aqueles que, pela grande e árdua causa do avanço da Ciência, se dedicarem à leitura do meu pequeno contributo.

Contents

1 Introduction **1**
1.1 Motivation 1
1.2 Aim of the thesis, goals and applicability 2
1.3 Thesis structure 3

2 Literature review **5**
2.1 Multiagent systems 5
2.1.1 Agent Oriented Software Engineering 6
2.1.2 JADE - Java Agent Development Framework 7
2.2 Argumentation-based negotiation 7
2.2.1 Argumentation theory 8
2.2.2 Dialogues of deliberation for collaborative decision-making 10
2.2.3 Abstract argumentation frameworks 12
2.2.4 Argumentation and Decision Support Systems: motivation and applications 15
2.3 Transportation 18
2.3.1 Technology in transportation planning 18
2.3.2 Simulation tools for transportation 19
2.3.3 Assessing network performance 20
2.4 Summary 22

3 Preliminaries **25**
3.1 Overview of the proposed solution 25
3.2 Key concepts 27
3.3 Summary 35

4 Arguments: structure, attacks and acceptability **37**
4.1 Arguments about beliefs and arguments about decisions 37
4.1.1 Definition of epistemic arguments 37
4.1.2 Definition of practical arguments 40
4.2 The abstract argumentation framework 43
4.3 Summary 48

5 Criteria for choosing decisions **49**
5.1 Establishing individual preference order 49
5.2 Computing goal satisfaction 51
5.3 Establishing joint preference order 53
5.4 Analysis of joint preference computation 56
5.5 Summary 58

6 Communication protocol **63**
6.1 Outline of the architecture 63
6.2 Stages 64
6.3 Locutions 69
6.4 Analysis of the communication protocol 70
6.5 Summary 76

7 An example dialogue **77**
7.1 The scenario 77
7.2 Requesting help and conveying the situation 79
7.3 Defining goals and actions 82
7.4 Debating potential decisions 84
7.5 Gathering suggestions 91
7.6 Summary 94

8 Conclusions **95**
8.1 Contributions 95
8.2 Possibilities for future work 97

References **99**

A Locution rules **107**
A.1 Auxiliary locutions 107
A.2 Primary locutions 113

List of Figures

3.1 A simplified diagram of the deliberation process 26

7.1 An illustration of the scenario 78
7.2 An illustration of the relations between arguments 92

LIST OF FIGURES

List of Tables

2.1 LOS for urban facilities and road segments 21
2.2 LOS for intersections with and without traffic lights and roundabouts 22

6.1 Quick reference list of auxiliary locutions 71
6.2 Quick reference list of primary locutions 72

Abbreviations

AF	Argumentation Framework
AFD	Argumentation Framework for Deliberation
AI	Artificial Intelligence
AOSE	Agent Oriented Software Engineering
API	Application Programming Interface
CBR	Case-Based Reasoning
CL	Conversation Log
DDF	Deliberation Dialogue Framework
DN	Do Nothing
DSS	Decision Support System(s)
EA	Epistemic Argument
EAFD	Expert-specific Argumentation Framework for Deliberation
GUI	Graphical User Interface
HBEFA	HandBook Emission FActors for road transport
LOS	Level Of Service
MAS	MultiAgent System(s)
PA	Practical Argument

Chapter 1

Introduction

In this chapter I begin by explaining the general purpose of this thesis, including its motivations and the goals that I seek to address, followed by an outline of the document's structure.

1.1 Motivation

Decision-making is everywhere. Sometimes it is informal, sometimes it's a well-structured process. Sometimes it's individual, sometimes it's in groups. Sometimes it's grounded on reasonable expectations, sometimes it has to deal with much uncertainty. And while it's often about simple things, sometimes it's so complex that it takes vast amounts of time, money and other resources.

We make decisions every day, often without being conscious of doing so: what to do in our free time, what to buy at the market, what to wear, and many other little things. But decision-making is also used for important things: we may not always be aware of it, but it affects our lives. From corporate meetings to political debates, big decisions have an impact on all of us.

Treating complex problems is often done in groups. Such groups discuss potential alternative decisions. Such discussions are usually called deliberations, or deliberation dialogues. In a deliberation, each member of the group presents reasons in favor and against different alternatives. Such reasons are grounded on arguments. Those arguments need to be correct, otherwise the reasons they support can't be considered valid. When in disagreement, people use argumentation to defend their points of view. Therefore, arguments are an intrinsic part of decision-making in groups.

Complex problems often require dealing with vast amounts of information, something that computers can greatly help with. Using decision support systems (DSS), decision makers can more efficiently discover the best alternatives to put into practice in each situation. Therefore, it is of general interest that planners are increasingly more assisted by good DSS. Most DSS, however, are merely aimed at tackling complex numerical problems like projecting profits, impacts or damages.

Couldn't DSS combine such features with argumentation-based justifications in favor or against different decisions, similar to how human beings reason?

1.2 Aim of the thesis, goals and applicability

The aim of this master's thesis project is to develop an argumentation model for deliberation in heterogeneous multiagent systems (MAS). Heterogeneity means that each agent specializes in a different area of knowledge. Agents argue with each other, debating alternative decisions. Their ultimate purpose is to collaborate in search of solutions to solve a given problem as effectively as possible, while trying to respect the different quality parameters valued by each other. The model shall be generic enough to be easily adaptable to different application contexts, facilitating the construction of DSS based on computational agents with argumentative capabilities. This MAS approach should be particularly appropriate to solve complex problems, composed of several sub-problems associated to different domains of knowledge, for which it is difficult to implement regular optimization algorithms. Additionally, users will benefit from argumentation-based justifications for proposed solutions. After all, arguments are the most natural way to elicit suggestions, making those solutions clearer to understand, analyze and discuss by different stakeholders.

To achieve my aim, I propose to address five goals, each of which is tackled on a separate chapter, starting at chapter 3. Those goals are:

1. Determine the high-level key concepts that are common to any deliberation dialogue, independently of the context on which it might take place;

2. Establish an abstract argumentation framework that, based on the previously defined key concepts, allows deliberating agents to represent interactions between arguments and compute which ones are acceptable;

3. Establish a set of criteria that allows agents to determine a preference order for decisions, based on personal values and acceptable arguments in favor and against such decisions;

4. Build a communication protocol that allows agents to conduct deliberation dialogues in accordance with my argumentation model;

5. Demonstrate the capabilities and potential of my theory with an example inspired by a complex problem from the real world.

Each chapter will present a variety of concepts that are novel in literature, building on top of well-established theory and integrating knowledge and techniques from different subjects like decision-making, argumentation theory, social choice theory and simulation, among others. Furthermore, the unified nature of this thesis is by itself an innovating aspect: few approaches consider the issue from so many angles, all the while maintaining consistency. Indeed, most other documented efforts attempt to address one, at most two of these goals.

The application domain that will be used for demonstrating the model's capabilities and potential is that of urban transportation networks, in accordance with the ambitions of a larger project in which this work is integrated, the MAS-Ter Lab platform. This platform, currently in development at the LIACC laboratory at Faculdade de Engenharia da Universidade do Porto, consists of an integrated MAS for cooperative design, visualization and engineering of intelligent transportation solutions by computational agents specializing in different subjects related to transportation planning [ROB07, FERO08]. Although, as far as I was able to determine, the argumentation-based deliberation approach has never been used to tackle the problem of designing urban transportation networks, this paradigm has proven to be particularly useful for solving problems of similar complexity in other areas [BBG12].

Nowadays, the role of urban and transportation planners is increasingly more complex and multidisciplinary. Historically, as the economy of a region grows, so does the demand in terms of passenger and freight transportation [BB01]. The perception of the consequences of this phenomenon has greatly increased over the last few decades, as both the scientific community and the society in general realize the negative impact of unsustainable transportation on the environment and quality of life [SG05], with direct consequences for the economy. Therefore, to be capable of satisfying the transportation requirements of a region, planners must provide efficient transit mechanisms while respecting environmental, social, economic and legal factors. This is a highly complex problem due to the diversity of quality measures to take into account, such as traffic flow, energetic costs, air pollution levels and the existence of green spaces. The search for appropriate measures is an expensive process in terms of time and resources. In the past, this process has often produced rather inappropriate solutions because planners sometimes base their decisions on inadequate models or, when faced with a problem of seemingly intractable complexity, adopt straightforward measures, like expanding highways to accommodate growth in traffic, without taking sustainability factors into account [RCS06].

If transportation planning impacts the general population, it is of general interest that planners are assisted by good software to help them make good decisions. Using a system that presents suggestions substantiated by arguments seems like a good way to achieve this aim, especially if such arguments are previously subject to scrutiny by different computational agents, experienced in different topics. Obviously, not only transportation planning could benefit from this approach: many other problems of similar complexity already rely heavily on DSS, such as engineering design processes [JG09], agriculture [PS04] and environmental impact assessment [CSMC+00]. This is why a generic argumentation model is of particular usefulness.

1.3 Thesis structure

To sum up this chapter, this thesis project concerns the area of argumentation-based deliberation in MAS, in particular, it is sought to determine how an argumentation model could be used as basis for implementing multiagent deliberations, with the ultimate purpose of assisting decision-makers.

The complex problem of designing urban transportation networks is explored as a potential application of the argumentation model. This thesis is structured as follows:

- Chapter 2 contains a literature review on the topics of MAS, argumentation-based negotiation and transportation planning;
- Chapter 3 outlines the general aspects of the proposed solution and the key concepts underlying any deliberation dialogue, which are necessary to the development of subsequent chapters;
- Chapter 4 defines an abstract argumentation framework, which deals with the issues of representing conflicts between opposing arguments, deciding which ones are acceptable and enabling the comparison of decisions grounded on accepted arguments;
- Chapter 5 establishes some criteria for rank-ordering decisions from the perspective of individual agents and joint decision-making, taking into account priorities between goals, accepted arguments and other factors;
- Chapter 6 defines the rules and admissible locutions of a communication protocol that allows agents to engage in deliberation dialogues, exploiting the theory designed over the previous chapters;
- Chapter 7 shows an example dialogue conducted in accordance with my theory, in the context of planning improvements for an urban transportation network;
- Chapter 8 contains a critical review of the theory and suggestions for future work.

Additionally, appendix A provides the detailed rules of each the communication protocol's locutions introduced in chapter 6.

Chapter 2

Literature review

The objective of this chapter is to present an overview of the literature that is relevant to the understanding of the solution proposed in this thesis, focusing on the topics of MAS, argumentation-based negotiation and transportation planning.

2.1 Multiagent systems

MAS have been studied since the early 1980s. Over the years, they have gained massive popularity thanks to the growing perception of their appropriateness in dealing with many different kinds of problems. Their background and some early uses are well explained by Jennings [JSW98].

The applications of MAS are greatly varied, with many examples in industry, commerce, entertainment and other domains. They are employed in contexts as different as controlling manufacturing processes [SWH06], managing air traffic [LL92], bidding automatically on electronic auctions [PCS10] and many others.

The kind of problems typically addressed by MAS present a combination of the following characteristics [BBG12, DPH07]:

- large complexity, with many variables and constraints to be considered;
- heterogeneity, in that they are composed of several sub-problems from different domains of knowledge;
- decentralization, because the information and resources required to solve the problem are dispersed;
- time-constrained, because a solution has to be reached in a short amount of time.

MAS are particularly useful in those situations because they are generally more modular, efficient and adaptable than typical software approaches like optimization algorithms. As a consequence, they can be applied in contexts where it's more important to produce good solutions in a short amount of time than the best possible solutions with high temporal complexity.

2.1.1 Agent Oriented Software Engineering

With the growing popularity of MAS, researchers began to realize the necessity of identifying clear and systematic methodologies to analyze, design and implement agent-based systems [Jen00]. These are generically called Agent Oriented Software Engineering (AOSE) methodologies. A qualitative comparison of different AOSE methodologies can be found in [Dam03]. They differ in aspects like application scope and in the coverage of different software development phases. Some mature, general-purpose methodologies are:

- **GAIA**: one of the first AOSE methodologies [WJK00]. It is composed of two phases: analysis and design. During analysis, developers define the key roles (composed of responsibilities, permissions, activities, and protocols) that exist in the system and the relationships between those roles. During the design phase, each role is assigned to one or more agents, the services provided by each role (in terms of inputs and outputs, pre- and post-conditions) are clearly identified and all communication links that may exist between the agents are defined. GAIA requires a good previous understanding of the system requirements (nothing is said about the elicitation of requirements). It does not address the implementation phase.

- **Tropos**: a methodology that encompasses all phases of the development process, from early requirements to implementation [BPG+04]. The concepts of agent, goal, task and social dependency are used throughout the entire process. In the early requirements phase, developers discover the intentions of stakeholders by understanding their goals and the relationships maintained with each other in the real world. A distinction is made between hardgoals and softgoals (goals whose satisfaction can't be precisely verified, such as the user-friendliness of a computer program). The late requirements phase focuses on the system to be implemented and its interactions within the environment. The system is modeled as an actor with dependencies on other actors. In the architectural design phase, the system is decomposed into several interconnected sub-actors, each one responsible for fulfilling a portion of the system's functions. Afterwards, the capabilities that each actor needs to possess in order to fulfill their goals are clearly identified. Those capabilities are assigned to agents in the system to be implemented. The design phase follows, involving the specification of diagrams for capabilities, plans (which are the building blocks of capabilities) and agent interactions using UML activity and sequence diagrams. Finally, there are a variety of tools to assist developers in designing a system using the Tropos methodology. Some are capable of automatically generating code based on the design, therefore covering part of the implementation phase.

- **Others**: there are other general-purpose AOSE methodologies with lesser degrees of adoption, such as O-MaSE, Prometheus and MESSAGE.

Given the dynamism and complexity of the interactions in urban transportation systems, none of the above-stated general-purpose AOSE methodologies is capable of adequately modeling this scenario and its requirements without significant adjustments. Acknowledging this difficulty, Passos, Rossetti and Gabriel [PRG11] have developed a new AOSE methodology that expands upon Gaia, Tropos and other approaches. The aim of their methodology is to better represent the dynamism of complex systems by integrating concepts from business process modeling and service oriented architectures (SOA) in MAS. The notion of services becomes one of the core concepts of the system model, with the advantage of allowing implementers to exploit the benefits of modularity, flexibility and user-centricity from SOA. They demonstrate a practical example of their methodology to design a MAS that helps people plan efficient trips in urban scenarios, using real-time information about different transportation offers.

Several software tools and libraries have also been developed with the purpose of facilitating the implementation of MAS, based on the perception of common needs across different scenarios.

2.1.2 JADE - Java Agent Development Framework

JADE [Tel] is a middleware framework written in Java whose main purpose is to ease the implementation of distributed MAS in compliance with FIPA (Foundation for Intelligent Physical Agents) specifications. It provides a software library to build and execute agents in a platform-independent run-time environment. JADE's Application Programming Interface (API) allows efficient asynchronous passing of ACL (Agent Communication Language) compliant messages [FIP02] between agents by abstracting over lower level communication issues and automatically selecting the best available communication protocol in each situation. It also incorporates many of the interaction protocols devised by FIPA, such as Contract Net. A set of GUI tools is provided to facilitate the configuration and monitoring of a distributed agent system at run-time.

According to a survey in 2013 [MF14] regarding the impact of MAS technologies, JADE (and its extensions, such as WADE) is the most widely used agent development framework in mature applications in terms of absolute numbers when compared to other frameworks, such as JACK and KOWLAN.

2.2 Argumentation-based negotiation

One of the branches of MAS is that where agents have the ability to communicate with each other in order to agree on a solution that is acceptable to all parties. Some of those communication mechanisms (protocols) are greatly simplified (when compared to regular communication between humans) in order to streamline the decision process and minimize the implementation effort [Woo02]. An example could be an auction platform where the only possible communicative acts are the opening and closing of the auction (by the auctioneer agent) and making bids (by the bidder agents).

To communicate with each other and negotiate solutions, agents can also be implemented with the ability to argue. Although difficult to implement, argumentation-based negotiation has the advantage of being applicable to general, complicated settings, allowing agents to justify their positions and influence each other's thoughts.

Argumentation is everywhere. It is used in conversations in everyday life, courts of law, politics, scientific research and many other contexts. According to Frans Van Eemeren and Rob Grootendorst, argumentation is "a verbal, social, and rational activity aimed at convincing a reasonable critic of the acceptability of a standpoint by putting forward a constellation of propositions justifying or refuting the proposition expressed in the standpoint" [VEG04].

2.2.1 Argumentation theory

The necessity to understand the requirements that make arguments good or bad is not new. Original work in this area dates back to ancient Greece, with Aristotle being one of the most influential authors. He writes about the art of persuasion, taking into account the factors of credibility (*Ethos*), emotion (*Pathos*) and logic (*Logos*) as the major parts of rhetoric speech. His works about logic are particularly famous, setting up the foundations to the modern discipline of formal logic.

Formal logic, to put it simply, is the study of how abstract inference rules can be combined together to make valid reasoning, so that one can determine the validity of a conclusion drawn from a set of premises. The rules are said to be abstract because their application does not concern any particular type of object: what is important is the form of the argument (how the premises are put together), not the content (what is being talked about). Over the last century, developments in formal logic were heavily influenced by mathematics, which was perceived by many as a step away from its original purpose of the study of argumentation as a human practice [Joh00]. Students, teachers of critical thinking and philosophers alike began to see formal logic as too detached from real-life argumentation to be of much use in practical contexts [Joh00, Wal09]. Thus informal logic and argumentation theory were born, taking knowledge from fields like philosophy, law, and, more recently, computer science.

Walton [Wal09] provides an accessible introduction to the various concepts of argumentation theory, including important notions like the types of tasks undertaken by argumentation, attacks and refutations, argumentation schemes, implicit premises, types of dialogue and others. He explains that there are four major tasks undertaken by argumentation.

- **Identification**: extract the premises and conclusion of an argument in textual form;
- **Analysis**: discover implicit premises or conclusions in the argument;
- **Evaluation**: determine whether the argument is weak or strong and why;
- **Invention**: generate new arguments for supporting or refuting some conclusion.

Walton also proposes that one of the tasks of the identification phase is to determine if the argument under analysis fits a well-known, standard pattern or form of argumentation. In [Wal13], Walton provides an analysis of several such patterns of argumentation that typically arise in everyday conversation, legal argumentation and other contexts, called argumentation schemes. Each scheme is associated to a set of critical questions representing standard ways to find flaws in the argument. Many different schemes exist, such as *argument from expert opinion*, *argument from negative consequences* and *practical inference*. The latter is particularly useful in the context of decision-making because it can be employed to justify the necessity to undertake some candidate action. Walton's scheme of practical inference can be defined as follows:

- **Major Premise**: G is a goal for agent X.
- **Minor Premise**: Carrying out action A is necessary/sufficient[1] to realize G.
- **Conclusion**: Therefore, agent X ought to carry out action A.

And the corresponding critical questions are:

1. Does agent X have any other goals that might conflict with goal G?
2. Are there any alternative actions which could realize goal G?
3. Between action A and the alternatives, which are presumably more efficient?
4. Is it possible for agent X to carry out action A?
5. Are there any (negative) consequences of agent X carrying out action A that should be taken into account?

Long before artificial intelligence (AI) researchers realized the potential benefits of using argumentation in MAS, argumentation theory was already a fertile research area. One of the most influential authors from this area on AI is Toulmin [Tou58]. Although the field of argumentation theory has not yet produced a single unifying, widely accepted theory, Toulmin's argument model is one of the major sources of inspiration both for argumentation theorists and AI researchers. An analysis of its impact on AI can be found in [Ver09]. Toulmin suggested a simple model for practical argumentation by identifying six major components for structuring arguments (of which the first three are most commonly employed in AI).

- **Claim**: the conclusion that one seeks to convince others of (e.g. "All birds can fly");
- **Ground**: the fact that one uses as grounds for making the claim (e.g. "All birds have wings");
- **Warrant**: the statement one uses to justify why the ground proves the claim (e.g. "Any animal with wings is able to fly");
- **Backing**: a statement that attempts to strengthen the perceived validity of the warrant (e.g. "Wings are parts of the body responsible for producing aerodynamic forces that are essential to initiate and maintain flight");

[1] Walton [Wal13] distinguishes between two forms of practical inference schemes: the *necessary condition* scheme and the *sufficient condition* scheme

- **Rebuttal**: a statement that seeks to reduce the validity of the claim (e.g. "Chickens are birds but they cannot fly");
- **Qualifier**: an expression that conveys one's certainty concerning a claim (e.g. "All birds can probably fly").

Gottsegen [Got98] provides an interesting analysis of a land use debate in Santa Monica based on Stephen Toulmin's argumentation model. He shows, through the use of diagrams, how the structural components of the arguments put forward by each stakeholder interact with the components of other arguments. This exposes the ways in which the views of the different stakeholders relate to each other in terms of support and rebuttal.

2.2.2 Dialogues of deliberation for collaborative decision-making

In [Wal95], Walton introduces the concept of types of dialogue in argumentation. Each type of dialogue has different purposes, i.e. is suitable in different conversation settings, and the standards for properly conducting argumentation are different as well depending on the type. Types of dialogue can be differentiated by the initial situation of the participants, the goals of each participant and the overall purpose of the dialogue. Walton identifies six fundamental types of dialogue: **persuasion** (prove the validity of one's thesis against an opposing interlocutor), **inquiry** (obtain evidence to verify an hypothesis), **negotiation** (achieve a reasonable deal with another party when there is a conflict of interests), **information-seeking** (exchange information about some topic), **deliberation** (coordinate goals and actions in a decision group with slightly different views) and **eristic** (arguing solely for the sake of winning a conflict, instead of finding the truth). For example, Walton defines a deliberation dialogue as having the initial situation of a "dilemma or practical choice", the goals of the participants as "coordinating goals and actions" and the purpose of the dialogue as "deciding the best available course of action".

The concept of dialogues of deliberation is particularly useful in the context of this work, because such are the dialogues typically conducted by humans in joint decision-making processes. McBurney, Hitchcock and Parsons [MHP07] presented a very abstract protocol for deliberation dialogues between computational entities called Deliberation Dialogue Framework (DDF). As they explain, the need for deliberation dialogues arises when there is a perception of some problem, in the form of a question, that needs to be resolved by taking an action. It can start from a question like *"How should we respond to the prospect of global warming?"*, or, in the context of this work, *"What could be done to improve the performance of this traffic network?"*. Different actions can be evaluated on a large number of attributes like potential consequences (positive and negative), economic costs, practical feasibility, likelihood of success and so forth. They go on to propose a system with different types of sentences (about *actions, goals, constraints, perspectives, facts* or *evaluations*), stages of dialogue (*open, inform, propose, consider, revise, recommend, confirm, close*) and axiomatic semantics for specific locutions (*open_dialogue, enter_dialogue, propose, assert, prefer, ask_justify, move, reject, retract* and *withdraw_dialogu*e).

As the authors recognize, however, no practical assessment is provided about the quality of the outcomes achieved in dialogues using the DDF protocol, nor is the protocol sufficiently complex to cover all relevant interactions of every potential application context (which would be extremely difficult). They do, however, perform an assessment of the framework in light of (proposed) normative principles for conducting deliberations, i.e. sets of rules for rational, productive conversation between humans. They discuss and measure to what extent the DDF adheres (or rather, allows implementers to adhere) to the various principles suggested by Alexy's rules for discourse ethics [Ale90] and Hitchcock's principles for rational mutual inquiry [Hit91]. The authors also make a brief mention to the work of Webler et al. [WTK01], which concerns a set of normative principles identified by participants in public debates about environmental matters like land use. This study shows that people have different perceptions about what constitutes a good deliberation process and that five major ideologies can be identified, contrasting in areas like the necessity for leadership, democratic equality among participants, discussion of fundamental values, etc. McBurney et al. rightfully claim that these principles are too abstract to assess the DDF protocol. Nevertheless, it might have been interesting if the authors made some considerations about how implementers could adapt the protocol in order to account for those different ideologies, especially since they are based on data collected from people involved in real deliberations.

McBurney et al. also make a clear distinction of the concepts of deliberation and negotiation, which may help understand my argument in favor of deliberative argumentation in DSS in subsequent sections:

> "A (...) characteristic of deliberations relates to their mutual focus. Although the participants may evaluate proposed courses of actions according to different standards or criteria, these differences are not with respect to personal interests that they seek to accommodate in the resulting decision. In this respect, a deliberation dialogue differs from a negotiation dialogue, which concerns the division of some scarce resource between competing allocations and so must deal with reconciling potentially competing interests. In a negotiation, for example, it may be deleterious for a participant to share its information and preferences with others. But a sharing strategy should behoove participants in a deliberation; to the extent that agents are unwilling to share information or preferences, we would define their discussion to be a negotiation and not a deliberation." [MHP07].

For those wishing to better understand the specific dynamics of negotiation, Raiffa [Rai02] provides an excellent analysis of this area, including two-party win-lose/win-win and multiple party interactions. He works from the perspective of providing guidelines to help humans become better negotiators, drawing from fields like economics (e.g. game theory) and psychology (e.g. behavioral analysis).

Walton's concept of dialogue shift is also accounted for in the work of McBurney and his colleagues. Participants in a deliberation dialogue may, at any point of the process, disagree about some fundamental concept which needs to be clarified. In such cases, participants may engage in an embedded dialogue for a few moments until the issue is resolved. This new sub-dialogue could be one of deliberation, or it could represent a temporary shift into a persuasion or negotiation dialogue. While the concept of dialogue types and shifts is not explicitly mentioned by Jin [JG09] (see section 2.2.4 for an analysis of his work), his example conversations in the context of collaborative vehicle design provide a good view on how a dialogue of deliberation may embed both negotiation (to settle minor conflicts) and persuasion (to justify or reformulate major design decisions).

2.2.3 Abstract argumentation frameworks

The purpose of an abstract argumentation framework is to determine if some claim is true by analyzing the attack relations between the arguments put forward at a given point in an argumentation process. Such frameworks do not concern the invention of arguments or their internal structure, in fact, for them the content of the argument per se is not relevant. The only information that is necessary for building an argumentation framework is a set of arguments and a binary relation of attacks between those arguments. Therefore, nothing is assumed about the form of the argument, other than that it may attack or be attacked by other arguments. Such a framework can be represented by a directed graph where the nodes are arguments and each edge is an attack between a pair of arguments (the direction of the attack being coincident with the direction of the edge).

Dung [Dun95] proposed the first abstract argumentation framework in a seminal paper in 1995, which has become one of the most important foundations for many developments in argumentation. He starts by defining the concept of argumentation framework in a manner similar to the previous paragraph, then defines different semantics for determining the acceptability of arguments, concluding with a demonstration of the correctness of his theory in light of other approaches to that date. What follows is a selection of some of the most important definitions in Dung's work, which are essential to its comprehension.

Argumentation framework "An *argumentation framework* is a pair $AF = \langle AR, attacks \rangle$, where AR is a set of arguments, and *attacks* is a binary relation on AR, i.e. $attacks \subseteq AR \times AR$." $attack(A, B)$ represents an attack of argument A on argument B.

Conflict-free set "A set S of arguments is said to be *conflict-free* if there are no arguments A and B in S such that A attacks B", i.e. if S has no internal attacks.

Acceptable argument "An argument $A \in AR$ is *acceptable* w.r.t. a set S of arguments iff for each argument $B \in AR$: if B attacks A then B is attacked by [an argument in] S."

Admissible extension "A conflict-free set of arguments S is an *admissible extension* iff each argument in S is acceptable w.r.t. S", i.e. iff S defends all of its elements (by attacking all attackers).

Complete extension "A conflict-free set of arguments S is a *complete extension* iff it is admissible and contains all the arguments it defends".

Grounded extension "A conflict-free set of arguments S is a *grounded extension* iff it is a minimal (w.r.t. set inclusion) complete extension of AF".

Preferred extension "A conflict-free set of arguments S is a *preferred extension* iff it is a maximal (w.r.t. set inclusion) admissible extension of AF", i.e. iff S defends all of its elements and is as large as possible. All preferred extensions are complete.

Stable extension "A conflict-free set of arguments S is a *stable extension* iff S attacks each argument which does not belong to S [even if some of those arguments do not attack S]", i.e. iff S is a preferred extension that attacks any argument not included in itself.

It is expected that participants are rational and agree about what arguments have been put forward and what attacks exist between those arguments (naturally this does not implicate agreement). In Dung's framework, however, it is also assumed that arguments have the same strength, i.e. that each argument attacks other arguments with the same force and that participants agree about that fact. This may be sufficient for some uses, but it's obviously not a good model of what happens in argumentation between humans. In real life, people may recognize that an opposing argument is valid in its essence (structure and logic) but still disagree about its relevance.

This limitation in Dung's abstract framework has led to the development of many extensions to address the problem of different argument strengths. Bourguet [BAT10] provides a summary of some of those extensions, such as PAFs [AC02] and CPAFs [APP00] (which are briefly introduced in chapter 4), and a new framework unifying their benefits. It is particularly interesting to analyze Bench-Capon's [BC03] extension in the context of this thesis because he incorporates the concept of values (as in social or personal values, not economical or statistical values), which can be used as inspiration for modeling the preferences of different expert agents.

Bench-Capon theorizes that the perception of the strength of an argument is closely related to the concept of social or personal values. He exemplifies this way:

> "(...) two arguments can conflict, and yet both be accepted. For an example suppose that Trevor and Katie need to travel to Paris for a conference. Trevor offers the argument *"we should travel by plane because it is quickest"*. Katie replies with the argument *"we should travel by train because it is much pleasanter"*. Trevor and Katie may continue to disagree as to how to travel, but they cannot deny each other's arguments. The conclusion will be something like *"we should travel by train because it is much pleasanter, even though traveling by plane is quicker."* [BC03]

He draws attention to the fact that the persuasive impact of an argument is largely determined by the audience to which it is addressed. So an argument can be sound from a logical perspective but still be perceived as of little importance. Such an audience consists of one or more people with the same notion of values, or to be more specific, the same preferential order of personal values. In a different work [ABC07], Bench-Capon and Atkinson also indicate a shortcoming in Walton's practical inference scheme (see section 2.2.1) because it does not explain why G is a goal for agent A or how important it is for agent A to bring about goal G, therefore recommending a more elaborate scheme in the lines of "In the circumstances R, the agent should perform action A to achieve new circumstances S, which will realize some goal G which will promote some value V of the agent". Value V is thus clearly stated as the reason why the agent desires to achieve goal G. Each agent can independently choose the order of importance of his values. Nevertheless, during the argumentative process, it may sometimes be impossible to satisfy the values of both agents simultaneously. In such cases, it's necessary for the agents to conciliate their views on the importance of their values in order to agree on a course of action. Bench-Capon also accounts for the fact that some arguments are taken to be true regardless of the preferences of the agents (in the context of the broader example quoted above, he gives the example of a train strike, which makes it impossible to decide in favor of taking the train). Such arguments become associated to the value *Truth*, which is respected by all audiences above all other values.

Bench-Capon's framework, besides providing semantics for acceptability, includes two important definitions:

Value based framework "A *value-based argumentation framework* (VAF) is a 5-tuple: $VAF = \langle A, R, V, val, P \rangle$", where A and R are equivalent to AR (set of arguments) and $attacks$ (binary relation) in Dung's framework respectively, V is a non-empty set of values, val is a mapping of each action to the value it promotes and P is the set of possible audiences (or partial orders on V).

Audience specific VAF "An *audience specific value-based argumentation framework* ($AVAF$) is a 5-tuple: $VAF_a = \langle A, R, V, val, Valpref_a \rangle$", where A, R, V and val are the same as for the definition of VAF, a is an audience (from set P of the VAF) and $Valpref_a$ is an irreflexive, assimetric and transitive preference relation $Valpref_a \subseteq V \times V$, which represents the order of preference of values from the perspective of audience a.

One of the limitations of Bench Capon's theory is that arguments can't simultaneously support multiple values. This problem has been addressed in an extension of his work in [KvdT08]. However, neither framework accounts for the fact that the relationship between preferences or values, in human discourse, is not necessarily linear. Extending Bench-Capon's example of Trevor and Katie, imagine that the only train ride they can afford to take is very slow, such that they would need a couple of weeks to arrive at their destination. Even if the conference they wish to attend is only to be held within a month (therefore taking the train would not implicate missing the conference), would they still prefer taking the train instead of the plane? Probably not, for various reasons. One could argue that this is a matter of inserting more values in the argumentation framework related to the inconveniences or risks associated to taking the slow train. Nevertheless, that would require a difficult process of enumeration. Typical human reasoning would take those factors into account implicitly by saying something like "I prefer to travel by train because it is more comfortable, unless the train is too slow, in which case I prefer to take the plane instead". This isn't possible to represent directly using Bench-Capon's framework or its extension, in part because they don't allow the explicit indication of negative impacts on values, or the representation of the fact that when two arguments promote the same value, one may be preferred to the other if it promotes the value better or with more likelihood of success.

2.2.4 Argumentation and Decision Support Systems: motivation and applications

In real life, people justify their beliefs and decisions based on arguments. Therefore, argumentation seems like the most natural and powerful way to suggest and defend a choice. It quickly becomes evident that DSS could greatly benefit from argumentation-based approaches to justify their suggestions. Recently, this position has been defended by several authors [RS09, MM12] and exemplified with applications in different domains, such as in collaborative engineering design [JG09], public health policy-making [BTMA13] and dynamic supply chain formation [WWV+09].

Girle et al. [GHMV03] provide a set of recommendations to adopt when designing DSS for practical reasoning (i.e. for reasoning about what has to be done in some situation), encouraging the use of argumentation-based MAS and exploring some of the typical, key challenges of DSS design. Their opinion about argumentation and DSS is similar to that described in the previous paragraph and compatible with the concept of using heterogeneous agents specialized in different areas of expertise: "when faced with difficult decisions about what to do, decision makers benefit from good advice. Good advice comes most reliably from advisors with relevant expertise". Girle et al. also offer some recommendations about how advice should be presented to the user.

> "First, the advice should be presented in a form which can be readily understood by the decision maker. Second, there should be ready access to both the information and the thinking that underpins the advice. Third, if decision-making involves details which are at all unusual, the decision maker needs to be able to discuss those details with their advisors". [GHMV03]

The first part concerns what information is relevant to the user (e.g. estimations of performance, comparison between alternatives) and how it should be presented so that it is easily understood (e.g. text, tables, charts, graphs). The second part concerns the system's ability to justify its suggestions, which can be supported by argumentation. The third part concerns the user's ability to interact with the DSS by configuring the inputs, seeking further explanation for certain suggestions (or even rejecting them), supplying more information to enrich the deliberation process, basically, to debate alternatives together with the system.

Regarding these points, Girle and his colleagues offer a list of recommendations for Graphical User Interface (GUI) design in DSS. Without going into detail about all those topics, it seems at least interesting to mention here the necessity to provide users with a clear representation of the arguments that support the decisions suggested by a DSS. Arguments pro and against a decision should be represented both in textual and visual format, since different users have varying degrees of capacity in comprehending text and images. Over the last few years, some software has been developed for representing arguments graphically through diagrams or argument maps (with the purpose of teaching and researching argumentation theory), for example Araucaria [RR04] and Carneades [GPW07]. There is also an area of research devoted to enhancing the acquisition of knowledge through visual methods, closely related to human-computer interaction and to methods for eliciting and modeling expert knowledge [CHCF02].

Girle et al. also provide a discussion on the matter of comparing the projected performance of different courses of action in a DSS. One could think of several criteria for carrying out this task, such as preferring the decisions with greater number of reasons in favor (pros) subtracted by the number of reasons against (cons), weighing the probability of outcomes, attributing importance values to different performance criteria, etc. Classical decision theory [MVN53] proposes a quantitative method for measuring the performance of a scenario by taking into account the probability of each different outcome under that scenario, as well as a number representing how positive (or negative) each outcome would be if it came to be true. Then one uses a criteria like the maximum expected utility to rank potential decisions. Some of the disadvantages of this approach are that, to be useful, it requires prior knowledge of all the possible outcomes of a decision (or, at least, the most likely ones), it requires knowledge of the probability of each outcome and it presumes that the user has a good understanding of the expected utility of each outcome. Most of these requirements do not hold in complex deliberations.

In addition, Girle et al. point to a study in the UK by Stirling and Mayer [SM99] regarding a debate about policies for genetically modified crops involving specialists in different areas of expertise. To rank options, they use a method of multi-criteria evaluation where weightings are assigned to different criteria (e.g. environment, health and economy) and a score is given to each option in light of each criterion. The final rank of an option takes into account the weighted combination of the scores of all criteria. Naturally, as indicated by Girle and his colleagues, this method is vulnerable to the inherently subjective nature of numerical weights and scores; nevertheless, it is one of many possible approaches.

Bourguet's work dealing with argumentation applied to food quality decisions [BTMA13] demonstrates a case of a real-world problem that could be solved by connecting argumentative-based reasoning to DSS. He proposes a model grounded on Dung's abstract argumentation framework [Dun95], taking into account three dimensions of argumentation models: the internal structure of an argument (how arguments are constructed), the relationships between different arguments (such as attack or support) and the varying perception of the strength of an argument by different audiences. The model is applied to a public health policy making context, in particular to the bread manufacturing process, which is a complex problem due to the contradictory interests of different stakeholders, such as public health institutions (which seek to decrease the emergence of diseases), millers (who are concerned about production costs) and bakers (who seek to maintain sensorial pleasure associated to their products). Bourguet concludes by suggesting that DSS involving argumentation models could be helpful to stakeholders for eliciting new arguments and suggestions, thus facilitating communication and making the decision process more productive. It's important to note, however, that because the different stakeholders have conflicts of interest, his is a work about negotiation dialogues, not deliberative dialogues. One of the shortcomings of his approach, as Bourguet himself admits, is the difficulty to find a set of actions that conciliates widely different concerns such as sanitation and economy ("we thus need to enhance the model with aggregation procedures recommending relevant actions in situations where weighted preferences can be elicited and heterogeneous concerns are merged"). Nevertheless, Bourguet's framework allows for the analysis of alternative courses of action in light of each individual concern, similar to Bench-Capon's concept of audiences [BC03].

In collaborative contexts, agents should be interested in maximizing the performance of the system as a whole while respecting the different quantitative and qualitative parameters valued by each other. This concept is well illustrated by Jin [JG09] in a collaborative engineering environment, where each individual is responsible for designing a specific part of a car. The individuals are not self-interested: their focus is on collaborating to provide the best possible design for the vehicle. Conflicts arise, for example, when individuals have different views of quality measures or when a certain design decision by individual A conflicts with a previous design decision by individual B. Such conflicts are solved by searching for mutually compatible solutions or, when impossible, trying to decide which design principles are more important to the overall quality of the car. During this process, individuals often have to explore new design alternatives they had not conceived before. This expands the argumentation process beyond a mere set of conquests and concessions by each individual, allowing them to find new alternative solutions that benefit the performance of the entire system. This a good example of using deliberative argumentation to successfully solve decision-making problems, though the practical implementation of Jin's model is not a DSS but rather a collaboration support system (agents don't argue spontaneously, instead, the system, using agents, suggests possible lines of argumentation to the user).

One could say that, if agents had a good understanding of solutions that worked well in the past, they could more easily select the best positions to defend in new situations. Therefore, in theory, a combination of argumentation-based negotiation with case-based reasoning (CBR) could be used to great effect in DSS. CBR [AP94] is an AI approach for solving problems based on the knowledge gained in previous experiences. To solve new problems, one starts by finding similar cases in the past, and then applying the solutions that worked best for those cases. Every time a solution is applied, its results are preserved so that we can learn from it in future experiences.

Heras [HJBJ13] shows a promising approach by combining in a single framework the concept of argumentation with that of domain-cases (representing previous problems and their solutions) and argument-cases (the success of different argumentation strategies in defeating an opponent's arguments in different situations). While the usage of domain-cases seems to produce better results (compared to not using them), some issues arise when all agents have the ability to use argument-cases, essentially because they all become equally competent in terms of selecting the most persuasive arguments (besides, the focus of the process shifts from finding good solutions to a given problem to finding the best arguments to convince other people, which is, obviously, greatly dependent on the characteristics of the interlocutors). Choosing arguments based on how well they work at convincing others and extrapolating suggestions from that process does not seem to be a good criteria for making neutral, responsible, informed decisions using a DSS. My view is that the usage of argument-cases and competitive argumentation mechanisms in general, while perhaps adequate for implementing competing agents, is inappropriate in most contexts of collaboration, because agents should work together to reach the best solution for the system without pursuing selfish interests (Heras [HJBJ13] implicitly agrees with this view at the conclusions in her article by suggesting the development of a new framework "(...) in a cooperative environment where agents are not interested and collaborate to reach the best agreement for them all"). To conclude, we are interested in the truth value of the arguments, not in the personal benefit of the arguer, the latter being seen by some as counterproductive to argumentation [Joh00].

2.3 Transportation

Applications of MAS in transportation typically concern traffic management and control to prevent (or resolve) congestions [Sch02]. This is part of a series of recent efforts towards improving the efficiency of transportation networks by directly influencing drivers' behavior, instead of carrying out physical modifications to the networks themselves. These approaches are generically called ITS (Intelligent Transportation Systems), and their overall purpose is to provide technological services to help people make more efficient use of the available transportation mechanisms. This requires good understanding of human behavior, in particular the reasoning process of drivers according to external circumstances and their reaction to new information conveyed by such technologies. Rossetti et al. [RBB+02] have proposed an approach for representing and assessing driver behavior by means of MAS with BDI (Belief–Desire–Intention) architectures. A follow-up on this work introduces a methodology for assessing the impact of pre-trip information systems on the cognitive process of drivers and its effect on overall network performance [RL05]. Another example of an application of MAS-based technologies in transportation is solving Vehicle Routing Problems, i.e. optimizing the service of several customers with a limited fleet of vehicles [BBG12].

The domain of transportation, and urban transportation planning in particular, is far too broad to be covered within this chapter with anything more than an introductory level of detail. Over the following subsections, I'll only touch upon topics regarding tools and techniques used in planning, simulating traffic networks and assessing network performance.

2.3.1 Technology in transportation planning

As far as I could ascertain, there are no documented practical applications of MAS with the purpose of directly assisting transportation planning processes (i.e. the actual process of designing transportation networks). Nevertheless, there is a long tradition in this area of using software tools to build simulation models and devise alternative traffic scenarios, e.g. using GIS (Geographic Information System) based decision support tools.

The main challenges related to transportation planning in the modern world have already been briefly presented in chapter 1. If the reader wishes to know more about this topic, the United States

Federal Transit Administration [2] [3] provides a vast selection of quality literature to the public (see footnotes and TPCB [Tra07]). This section is more concerned with summarizing the technological applications typically used in transportation planning.

According to [Tra07], for many decades, transportation planners have used a so-called "four-step" approach in modeling transportation demand. Before being able to employ this approach, planners have to understand the land use in a region (and its estimates for the future), as well as demographic and socioeconomic factors (e.g. population clusters, average household size, employment types). Once this is understood, planners use the "four-step" approach (trip generation, trip distribution, mode split and network assignment) to estimate the total number of trips through each link of the transportation network. This allows planners to forecast congestion at each link, which can be used to provide an estimate of the system's performance.

Emission levels can also be forecast by using a separate model that takes the results of the "four-step" model as input. Additionally, planners may use a variety of technologies to help stakeholders visualize transportation plans in an accessible way, as well as scenario planning tools to understand the impact of alternative decisions in the future.

It seems evident that the usefulness of technology is well understood by the professionals of the transportation planning sector. Therefore, it should not be difficult to explain the advantages of using a single system that encompasses the whole decision-making process. Using arguments to explain the merits of potential solutions seems like a good way to help both planners and stakeholders comprehend the pros and cons of different alternatives. Using agents as a development paradigm seems adequate to handle the size and multidisciplinary traits of the problem.

2.3.2 Simulation tools for transportation

A proper DSS in the context of transportation needs a good traffic simulation engine to estimate the implications of different design decisions on the real world with a reasonable degree of confidence.

Many traffic simulators exist, with different advantages and disadvantages. Most can be integrated or communicate with other applications via dedicated APIs. Some examples are Quadstone Paramics[4], MATSim[5] and SUMO[6]. A tool in development by researchers at Faculdade de Engenharia da Universidade do Porto called TraSMAPI [TARO10] aims to provide a common API to observe and interact with different traffic simulators in an accessible manner, hiding low level communication details from the user. The obvious advantage of using such a tool is to allow researchers to spend more time on actual research and less time on implementation details (like adapting the code to run on different simulators). A documented example of the use of this tool in combination with JADE and SUMO can be found in Azevedo [AARR14].

SUMO [KEBB12] is an open source microscopic traffic simulator, providing information and control over each separate vehicle in the simulation model independently. It supports multimodal

[2] http://www.fta.dot.gov/15576.html
[3] http://planning.dot.gov
[4] http://www.paramics-online.com
[5] http://www.matsim.org
[6] http://sumo-sim.org

traffic systems including personal vehicles, public transportation and pedestrians. Traffic lights can be freely positioned and controlled. It also provides an OpenGL based GUI for real-time visualization and an API called TraCI to allow other applications to remotely control the simulation. Additionally, SUMO supports several extra features, such as importing road networks from different formats (including Open Street Maps), automatically planning routes, calculating noise and particle emissions and generating traffic flow statistics.

SUMO allows users to define their own traffic networks using a combination of mandatory and optional configuration files containing road definitions (edges), intersections (nodes), traffic lights, vehicle types, vehicle routes, emission detectors and others. The SUMO distribution also contains many programs and scripts to assist users in configuring networks and analyzing simulation results. In addition, researchers have compiled ready-to-use simulation models containing vast amounts of data obtained from real traffic networks, which can be easily used to study the impact of different traffic-related measures on the performance of a network. One of those scenarios is based on the city of Cologne, Germany [Ins] and another is based on the city of Bologna, Italy [BKM+14].

SUMO has been used in several multipurpose studies, from vehicle-to-vehicle and vehicle-to-infrastructure projects [iTE10] to forecasting traffic congestions during major social events. One particularly interesting study uses SUMO to analyze the impact of different measures for the reduction of air pollution in the city of Brunswick, Germany [TKGD13]. In this study, various tools of the SUMO suite are used together to simulate the enforcement of an environmental friendly zone in the city (where only low emission vehicles can pass) and the reduction of the speed limit in certain areas. It is concluded that implementing the environmental friendly zone causes a large reduction on the emissions of particulate matter (PM_x) and oxides of nitrogen (NO_x) both in the area of application and the city as a whole, while the reduction of the speed limit causes the same effect locally but deteriorates the global emissions because impatient drivers choose roads with higher speed limits.

2.3.3 Assessing network performance

Many different metrics exist to evaluate the quality of service of transportation networks. Such variety stems not only from the necessity to take into account different quality parameters (e.g. environmental, financial, legal) but also the application context (metrics for freeways are different from those for urban facilities and rural areas) and user perspective (a pedestrian demands different services from the transportation network compared to an automobile driver).

To determine how appropriately a road serves traffic demand, the United States Transportation Research Board publishes a very detailed, regularly updated manual for use by analysts and transportation engineers: the Highway Capacity Manual [HCM10]. Should the reader wish to know more about the broad topic of transportation planning, this is a good manual to study. It explains a set of methodologies for evaluating the capacity of freeways, multilane and two-lane highways, urban facilities (i.e. major urban arteries like avenues) and urban segments (i.e. a segment of road between two consecutive intersections), intersections with and without traffic lights, roundabouts and other network facilities. The perspectives of automobile drivers, pedestrians, bicyclists and transit passengers (public transportation users) are all taken into account. There are some important base concepts.

- **Level of service (LOS)**: it is one of the major concepts involved in the understanding of a network facility's quality of service. It is typically used by analysts to more easily convey their observations to non-specialized stakeholders. LOS has six levels, ranging from A to F. A represents exceptionally good service, while F stands for the worst possible service;
- **Free flow speed**: the average speed of vehicles running on a portion of the network when no obstacles are present. Although similar to the concept of speed limit, free flow speed also takes into account the diminished velocity when taking turns and riding curvy roads.
- **Control delay**: the average amount of time that a vehicle has to wait at a roundabout or intersection (with or without traffic lights) before being able to proceed;
- **Volume-to-capacity ratio**: the quotient between demand in a given traffic facility and its maximum capacity (in vehicles per hour) before traffic jams occur. Values greater than 1 signify that the capacity of the road is exceeded and traffic congestions will occur. A road's capacity can be estimated from observation or approximated using the road's speed limit, average car length and average distance between cars, using some formula similar to 2.1.[7]

$$Capacity = \frac{maxSpeed}{avgTimeDistanceBetweenVehicles * maxSpeed + avgVehicleLength} \quad (2.1)$$

According to the Highway Capacity Manual, for urban facilities and road segments, the values of table 2.1 apply when computing LOS.

Table 2.1: LOS for urban facilities and road segments

Average travel speed as a percentage of free flow speed (%)	LOS by volume-to-capacity ratio	
	≤ 1.0	>1.0
>85	A	F
>67-85	B	F
>50-67	C	F
>40-50	D	F
>30-40	E	F
≤ 30	F	F

The values of table 2.2 apply when computing LOS for intersections with traffic lights (signalized), intersections without traffic lights (unsignalized) and roundabouts. Control delay values are higher for intersections with traffic lights because drivers expect the existence of lights as indicators of roads with higher volumes of traffic (causing greater delays). Also, since drivers are in control at intersections without traffic lights, delays become less predictable, reducing tolerance.

[7] As an example, for a road with speed limit of 50 km/h, considering an average time distance between vehicles of 2 seconds and an average vehicle length of 5 meters, the estimated capacity is of approximately 1525 vehicles per hour.

Table 2.2: LOS for intersections with and without traffic lights and roundabouts

Control delay by junction type (seconds per vehicle)		LOS by volume-to-capacity ratio	
At intersections with traffic lights	At intersections without traffic lights and roundabouts	$\leq$ 1.0	>1.0
$\leq$ 10	$\leq$ 10	A	F
>10-20	>10-15	B	F
>20-35	>15-25	C	F
>35-55	>25-35	D	F
>55-80	>35-50	E	F
>80	>50	F	F

Some of the most damaging impacts of a network facility on the environment are the decrease in air quality, which leads to respiratory conditions and smog; greenhouse gas emissions, which cause global warming; and noise, which, above certain levels, has negative effects on the daily life of a city's inhabitants.

The European Parliament and Council have established regulations with the purpose of keeping air pollutants in the atmosphere below certain levels to preserve human health and vegetation [Eur08]. They also enforce limit levels for vehicle manufacturers, regarding both air pollutant emissions [Eur07] (called Euro standards, the last of which is Euro 6), and greenhouse gas emissions [Eur09]. Computational models have been developed with the purpose of describing average emission factors for different classes of vehicles under several operating conditions, such as the Handbook Emission Factors for Road Transport, also known as HBEFA [INF]. This model is implemented in SUMO.

The legislation for maximum noise levels is not so well defined, but there are studies from accredited authorities, like the United States Environmental Protection Agency, concerning safe levels for the protection of human health (both indoors and outdoors), some of which date back to the 1970s [U.S78]. In Europe, maximum noise levels are not enforced at a federal level, but some directives [Eur02] suggest the need to develop methodologies for the assessment and management of environmental noise. One of the results of those directives was the development of the Harmonoise model for the estimation of noise caused by road traffic and other transportation infrastructures such as train rails. Like HBEFA, the Harmonoise model is implemented in SUMO.

2.4 Summary

This chapter provided a literature review on the topics of MAS, argumentation-based negotiation and transportation planning, with special emphasis on the foundational literature about abstract argumentation frameworks and deliberation dialogues, as well as the relationship between those concepts and decision-making. The review covered studies about the structure of arguments, different types of dialogues in argumentation, acceptability semantics, recommendations for the development of good DSS, new approaches in the area, and several other subjects.

Given the focus of this thesis on argumentation, the various topics covered in section 2.2 will be of particular usefulness to the reader's understanding of the remaining chapters. Nevertheless, in punctual cases, I will introduce certain bits of useful literature directly within subsequent chapters, as well as some short recaps, with the purpose of facilitating comprehension.

Chapter 3

Preliminaries

The purpose of this chapter is to provide an overview of the proposed solution that will be explained in detail throughout subsequent chapters. Hopefully, this will allow the reader to feel more comfortable as I delve into the specifics of my theory. In addition, I'll introduce some of the key concepts that will be addressed in the specification of my argumentation model.

3.1 Overview of the proposed solution

It is helpful to think of my theory applied in the context of a DSS backed up by a MAS. Such a DSS may consist of a GUI where the decision-maker defines the inputs of a problem to solve and any constraints on its resolution. The program fires up an agent, which from now on I'll designate *initiator agent* or simply *initiator*, who acts on behalf of the user's interests. At the beginning, the initiator has the responsibility of searching for *expert agents*, or simply *experts*, in the agent network. Such experts may specialize in different domains of knowledge involved in the resolution of the problem defined by the human user.

For example, if we were dealing with a problem in the context of urban transportation networks, we might be interested in the opinions of a traffic manager, an environmentalist, a civil engineer, and so on. To take full advantage of the MAS paradigm, agents are not necessarily expected to exist in the same machine. For instance, different organizations could be responsible for building different expert agents, e.g. an environmental protection agency could be in charge of implementing and continuously improving an environment expert agent, while a traffic administration agency could be responsible for implementing a mobility expert agent. An administrative organism, such as a local government, could use a DSS which, through an initiator agent, would start a deliberation process between different expert agents, supplying the design and expected traffic demand of a network as input for the experts and gathering suggestions for the potential improvement of the network. A complete example along these lines is presented in chapter 7, making full use of my theory.

Once the initiator agent gathers an assembly of expert agents and conveys the inputs of the problem, each expert will define a number of goals to achieve, consistent with the personal and social values that he seeks to uphold. In the context of the previous example, a traffic manager would likely concern himself with assuring stable traffic flow in the network, while an environmentalist would attempt to avoid air and noise pollution. Obviously, all experts will share the concern of resolving the underlying issue indicated by the user.

Next, each expert will propose a number of potential decisions to solve the problem at hand while attempting to fulfill his goals and, if possible, the goals of other expert agents. Each decision will likely have different advantages and disadvantages, with varying importance depending on the perspective of each expert. Experts exchange such considerations using arguments. Those arguments may be grounded on the observation of some relevant properties of the problem, the consequences of undertaking a decision, the impact of such consequences on different goals, etc. If those grounds are incorrect, agents may attack each other's arguments until they establish what is true. Determining what arguments are acceptable and what arguments become invalidated requires an argumentation framework, which is explained in chapter 4.

After consideration of the acceptable arguments and the relative importance of goals, each expert agent will report his preference order on decisions back to the initiator agent, who gathers these opinions and also attempts to condense them into a general recommendation. Establishing preference is a complex topic, and is dealt with in chapter 5. The considerations of each individual expert agent and the group as a whole are then presented back to the user. A simplified diagram summarizing the interactions outlined in this section can be observed in figure 3.1.

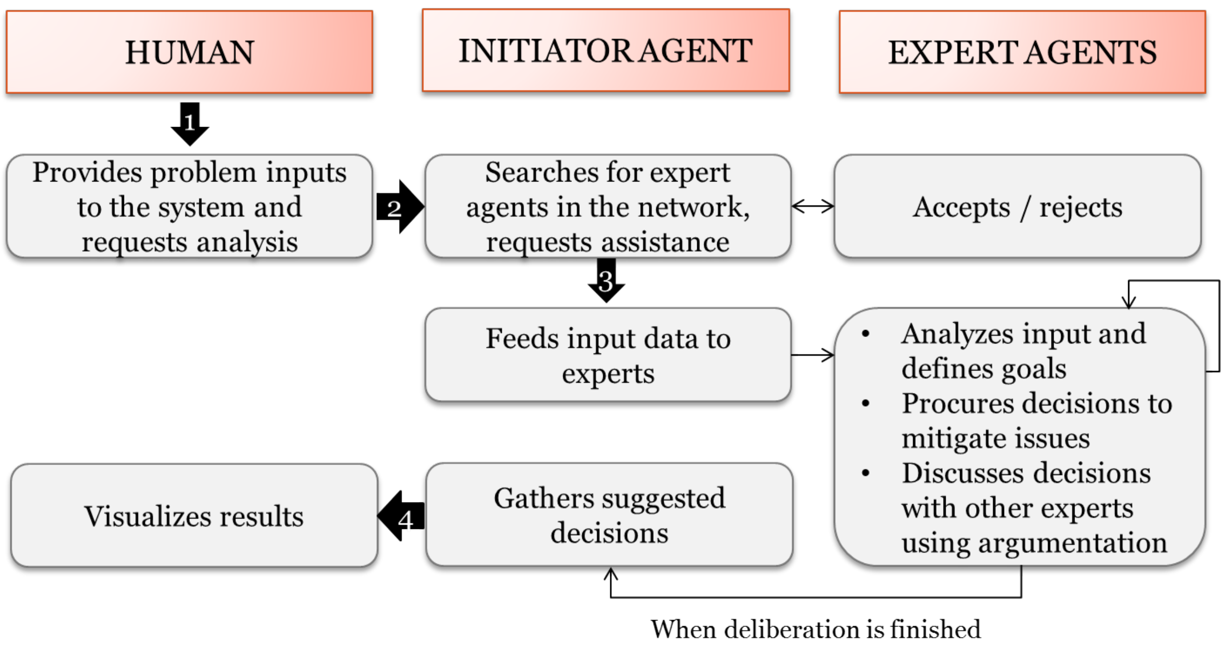

Figure 3.1: A simplified diagram of the deliberation process

Notice how I touched upon a series of generic concepts here: goals, values, decisions, consequences, etc. All these relevant concepts, which are common to any deliberation dialogue and thus must be accounted for in my argumentation model, are presented in detail in section 3.2. Also, if the initiator agent and the expert agents are to be able communicate with each other, they must abide to some form of protocol that is built upon my argumentation model. This protocol is presented in chapter 6.

3.2 Key concepts

There are several key concepts common to any decision-making deliberation dialogue, independently of the context on which it might take place. Those concepts must be accounted for in my generic argumentation model. This section presents each one of those key concepts, from the perspective of generating arguments in favor or against potential alternative solutions to a problem, introducing some rules for decision assessment and dialogue participation which will be discussed in detail in subsequent chapters.

Governing question

A deliberation process usually starts by the exposition of some problem to solve in the form of a governing question, such as *"How shall we decrease the frequency of accidents in road X?"* or *"How shall we improve traffic flow in highway Y?"*. Typically, this is an open question, allowing for the discussion of different, not always consensual, alternatives.

If there is more than one question to solve, several deliberation dialogues may be carried out in succession, unless the resolution of one question is presumed to have an impact on another, in which case they should be considered together (reformulated as a single question).

In a DSS, the exposition of the problem can be carried out by the end user (for whom the initiator agent will act on behalf of) or by the expert agents themselves through observation of a representation of the world (potentially using simulation) which enables autonomous problem discovery.

Initial circumstances of the problem (properties)

By initial circumstances we understand a set of properties obtained from the observation of the problem, i.e. from the representation of the state of the world, before any corrective actions are undertaken. Some examples in the context of transportation networks could be *"Speed limit in road X is 90km/h"* or *"Road Y has no public illumination"* or *"Average CO_2 emissions per vehicle in region Z is 145g/km"*. The reason for starting a deliberation process could be related to one or several initial circumstances considered to be negative.

Properties can be of different kinds, for example:

- **True or false**: *"Road X has public illumination - has_illumination(x, true)"* or *"Road Y has no public illumination - has_illumination(y, false)"*;

- **Numeric**: *"Average CO_2 emissions per vehicle in region X is 145g/km - co2_g_veh(x, 145)"*;

- **Enumeration**: *"Road X is of type highway - road_type(x, highway)"*.

These properties amount to the input of the problem, and as such must be supplied by the initiator agent, leaving no reason to question their trustworthiness. Some complex properties may be derived from these elementary ones by the expert agents, using rules or mathematical formulas that are part of their base knowledge, which in turn may be questioned for their lack of legitimacy by other experts (this issue is approached in the definition of the next concept).

In certain application contexts, some initial circumstances may have an associated level of uncertainty. For instance, in a medical DSS for determining the best drugs to prescribe to a patient depending on his disease, the amount of certainty on the fact that the patient truly has the disease under consideration is often not complete (depending on the patient's symptoms, errors in his diagnosis, etc).

Initial circumstances can also be extracted using simulation with a representative model of the problem, so long as the initiator agent provides that model (or if he provides the raw simulation results, at least).

Base knowledge

Independently of the problem under consideration, expert agents possess some base knowledge consisting of generally accepted information or well accepted scientific knowledge within their domain of expertise. Base knowledge can be used in argumentation for many different purposes, such as expressing the premises for believing the consequences of an action, expressing the effects of such consequences on goals, indicating the existence of some properties in the state of the world, criticizing the beliefs of other participants in the debate, and defending one's point of view.

This knowledge varies among agents. It isn't necessary for expert agents to share all their base knowledge amongst each other, only that which is considered relevant to deliberate about the problem in study. Any kind of knowledge presented by an expert agent may be questioned and debated with his colleagues. Furthermore, the initiator agent may provide base knowledge acquired from the human user, which will be assumed as trustworthy by the agents. Similarly to the initial circumstances, some base knowledge may have an associated level of uncertainty, which must be indicated by the agent that presents it.

What follows is an example of some types of knowledge that an agent might possess:

- **Facts and rules**, representable, for instance, using first-order logic;
- **Mathematical formulas and algorithms** to evaluate the problem's properties, i.e. metrics;
- **Argumentation schemes** [Wal13] for generating arguments in favor or against some perspective. Each argumentation scheme is associated to a set of critical questions to expose typical vulnerabilities. During the deliberation process, generating new arguments may be useful to increase the knowledge base by drawing information from their conclusions;
- Knowledge obtained from records of previous decision processes, or **CBR**. Each record must keep the initial circumstances of the problem, the set of actions undertaken and the consequences of those actions considered relevant. It's up to the expert agents to check if the initial properties of some of those problems are similar to the ones under consideration, in which case they might be useful to discover (or stay away from) potential alternative solutions. This makes it possible for agents to benefit from knowledge about previous experiences, which is often better accepted than facts and rules *a priori*. There are some typical forms of attack against the appropriateness of employing a registered case:
  - If the initial circumstances of the case aren't similar to those of the problem;
  - When there are other cases with similar initial circumstances and suggested actions but different consequences (indicating that perhaps the success of the case can't be generalized);
  - When there are other cases that achieved similar consequences (or a better superset), starting from the same initial circumstances, by taking a different set of actions. This is not so much a form of criticism but rather a suggestion of an alternative set of actions.

There is one other type of knowledge that isn't known at the beginning but can be acquired during the deliberation process, which can be very useful to decide on what actions to suggest to the user. It consists in observing the results of a simulation by applying the set of actions under consideration to the initial model, in order to study their effectiveness. If the changes prove to have a positive impact, this type of knowledge will probably be better accepted than the other types. There are ways to contest this information.

- When the simulated dataset is too small (in size or time) to be considered conclusive (simulating a model which isn't representative of reality);
- When the phenomena observed in the simulation are caused by other changes in the model besides the suggested set of actions (poorly configured simulation);

Although simulation allows for great confidence in the performance of the suggested actions, it has the disadvantages of being a slow process (which may or not be a problem depending on the scenario) and requiring a complex modeling activity prior to the consideration of the problem.

Concerning the first issue, it is recommendable that expert agents only resort to simulation when they have to choose between several promising courses of action.

Actions and decisions

Actions are operations which can be carried out in the real world to solve problems. Each action may trigger a set of consequences, some positive, others negative. The consequences of an action may vary according to the initial circumstances of the problem. For example, placing traffic lights in a low traffic street will not have the same effect as in a high traffic street. In the former case, traffic flow will likely decline, in the latter, the extension of traffic queues will be reduced.

Agents may question if, in light of the initial circumstances, an action under consideration does indeed trigger the consequences foreseen by the agent in favor the action, or if there are other (potentially more beneficial) actions which would bring about the same consequences or accomplish the same goals. The practical feasibility of an action may also be questioned.

In some decision contexts, such as in the domain of transportation networks, deliberations may have to take into account the existence of infrastructures *a priori*, such that it may be preferable not to enact any actions, otherwise the performance of the system may actually decline. This contrasts with other types of decisions where something must necessarily be done to mitigate the problem. Also, it might be that an action which does not satisfy the minimum requirements of the decision makers is still preferable to taking no action, given that it improves the state of things a little (assuming, of course, the lack of better alternatives).

Executing an action might preclude the execution of other actions or render them useless. For example, *"preventing the circulation of vehicles on road X"* and *"placing parking spaces on road X"* can't be combined rationally. Therefore, two actions may be considered mutually exclusive (i.e. incompatible), which could be grounds for arguing against an action if it prevents the execution of another that is presumed to be beneficial. On the other hand, some actions may be complementary or independent (i.e. compatible), resulting in an accumulation of effects which may or not be linear.[1]

When an expert agent puts forward an argument in favor of an action or set of actions, an attack (i.e. another argument) against any assumptions he might have made about the problem in question (initial circumstances, consequences, relationship between consequences and goals, etc.) will invalidate the argument if it is not defended. Sometimes, there is a dependency between the components affected by an attack. For example, if the initial circumstances of the problem are questioned, the presumed consequences of the action will be invalidated, unless further arguments are exchanged. If the proponent of the action agrees with the attack, he will have to reevaluate the performance of the action in light of the new initial circumstances.

[1] As a simplification, one should assume that in those cases where two actions only make sense when carried out together as part of a sequence, they form a single action. For instance, contrast the single action *"serving a cup of coffee"* with the sequence *"brew coffee"*, *"get spoon"* and *"get sugar"*.

Before the end of the deliberation process, each agent will consider, among the potential actions, which ones satisfy the minimum requirements according to his goals, and among those he will indicate his order of preference (using decision criteria which are discussed in chapter 5). The best actions are the ones that simultaneously satisfy the minimum requirements of all agents, which, of course, may be impossible to find.

A combination of complementary (i.e. non-mutually exclusive) actions is called a decision, or, to be more precise, a decision is a set of zero (because one might decide to do nothing), one or several elementary actions. The term "elementary action" is a redundancy for emphasizing the concept of a single action in contrast with a combination of actions - a "decision". This distinction is important from a formal perspective, because it allows one to discuss the possibility of employing a variable number of actions simultaneously, instead of only being able to consider each action on its own. While I often use the terms "action", "set of actions" and "decision" interchangeably in this chapter to ease comprehension, in subsequent chapters I'll frequently rely on this distinction.

When the deliberation process finishes, the system outputs the recommended decision to the user. The system may also present alternative decisions which were less consensual among the experts or which might be more desirable if there was a different priority of goals. Regardless of achieving consensus, the DSS should always present the preferred decisions from the perspective of each expert, in order to provide additional information to the decision makers.

Consequences of the actions

Carrying out an action typically results in a modification of the circumstances of the problem, i.e. a modification of some initial properties of the state of the world or an addition of new properties. As mentioned before, the consequences of an action may vary according to the initial circumstances of the problem. Some consequences may be desired, others undesired, others unimportant, depending on the goals of each expert (i.e. the assessment of the consequences varies according to perspective). One of the major reasons for an agent to be in favor of some course of action is that it would (presumably) have consequences which would satisfy his goals (and, if possible, those of other agents), without violating any established restrictions. Other agents may question if those consequences truly satisfy the goals mentioned by the supporter. One could also question if the action has other consequences which were not foreseen and whether those consequences have a positive or negative impact.

The belief that an action, under some circumstances, will bring about determined consequences, is part of the base knowledge of the agents. Therefore, consequences may be affected by uncertainty, and agents may need to review different sets of consequences for the same action with varying degrees of plausibility. This is where the expression of the amount of certainty in the initial circumstances and base knowledge becomes useful.

Goals

Goals are properties that one seeks to attest in the system under study, which, in the initial circumstances, may or may not be fulfilled. As mentioned before, carrying out an action triggers a set of consequences which may favor or disfavor certain goals.

Each expert agent is responsible for a set of goals to uphold. Agents must inform other colleagues of those goals. Additionally, agents must define a priority order on their goals, which will help them choose the alternative options which are more in favor of their points of view. An agent may view several goals as equally important.

Experts compute the satisfaction degrees of their goals using a metric or a combination of several metrics (see next subsection). For example, a goal may have a set of numeric intervals (strata) of satisfaction degrees, representable by a score ranging from F to A, including a minimum value which translates to C. One can arrive at such scores using metrics. Other agents could question if the requirements for achieving such scores are too restrictive, which in some cases could be sorted out through a negotiation dialogue.

Each goal must have, at least, a numeric value that represents its minimum requirement for satisfaction. Additionally, an agent can distinguish between different degrees of satisfaction below and above the minimum, including complete satisfaction and dissatisfaction. This concept can be generically represented using a bipolar scale of real numbers, where 0 stands for the minimum requirement for satisfaction. Values below 0 signify that the goal is not fulfilled, 0 means that the goal is fulfilled, while values above 0 signify that the goal is fulfilled beyond the minimum requirements. We can see that this is compatible with a great variety of situations.

- Bipolar scales with discrete qualitative values. These scales are employed in a wide range of human decision-making contexts and, because they are split into discrete intervals, they have the advantage of minimizing the impact of small numeric differences between satisfaction degrees when comparing decisions. For example, consider a scale with values "terrible", "bad", "sufficient", "good" and "very good", with "sufficient" standing for the minimally satisfying level. This can be represented by a scale of numbers with admissible values -2, -1, 0, 1, and 2;

- Utility measures without bounds, such as money. For example, consider the goal of requiring at least 1000$ worth of sales to turn a profit. This can be represented as a bipolar scale where the satisfaction degree is the difference between the expectable sales and 1000$, e.g. 5000$ of expected sales scores 4000, 200$ scores -800, and 1000$ scores 0 (minimum satisfaction);

- Purely opportunistic or secondary goals, whose fulfillment brings about an extra positive outcome, but whose lack to comply does not cause any harm. These can be represented by a scale with no negative part, i.e. starting at 0;

- Situations to be avoided, i.e. pure restrictions, where only two scenarios are possible: either the restriction is violated or it isn't. These can be represented by any two numbers equidistant from 0, e.g. -1 and 1. Violating the restriction amounts to scoring -1, the contrary amounts to 1. This is also compatible with a qualitative approach of decision-making using only pros and cons (ignoring variable amounts of satisfaction), where 1 and -1 could stand for argument pros and argument cons, respectively.

For an expert, a set of actions is only satisfying if it fulfills the minimum requirements of all the goals for which said expert is responsible, otherwise he won't fully accept it. The need for accomplishing all proposed goals is not as restrictive as it may seem, since the agent is free to define very relaxed minimum requirements. In the event there are two (or more) minimally satisfying sets of actions, an agent will prefer the one that favors his goals with greatest priority. The agent will also prefer actions that satisfy his goals to a greater extent, instead of merely abiding by the minimum requirements. Another aspect that the agent may account for is the uncertainty level associated to the consequences of the actions. Each expert is only responsible for attesting the fulfillment of the minimum requirements of the goals he proposes. The satisfaction of other experts' goals is not crucial for determining if a decision is good, but it can be useful to decide the preference order between alternative decisions. This is a way to take into account the concerns of other experts without placing one's own concerns at stake. These and other issues are discussed in detail in chapter 5.

Some authors defend the distinction between the concepts of goal and restriction, seeing goals as objectives to fulfill and restrictions as negative aspects to avoid at all costs, with greater importance than goals. In all examples of restrictions given by those authors, I didn't find one that couldn't be represented by this system of goals with different degrees of priority and different levels of satisfaction, which, in my opinion, makes representation and computation simpler without losing the power of restrictions. For instance, one could distinguish between these two concepts using the example of a person with the goal of either going to the opera or to a concert and avoiding staying at home (the latter being a restriction). Going to the movies respects the restriction, but doesn't satisfy the goal. Using the system proposed in this work, we can formulate the person's wishes with a single goal, which is *"going out"*, whose minimum requirement is *"going somewhere independently of the destination"* and whose maximum satisfaction is *"going either to the opera or to a concert"*. These alternatives easily translate to a bipolar scale, where *"staying at home"* amounts to scoring -1, *"going somewhere independently of the destination"* amounts to scoring 0, and *"going either to the opera or to a concert"* amounts to scoring 1.

Metrics

Metrics are a (usually numeric) method for assessing the satisfaction of goals. They allow experts to check the conformity of the initial circumstances in light of one of their goals. Metrics can also serve the purpose of assessing the consequences of an action w.r.t. some goal by using some rule to forecast their effects, or simulation if possible, and then analyzing those effects. Goals may be evaluated by several metrics, potentially with varying levels of importance, combined in a linear or non-linear fashion. In such cases, the only requirement is that the combined result conforms to a global evaluation of the goal where the value 0 signifies achieving minimum satisfaction, as explained in the previous subsection. It is up to implementers to decide what criteria to use, e.g. the average value of all metrics, having different degrees of importance for each metric or assuming that the fulfillment of the goal is below 0 if any metric falls below 0.

Metrics are based on the knowledge of some relevant properties from the state of the world, as well as mathematical formulas or algorithms to process those properties, in order to extract some kind of meaning from them if necessary. In other words, to evaluate the satisfaction of a goal according to a metric, agents must take measures of some properties of the state of the world, which in turn may be subject to disagreement.

Values

A value, in the sense of personal or social value (instead of quantitative value) can be seen as a grouping of goals with similar high-level purposes.

Each goal is related to a value. For instance, the goal *"quit smoking"* is related to value *"health"*. The value *"health"* might be related to goals *"quit smoking"* and *"lose weight"*. One can think of values as high level responsibilities. When an agent proposes a goal, he should indicate its associated value in order to facilitate the comprehension of its importance by other agents (who may in turn question if the goal is indeed related to the specified value).

Although based on the same principles, the practical use of the concept of values in my framework is very different from that of value-based argumentation frameworks [BC03]. In such frameworks, the explicit rationale for taking some action is based on the promotion of a personal or social value. The concept of values, in its original sense of some subjective idealistic vision, is probably a little too abstract to be easily modeled in most practical reasoning contexts, as it pertains more to philosophical, ethical and political issues than everyday situations. Here, I prefer to relate the direct benefits of actions to the concept of goals, which is a more objective notion and as such, more easily measurable. This is not merely a question of semantics, because values have a purpose in this framework as well, just not so formal. Moreover, none of this goes against the view that values should be the primary motivation of the participants in a decision-making debate; on the contrary, this is one of the major underlying principles of my argumentation model.

Each expert agent defines an order of importance of his personal values. Unlike goals, no two values may be equally important. In a sense, one could say that the role of the expert is to embody the requirement for upholding some relevant social value in a decision problem. For instance, an environmentalist would likely prefer the value *"Environment"* to *"Traffic Flow"*, in contrast with a traffic manager. Both may agree, however, that value *"Public Safety"* is more important than all the others. It is up to the experts to define the proper objective goals to satisfy the values they promote. This clarifies the usefulness of values in establishing a clear, easy to follow modeling process, becoming a driving force for the implementation of expert agents. To sum it up, each expert is responsible for upholding a particular value (or several, with varying preferences). The fulfillment of that value requires the definition of specific goals, and each goal is evaluated by at least one metric.

Values are also useful to help establish the priority order of goals proposed by other experts, facilitating the process of tiebreaking several satisfying decisions. This means that, when an agent becomes aware of another expert's goal, he accommodates it in his priority order of goals in a way that appropriately takes into account the importance of the value promoted by that goal, in light of his preferred values. This doesn't make the agent responsible for the fulfillment of the goal; it is simply a mechanism for taking into account the concerns of others without disregarding one's responsibilities. A suggestion of a formal method for carrying out this process is explained in detail in chapter 4.

A DSS may also automatically present alternative decisions that are more in favor of specific values. These alternate options should be suggested by the agents that promote each of those values. In this way, users can not only obtain suggestions of decisions which are satisfying to all, but also become aware of the preferences of audiences with specific concerns.

3.3 Summary

I started by presenting an overview of the proposed solution that will be developed throughout subsequent chapters, and proceeded to define a series of key concepts which are at the core of my theory.

Each expert agent embodies the requirement for upholding some relevant social **value** in a decision problem. The fulfillment of that value requires the satisfaction of certain **goals**. Each goal is assessed by a **metric**, or a combination of several metrics. Such metrics assess the **consequences of actions** w.r.t. the goals they concern. The need for **actions** is warranted by the existence of an issue, conveyed through a **governing question**. That issue arises from some undesirable **initial circumstance** of the state of the world. The agents' **base knowledge** provides them with the necessary means to study potential decisions to solve the problem.

Chapter 4

Arguments: structure, attacks and acceptability

In the last chapter, we looked at some of the major concepts underlying deliberation dialogues and decision-making problems. Now I need to establish a method to formally evaluate the quality of candidate decisions, based on the exchange of arguments. For now, I will focus on defining an abstract argumentation framework, which deals with the issues of representing conflicts between opposing arguments, deciding which ones are acceptable and enabling the comparison of decisions grounded on accepted arguments, a topic which will be further developed in chapter 5.

4.1 Arguments about beliefs and arguments about decisions

Some authors in argumentation theory, such as Prakken [Pra06], commonly distinguish between arguments about beliefs, known as epistemic arguments, and arguments in favor or against decisions, known as practical arguments. Here, I choose the same approach.

4.1.1 Definition of epistemic arguments

Epistemic arguments are those that justify or contest the beliefs of the agents. An agent's beliefs may consist, for example, of the initial circumstances of the state of the world, a combination of facts and rules of his base knowledge, the consequences of taking some action, the effect of such consequences on certain goals, etc.

Epistemic arguments can be used for two purposes:

- **Supporting practical arguments**, i.e. as premises for accepting a belief that is part of the reason for supporting or refuting some decision;

- **Attacking other epistemic arguments**, i.e. disputing the beliefs supported by other epistemic arguments. Because those arguments may, in turn, be elements of the set of epistemic arguments supporting some practical argument, epistemic arguments may indirectly attack practical arguments.

The structure of epistemic arguments may vary greatly according to the context and purpose of each implementation. This work is much more focused with providing a formal structure for representing and assessing practical arguments. Generally, as far as abstract frameworks go, the structure of epistemic arguments is not relevant: it is their interactions with other arguments that matter, so implementers can employ whichever theory better fits their purpose (including, for example, Toulmin's model, argumentation schemes, etc). Nevertheless, practical arguments are supported and indirectly attacked by epistemic ones, so I must at least suggest some kind of generic definition. One could say, on a general basis, that an epistemic argument is composed of a set of premises and a conclusion.

Definition 1 (Epistemic argument): An *epistemic argument* can be defined as a tuple:

$EA = \langle P, C \rangle$; where:

- P is a set of premises which are relevant to the conclusion;
- C is the conclusion, a belief, which may be drawn from the premises, for instance, by using rules of inference or argumentation schemes.[1]

The origin of attacks between epistemic arguments is a complex issue which is not relevant to an abstract argumentation framework, i.e. the nature of the attacks is irrelevant, what is important is to know what attacks exist so that one may find out which arguments are acceptable. Suggestions for generating attacks in the form of critical questions were discussed in section 3.2. Speaking in very broad terms, one argument attacks another when its conclusion goes against some belief that is premises to the other (undercut), or when the conclusions of the two arguments contradict each other (rebut).

Additionally, epistemic arguments may have different strengths or preferences, which in turn are perceived in different ways by different agents. Before going into further detail about this concept, it is helpful to introduce two important approaches concerning abstract argumentation frameworks with preferences: PAFs [AC02] and CPAFs [APP00].

- **PAF:** A preference-based argumentation framework is a tuple: $PAF = \langle A, R, \succeq \rangle$; where A is a set of arguments, $R \subseteq A \times A$ is a binary relation representing attacks between arguments, and $\succeq \subseteq A \times A$ is a (partial or total) preorder representing preference between arguments. $(\alpha, \beta) \in \succeq$ means that argument α is at least as preferred as argument β (in other words, α is at least as strong as β).

[1] Some beliefs may not need to be supported by any kind of argument, which may normally occur, for instance, when modeling uncontested facts of common knowledge. In order not to introduce any superfluous concepts in the framework, such blindly accepted beliefs can be modeled as epistemic arguments with an empty set of premises.

The Dung-style AF associated with PAF is the pair $\langle A, Def \rangle$, where $Def \subseteq A \times A$ s.t. $(\alpha, \beta) \in Def$ (α defeats β) iff $(\alpha, \beta) \in R$ and $(\beta, \alpha) \notin \succ$, with $\succ$ being the strict relation associated to $\succeq$ (i.e. α defeats β iff α attacks β and β is not strictly preferred to α).[2]

- **CPAF:** A contextual preference-based argumentation framework is a tuple:

  $CPAF = \langle A, R, C, \rhd, \succeq_1, \ldots, \succeq_n \rangle$; where A and R are the same as for PAF, C is a finite set of contexts s.t. $|C| = n$, $\rhd$ is a strict total order on the contexts, and $\succeq_i$ is a (partial or total) preorder related to argument preferences issued from the context c_i. Therefore, each context has an associated argument preference preorder ($\succeq_i$), and contexts themselves are globally ordered by preference ($\rhd$). Contexts could be anything, including people.

  Using an aggregation procedure, it is possible to compute a single aggregated preference relation between arguments, denoted $Pref$ (the authors show multiple methods to achieve this), so that the $CPAF$ becomes equivalent to a PAF, with $Pref$ being equivalent to $\succeq$. This, in turn, allows the computation of a Dung-style framework AF equivalent to $CPAF$.

In my framework, I assume the existence of several agents specializing in different fields of knowledge, each of which may be implemented by an independent development team. As such, it is natural that different experts have different preferences over epistemic arguments, reflecting the design decisions undertaken by the team. Each expert agent can be thought of as a context w.r.t. CPAFs. I make no assumptions about the nature of epistemic argument preferences, but I can make some examples as to why they would be necessary, and why they might vary among different agents:

- **Credibility of external sources**: if experts need to discuss information provided by several third parties and they find contradictions between two sources, they are more likely to prefer information from the source which can provide more proof of credibility, such as accreditation by an independent organization, years of experience in the field, trust built from previous interactions, etc.

- **Maturity of standards**: if experts have to argue about alternative metrics for assessing the performance of a decision regarding some goal, more conservative experts are likely to prefer metrics which have long standing tradition over novel methodologies. One can draw an analogy with the debate over the use of sabermetrics in baseball management instead of traditional statistics.

- **Verifiability of information**: as pointed out in section 3.2, in the presence of contradicting arguments, agents are more likely to prefer those which draw information from direct observation or past experiences, such as simulation and CBR, over knowledge *a priori*.

A practical effect of allowing inconsistent preference orders for epistemic arguments, without further restrictions, would be that the set of acceptable practical arguments could slightly vary

[2] $(\alpha, \beta) \in \succ$ iff $(\alpha, \beta) \in \succeq$ and $(\beta, \alpha) \notin \succeq$, i.e. argument α is strictly preferred to argument β iff α is at least as preferred as β and β is not at least as preferred as α.

according to the perspective of each agent. Using an aggregation procedure like in CPAFs would not be possible, as it would require establishing a preference order between the different expert agents, which goes against the very purpose of a democratic deliberation dialogue. This situation may be unfavorable from the perspective of decision makers, because ultimately it could amount to having different experts believing in different consequences for the same decisions, thus basing their recommendations on different background information. Nevertheless, this is a situation that is common among humans: two people might even agree about what constitutes a desirable state of affairs, but they might still disagree about the state of affairs produced by a potential decision.

The only means to amend the issue of disagreement about epistemic preferences in a mutually satisfying way is to require epistemic knowledge to be universally accepted, i.e. the underlying facts leading to decisions (or rather, supporting practical arguments) should be proven without contestation. If inconsistent epistemic preferences become a problem (e.g. if it becomes too difficult to debate over any decisions), implementers may agree on a consistent preference order for all agents, in which case my epistemic framework would amount to a PAF. Nevertheless, disagreement about criteria for accepting objective information (epistemic) is a lot less common than disagreement regarding the importance of goals and values (practical), since the latter is more related to subjective factors. Therefore, in most cases, the requirement for universal acceptance of epistemic arguments should not be too restrictive.[3]

4.1.2 Definition of practical arguments

Practical arguments point out some advantage or disadvantage of carrying out a certain decision w.r.t. some specific goal, in accordance with the personal opinion of the expert who proposed that goal. They are grounded on a set of epistemic arguments justifying their conclusions. Practical arguments are merely opinative and therefore don't attack or support other arguments, regardless of the kind, while epistemic arguments may attack each other, therefore being able to indirectly attack practical arguments (by attacking an argument of their support set). Agents will regard practical arguments with different amounts of importance depending on the goals those arguments refer to. Moreover, practical arguments have no preference relation with epistemic arguments.

Some authors allow practical arguments to attack each other, which would normally happen when their associated decisions are mutually exclusive, or when two practical arguments concern the same decision but one draws attention to a positive outcome while the other draws attention to a negative outcome. The set of acceptable potential decisions is then computed from the surviving practical arguments. Such approaches are very pro/con focused, and thus quite distinct from typical decision theory which incorporates notions of uncertainty and degrees of satisfaction to compute a rank-ordering of potential decisions. As a result, in complicated scenarios, it becomes very impractical to incorporate complex decision-making criteria and it is harder for decision

[3] An alternative would be to allow each expert agent to make his own evaluation of the pros and cons of a decision based on the practical arguments he accepts, i.e. experts could build practical arguments out of epistemic arguments which are accepted by themselves but not necessarily unanimously accepted. This approach might have advantages in terms of simplicity of implementation and temporal complexity, but it seems inappropriate for modeling a rational debate.

makers to understand the advantages and disadvantages of different decisions. Over the next few sections, I present a new methodology that draws benefits both from qualitative argumentation-based reasoning and numerical decision theory criteria, facilitating both the work of implementers and the comprehension of decision makers.

The notion of advantage and disadvantage of carrying out an action is not as straightforward as it may seem, though many authors seem not to notice this. Many works in literature about argumentation frameworks simplify the effects of an action as a promotion or demotion of a goal, ignoring the need for different levels of satisfaction in order to effectively compare two decisions which promote or demote the same goal. Some, like Bourguet [BTMA13], only consider the positive effects of an action, with negative effects being modeled as reasons for preferring other actions, which makes the representation of the deliberation debate absurdly complex in the presence of more than a couple of candidate decisions.

In order to provide implementers with a simple, yet powerful methodology for modeling decision-making processes, it is necessary to distinguish between two important facets of what is considered a positive attribute, which I'll use later when suggesting a decision calculus that takes full advantage of the framework.

- **Absolute positivity (or negativity)**: the extent to which a decision fulfills (resp. neglects) a goal's satisfaction criteria. Given the aforementioned bipolar scale of real numbers (see definition of goals in section 3.2), a goal is said to be satisfied by a decision when its satisfaction degree after undertaking said decision is equal to or greater than 0 (in some contexts, implementers may interpret 0 as a point of neutrality, but this is indifferent with regard to the decision calculus presented in the next chapter). The satisfaction degree of a decision w.r.t. some goal is its absolute positivity. This relates to the concept of being "good" or being "bad". The fact that a decision is good at achieving some goal does not mean that it is better than the other decisions at achieving that goal.

- **Relative positivity (or negativity)**: the extent to which a decision promotes (resp. demotes) a goal better than (resp. worse than) some other decision. Any pair of decisions can be compared and one decision is said to be better than another (w.r.t. some goal) when its satisfaction degree is greater than that achieved by the other. This relates to the concept of being "better" or being "worse". The fact that a decision is better than all other decisions at achieving some goal does not mean that it is sufficiently good.

An important concept linked to the notion of relative positivity is the possibility of the deliberation dialogue ending with the conclusion that the best decision is to do absolutely nothing. In this framework, independently of the application context, the decision of taking no action must always be evaluated by the expert agents, so as to compare the benefits of keeping things as they are to the benefits of other decisions. Therefore, the set of candidate decisions is a non-empty set that is guaranteed to contain, at least, the decision of doing nothing, which I'll designate from now on as *DN* (Do nothing) for convenience.

Definition 2 (Do nothing): *Do nothing* (or *DN* for short) is a potential decision, which amounts to taking no action (formally, it is an empty set of actions), and whose performance is equal to that of the default state of the world.

The last important concepts related to practical arguments are those of uncertainty and risk. When the consequences of an action w.r.t. a goal are not known for sure, it may be necessary to define alternative degrees of satisfaction of that goal with varying degrees of probability. If those probabilities are known, we are making decisions under risk; otherwise we are making decisions under uncertainty (for convenience, I'll sometimes adopt the term uncertainty when referring to both concepts). Where these probabilities come from is up to the implementers to decide. They could come from CBR, cause-effect rules with associated probabilities, chaining of multiple rules with some uncertainty, etc. In AI, there are many approaches regarding ways of representing uncertain knowledge, such as the certainty-factor model [SB75] and the theory of fuzzy sets [Zad65]. Here, we are only concerned with managing uncertainty regarding the effects of a decision, which is more akin to the perspective of decision theory.

Definition 3 (Practical argument): A *practical argument* can be defined as a tuple:

$PA = \langle S, D, G, O \rangle$; where:

- S is a non-empty set of epistemic arguments supporting the information conveyed by PA. The concrete contents of epistemic arguments are not relevant to the argumentation framework, only their relationship with other arguments of both kinds. Typically, the support set should concern aspects like the relevant initial circumstances of the state of the world, facts and rules of the agent's base knowledge, the consequences of taking the decision referenced by PA and the impact of such consequences on the goal in question. Suggestions for representing the knowledge and beliefs that are typically needed in decision-making scenarios can be found in section 3.2;
- D is the decision in discussion by PA. It is a set of non-mutually exclusive elementary actions, possibly zero (DN), one or more than one. The concepts of mutual exclusivity and elementary action were discussed in section 3.2. Assume the existence of:
  - a set $ELAC$ of elementary actions;
  - a symmetric binary relation I representing incompatible elementary actions, i.e. $I \subseteq ELAC \times ELAC$.[4]
- G is the goal in discussion by PA. Remember that the purpose of a practical argument is to present the (positive or negative) effects of a decision on some specific goal;
- O is a non-empty set of potential alternative outcomes, each of which is a tuple of the form $\langle Satisfaction, Probability \rangle$. $Satisfaction$ is the degree of satisfaction of G after

[4] One cannot assume the transitivity of this relation. For instance, imagine a set $ELAC$ with three elementary actions: $a =$ *"Close street X to motorized traffic by means of traffic signs"*, $b =$ *"Open parking spaces for cars in street X"* and $c =$ *"Establish more punishing fines for owners of parked cars in street X"*. $(a,b) \in I$ and $(b,c) \in I$, but $(a,c) \notin I$.

taking decision D in the aforementioned bipolar scale of real numbers, where 0 stands for minimum fulfillment. Agents can use metrics to arrive at such scores. *Probability* is the likelihood that the given outcome will happen, i.e. a number between 0 and 1. When the probabilities of the outcomes are unknown (decision-making under uncertainty), each probability can be marked with a special value outside the admissible scale, such as -1 or *?*. When the probabilities are known (decision-making under risk), they must add up to 1. When the outcome of the decision is known for certain, O consists of a single element with *Probability* 1.

It isn't possible that two (accepted) practical arguments refer both to the same decision and goal. Otherwise, this would implicate that it would be possible two arrive at two different sets of outcomes for one goal regarding the same decision. Clearly, there would be a conflict between the supporting epistemic arguments, which should be settled.

If there is no (accepted) practical argument to assess the effects of some decision on a particular goal, it can be assumed that the decision doesn't affect the performance of that goal in any meaningful way w.r.t. the present state of affairs. Therefore, the set of outcomes of the decision according to said goal is the same as for the *DN*. If a set of outcomes for the *DN* is also not provided, a default value of $\langle Satisfaction : 0, Probability : 1\rangle$ should be used (this situation should be quite abnormal, since if an agent proposes a goal to be fulfilled then he should also have metrics to assess its satisfaction).

The concept of preference relation between practical arguments is quite different from that of epistemic arguments, and it is useless from the perspective of my argumentation framework because practical arguments don't attack each other. One can think of two alternative factors for eliciting preferences between practical arguments: the relative importance of the goals addressed by the arguments, or the relative positivity of the outcomes of the decision on the goal promoted by the arguments (in this case, one can only compare two arguments referring to the same goal, and this comparison depends greatly on the risk-aversion of the implementers if one has to deal with uncertainty). Moreover, there are other issues to take into account, such as the notion of commitment to satisfy a goal, which may invalidate a decision from the perspective of some agent. Therefore, while preferences between practical arguments could be useful from the perspective of rank-ordering decisions, such considerations are quite more complex than those allowed by a mere linear comparison of arguments. It is preferable to use practical arguments as building blocks for eliciting preferences between potential decisions, instead of establishing a preference order between the practical arguments themselves. The issue of rank-ordering decisions is thus left to a specific chapter; see 5.

4.2 The abstract argumentation framework

Now that I have defined what an argument is and what different kinds of arguments exist, it is necessary to establish a framework for representing the relationships between arguments, such as attack and preference, as well as preference between values and goals, so that acceptable arguments can be computed and decisions can be evaluated.

Let's start by looking at the formal definition of my abstract argumentation framework, which from now I'll call *argumentation framework for deliberation*, or AFD for convenience.

Definition 4 (Argumentation framework for deliberation): Assume a set of expert agents E, with $|E| = n$. An *argumentation framework for deliberation* is a tuple:

$AFD = \langle A_e, R, Epistprefs, A_p, Goals, Persgoals, Goalprefs \rangle$; where:

- A_e is a set of epistemic arguments (*EA*);
- R is a binary relation representing attacks between epistemic arguments, i.e. $R \subseteq A_e \times A_e$;
- *Epistprefs* is a set of (partial or total) preorders representing preference between epistemic arguments. $Epistprefs_i$ represents the epistemic argument preferences from the perspective of expert agent E_i;
- A_p is a set of practical arguments (*PA*);
- *Goals* is a non-empty set of goals, comprising the personal goals of all expert agents;
- *Persgoals* is a mapping that maps each element in the set E to a subset of *Goals*, i.e. maps each expert agent to the set of goals that were proposed by said agent, in other words, his personal goals (which he commits to uphold with a minimum level of satisfaction);
- *Goalprefs* is a set of total preorders on *Goals*. $Goalprefs_i$ is the preference order on the set *Goals* from the perspective of expert E_i. More accurately, it is an order of importance of goals, but I'll often use the term preference instead since it has widespread use in argumentation theory. Naturally, $|Epistprefs| = |Goalprefs| = |E| = n$. In order to compute *Goalprefs*, assume the existence of:
  - a non-empty set of values V;
  - a mapping *Valprefs* that maps each element in the set of experts E to a strict total order on V (experts cannot consider two values as equally important);
  - a function $Val : Goals \rightarrow V$ that maps each goal to its associated value;
  - a mapping *Persgoalprefs* that maps each expert E_i to a total preorder on his set of personal goals $Persgoals(E_i)$.

The framework contains all the necessary information for each expert to evaluate decisions. Let's look at how $Goalprefs_i$ might be computed. This will be relevant in chapter 5 in order to establish a preference order between candidate decisions based on accepted arguments.

First of all, an expert agent is always responsible for upholding the minimum satisfaction of all the goals he proposes. Such goals are encoded in $Persgoals(E_i)$. The value associated to each goal must be given by the proposer, and it can be retrieved using function *Val*[5]. Secondly, it is assumed that the agent defines internally a preference order for the goals he proposes. This is encoded in $Persgoalprefs(E_i)$. Now the expert needs to decide how important the goals of other experts are, in comparison to his own goals. Naturally, implementers are free to devise a procedure for identifying the relevance of goals proposed by other agents and incorporating those results in the agent's preference order. This would not be a trivial thing to do, since it requires some form of introspection about one's own preferences (meta-reasoning) and knowledge of the potentially complicated scientific reasons for upholding those goals. In the absence of these capabilities, one can achieve sensible results by taking advantage of the existence of values, and making two assumptions which seem reasonable from the perspective of human reasoning:

- if two goals are associated to different values, an agent will prioritize the goal that is associated to his most preferred personal value (note that agents are free to violate this assumption if both goals were proposed by themselves; nevertheless this is always a good principle to adhere to from a design perspective);
- if two goals are associated to the same value, an agent will prioritize the goal that he is responsible for upholding. In the case that both goals were proposed by other agents, he will not prioritize either goal (the two will be equally preferred).

Under these assumptions, we can define an algorithm for comparing any two pairs of goals from the perspective of any agent, which can be used to compute $Goalprefs_i$. A pseudo-code version is presented in algorithm 1.

Now the notion of defeat between arguments needs to be defined. The concept of defeat only makes sense from the perspective of an individual expert. Therefore, first I must define what an expert-specific AFD is.

Definition 5 (Expert-specific argumentation framework for deliberation): An AFD has

$|E| = n$ *expert-specific argumentation frameworks for deliberation*, or $EAFD$ for short. The $EAFD$ of expert E_i is $AFD_i = \langle A_e, R, Epistprefs_i, A_p, Goals, Persgoals(E_i), Goalprefs_i \rangle$; where:

- A_e, R, A_p and *Goals* are the same as for the AFD;
- $Epistprefs_i$ is the element of *Epistprefs* associated to expert E_i, i.e. a (partial or total) preorder representing preference between the elements of A_e ($Epistprefs_i \subseteq A_e \times A_e$). Furthermore, $\succ_i$[6] is the strict relation associated to $Epistprefs_i$;
- $Persgoals(E_i)$ is a subset of *Goals* containing solely the goals proposed by expert E_i;
- $Goalprefs_i$ is the element of *Goalprefs* associated to expert E_i, i.e. a total preorder representing preference between the elements of *Goals* ($Goalprefs_i \subseteq Goals \times Goals$).

[5]This implies that the set of social values V is standardized system-wide and must be known to all agents *a priori*. Given that V amounts to a small set of high-level qualitative concerns, this seems like a sensible requirement when designing an application.

[6]$(\alpha,\beta) \in \succ_i$ iff $(\alpha,\beta) \in Epistprefs_i$ and $(\beta,\alpha) \notin Epistprefs_i$

Algorithm 1: compareGoals. Algorithm to compare the relative importance of any two goals from the perspective of an expert agent, enabling the computation of $Goalprefs_i$

```
Data: Expert a, Goal g1, Goal g2
Result: Returns less than 0 if g1 is less important than g2; more than 0 if g1 is more
        important than g2; 0 if they are equally important, from the perspective of expert
        agent a
if g1 ∈ Persgoals(a) AND g2 ∈ Persgoals(a) then
    if (g1,g2) ∈ Persgoalprefs(a) AND (g2,g1) ∈ Persgoalprefs(a) then
        return 0
    else if (g1,g2) ∈ Persgoalprefs(a) then
        return 1
    else
        return −1
    end
else if Val(g1) == Val(g2) then
    if g1 ∈ Persgoals(a) then
        return 1
    else if g2 ∈ Persgoals(a) then
        return −1
    else
        return 0
    end
else
    if (Val(g1),Val(g2)) ∈ Valprefs(a) then
        return 1
    else
        return −1
    end
end
```

Now I can formalize the expert-specific defeat relation between arguments, so that $EAFDs$ may have an associated Dung-style argumentation framework.

Definition 6 (Expert-specific defeat relation between arguments): Consider AFD_i. Let $A = A_e \cup A_p$. $Def_i \subseteq A \times A$ is an *expert-specific defeat relation between arguments*, such that $(\alpha,\beta) \in Def_i$ iff one of the following conditions holds:

- $(\alpha,\beta) \in A_e$, and $(\alpha,\beta) \in R$ and $(\beta,\alpha) \notin \succ_i$, i.e. for a given expert agent, an epistemic argument α defeats an epistemic argument β iff α attacks β and β is not strictly preferred to α by that agent;
- $\alpha \in A_e$ and $\beta \in A_p$ (with $\beta = \langle S,D,G,O \rangle$), and $\exists \gamma \in S$ s.t. $(\alpha,\gamma) \in Def_i$, i.e. for a given expert agent, an epistemic argument α defeats a practical argument β iff from the perspective of that agent, α defeats at least one epistemic argument from $\beta's$ support set S. This is similar to the concept of undercutting [Pol87].

Therefore, AFD_i is equivalent to a Dung-style framework $AF = \langle A, Def_i \rangle$, with $A = A_e \cup A_p$ and Def_i being consistent with the previous definition. Let's now recall the notions of conflict-free set, acceptable argument and some acceptability semantics from Dung's framework, which also apply to *EAFDs* due to their equivalence to AFs.

Conflict-free set A set S of arguments is said to be *conflict-free* if there are no arguments α and β in S such that $(\alpha, \beta) \in Def_i$, i.e. if S has no internal defeats.

Acceptable argument An argument $\alpha \in A$ is *acceptable* w.r.t. a set S of arguments iff for each argument $\beta \in A$: if β defeats α then β is defeated by an argument in S.

Admissible extension A conflict-free set of arguments S is an *admissible extension* iff each argument in S is acceptable w.r.t. S, i.e. iff S defends all of its elements (by defeating all defeaters).

Complete extension A conflict-free set of arguments S is a *complete extension* iff it is admissible and contains all the arguments it defends.

Preferred extension A conflict-free set of arguments S is a *preferred extension* iff it is a maximal (w.r.t. set inclusion) admissible extension of $EAFD$, i.e. iff S defends all of its elements and is as large as possible. All preferred extensions are complete.

Under a given acceptability semantics, an argument can be in one of three states.

Definition 7 (Argument status): Let $A = A_e \cup A_p$ and *Extensions* be the extensions of an $EAFD$ under a given semantics. Let $\alpha \in A$:

- α is *skeptically accepted* iff $\forall ext \in Extensions$, $\alpha \in ext$;
- α is *credulously accepted* iff $\exists ext \in Extensions$ s.t. $\alpha \in ext$;
- α is *rejected* iff $\nexists ext \in Extensions$ s.t. $\alpha \in ext$.

Definition 8 (Objective acceptance): Let $A = A_e \cup A_p$ and $\alpha \in A$. α is *objectively accepted* under some semantics iff α is skeptically accepted by all expert agents under the same semantics (i.e. iff $\forall i \in E$, α is skeptically accepted in AFD_i under the same semantics). This resembles the definition provided by Bench-Capon [BC03].

As mentioned in section 4.1.1, we are only interested in arguments which are accepted by all agents, or, to be more specific:

- Only practical arguments which are objectively accepted under skeptically preferred semantics should be considered as factors for rank-ordering alternative decisions.

This agrees with the views of Prakken [Pra06], who claims that epistemic arguments should be skeptically accepted since they deal with establishing the truth, but the decision about what to do need not be consensual since it depends on personal preferences[7]. It also agrees with the view of Bench-Capon [BC03] that for an argument in a VAF to be persuasive, it should be in every preferred extension of all audiences. So it follows that practical arguments should be skeptically accepted by all expert agents under preferred semantics. Grounded or ideal [DMT07] semantics could also be used, as they have more computationally efficient solutions, but they are also seen as overly skeptical in some situations (with ideal semantics being less skeptical than grounded semantics). Therefore, skeptically preferred semantics should be favored by default. It is also considered the most satisfactory semantics by many authors [BG09].

4.3 Summary

Drawing on the foundational work of Dung and other authors, I devised an abstract argumentation framework aimed at decision-making and deliberation dialogues, allowing the incorporation of the key concepts introduced in section 3.2 and drawing connections between argumentation-based reasoning and classic decision theory criteria, an idea which I'll fully develop in the next chapter.

I started by establishing the separate roles of two different kinds of arguments: epistemic arguments, which concern beliefs; and practical arguments, which, when acceptable, can be used as factors for assessing the quality of decisions. We saw how epistemic arguments can attack and support other arguments, and have different strengths according to the perspective of each expert. Then I devised a structure for practical arguments built around the purpose of evaluating the fulfillment of goals, allowing the incorporation of uncertainty and risk. I formalized an abstract argumentation framework for deliberation, called AFD, which allows us to represent relationships between arguments and preference between goals, so that acceptable arguments can be computed and decisions can be evaluated on the basis of goal preference and objectively accepted practical arguments under skeptically preferred semantics. Then I provided the formal definition of expert-specific frameworks or $EAFDs$, and established the equivalence between those frameworks and Dung-style AFs, which allows us to formalize the concept of objective acceptance.

[7] Due to the nature of the argumentation frameworks considered by Prakken, where practical arguments may attack each other and have different preferences according to the objectives of an agent, he assumes that practical arguments need only be credulously accepted. Naturally I share the same opinion, but since $AFDs$ allow no attacks or preferences between practical arguments, here the nature of credulous practical reasoning stems not from the practical arguments themselves but from each agent's decision-ranking criteria. Therefore, in my framework, Prakken's idea of considering only skeptically accepted epistemic arguments amounts to considering only skeptically accepted practical arguments, because a practical argument supported only by skeptically accepted epistemic arguments is itself skeptically accepted, at least under any extension that is complete (supposing the opposite, one could find an extension that defended all epistemic arguments in the support set of some practical argument without defending that very practical argument, which is impossible since the defeaters of practical arguments can only be the defeaters of its associated epistemic arguments).

Chapter 5

Criteria for choosing decisions

Now that I have established a set of core concepts for modeling decision-making dialogues and the means to evaluate arguments built on those concepts, I need to define criteria for rank-ordering alternative decisions.

Preference for decisions is a matter of perspective. Each expert agent will be more likely to prefer decisions that favor his personal goals. Assuming the agent prefers his own goals to those of other agents, only when two decisions are indifferent w.r.t. the satisfaction of his personal objectives will the agent be sensitive to other expert's goals.

When taking a joint decision, there has to be some way for agents to conciliate their preferences. The final decision may not even be the most preferred from the perspective of any individual agent, nevertheless it should be the one that is most unanimously accepted by all. Therefore, the issue of ordering decisions has to be considered from two perspectives: individual and joint.

In this chapter, I'm going to establish some criteria for rank-ordering decisions from the perspective of individual experts and joint decision-making, taking into account priority between goals, respecting minimum satisfaction requirements and reasoning under uncertainty. As mentioned in the previous chapter, in order to assess and compare the quality of potential decisions, agents need to look at the objectively accepted practical arguments under skeptically preferred semantics. For the sake of simplicity, assume the existence of a set *Decisions* which includes all decisions that were considered in the debate, and a mapping *Factors* that maps each decision in *Decisions* to the set of objectively accepted practical arguments concerning that decision.

5.1 Establishing individual preference order

One can make three major assumptions regarding an expert agent's preference for decisions.

1. The agent will prefer decisions which comply with the minimum requirements of satisfaction of all his personal goals. Decisions disrespecting this condition can be said to be unsatisfying for the agent;

2. The agent will prefer decisions which fail to satisfy the least important of his personal goals, should he have to compare unsatisfying decisions;

3. The agent will prefer decisions which allow a greater amount of satisfaction of the goals he considers to be more important. The satisfaction degree of other agent's goals may be taken into account for tiebreaking alternatives.

Regarding the first assumption, one can immediately separate decisions into two groups: *satisfying* and *unsatisfying*:

- *Satisfying* decisions are those that abide to the minimum requirements of satisfaction of all personal goals defined by the expert agent (satisfaction of other agent's goals is not important at this stage);

- *Unsatisfying* decisions are those which don't comply with the minimum satisfaction of one or several of the aforementioned goals.

Thus the first and most important criterion for decision ranking is the number of minimally satisfied personal goals:

- From the perspective of an individual expert agent, any decision D_1 is preferred to decision D_2 if the number of personal goals minimally satisfied by D_1 is superior to that of D_2.

Suppose now that two decisions are satisfying, or that they are both unsatisfying but nevertheless fulfill the minimum satisfaction requirements of the same number of personal goals. According to the first criterion, both decisions are equally preferred and thus some form of tiebreaking is necessary. This is where the second and third assumptions come into place. Notice that, even after application of all three criteria, it is still possible that two decisions remain tied. In such cases, they are considered to be equally preferred to the agent.

As we'll see later in this section, the second criterion is similar to the third in terms of computation, except that it only applies to the personal goals which aren't minimally satisfied. Besides, it is only useful to tiebreak unsatisfying decisions: satisfying ones will always be tied according to the second criterion, because no goals fail. Therefore, I'll start by considering the third criterion.

The third criterion is not as straightforward as the first. To begin with, the expert needs to compare decisions goal by goal, considering the goals proposed by all agents. A decision is preferred to another w.r.t. some goal if the degree of satisfaction of that goal is superior to that achieved by the other decision (naturally, it is possible that two decisions perform equally well for the same goal). Afterwards, the agent looks at his most preferred goal. If decision D_1 accomplishes said goal better than D_2, D_1 will be preferred. If, however, both decisions equally satisfy the same goal, the agent will have to consider his second most preferred goal, and so on, until a tiebreak is achieved or all goals have been considered, in which case both decisions are equally preferred.

Now, one has to consider the case when two or more goals are equally important. The agent's total preorder on goals can be split into several sets of goals, each of which contains one or many equally preferred goals that are strictly preferred to the goals from the next set. For example, consider a preference relation on goals $Pref = \{G_1 \approx G_2 > G_3 > G_4 \approx G_5\}$. $Pref$ can be split into an array of sets of goals, with goals contained in sets in earlier indexes being more important than those in later indexes: *PrefSplit*= $[\{G_1, G_2\}, \{G_3\}, \{G_4, G_5\}]$. The criterion must now be refined: instead of performing a goal by goal comparison starting at the most important goal, the agent needs to compare decisions set by set starting at the most important set. Within a set, if the number of goals more highly satisfied by decision D_1 is greater than the number of goals more highly satisfied by decision D_2, D_1 is preferred to D_2. If the number of goals more highly satisfied is equal to both decisions, the next set needs to be considered.

The second criterion amounts to performing the same calculations as explained in the previous paragraph, but only considering personal goals which fail to be satisfied by either decision. For example, consider an agent with two personal goals, G_1 and G_2, with G_1 being the most important. If decision D_1 fails to meet G_1 and D_2 fails to meet G_2, the agent will prefer D_2 because it fails the least important goal. The fact that the second criterion is considered more important than the third is because it is generally assumed that avoiding damage (accomplishing minimum satisfaction) is more important than achieving benefits (maximizing satisfaction).

We can build an algorithm in pseudo-code for comparing any pair of decisions according to the three criteria proposed above, see algorithm 2 and its auxiliary 3 (clarity was favored over efficiency). The algorithm assumes the existence of a function *satisf(Decision d, Goal g)* which computes the degree of satisfaction of a goal achieved by some decision. This computation may not be trivial because one could have to deal with reasoning under uncertainty or risk. This issue is discussed in detail in the next section.

5.2 Computing goal satisfaction

This section is concerned with discussing the inner works of function *satisf(Decision d, Goal g)* under different situations.

When the consequences of a decision are fully predictable, computing goal satisfaction is trivial: this information is encoded in a practical argument as a single outcome with 100% probability. If the argument is accepted, the satisfaction degree associated to this outcome is synonymous with the goal's overall satisfaction w.r.t to the decision under analysis. In other words, *satisf* needs only find the accepted practical argument concerning the respective decision-goal pair, and return the satisfaction degree encoded in its sole outcome.

In the opposite situation, i.e. when the consequences of a decision on some goal are virtually inexistent or completely unknown (i.e. when there is no practical argument concerning that decision-goal pair or when there is such an argument but it is not objectively accepted), the rules defined in section 4.1.2 apply: the satisfaction of the goal is the same as for *DN*, or 0 (minimally satisfying) if no accepted practical argument concerning said goal is provided for the *DN*.

Difficulties arise when the consequences of a decision are not fully predictable, i.e. when reasoning under risk or uncertainty. This occurs when there is more than one potential outcome for the same goal regarding some decision. This information is encoded in a practical argument as a set of different satisfaction degrees with associated probabilities, which may be unknown in case of reasoning under uncertainty. Assuming the practical argument is accepted, one has to find a way to compute the goal's overall satisfaction considering the various potential outcomes.

Decision theory has proposed several rules for reasoning under uncertainty or risk. What follows is a description of some of the most popular rules and how they can be used in combination with my theory.

Starting with reasoning under uncertainty or ignorance, let's take a look at four decision rules: Wald's maximin rule [Wal49], maximax [Mod49], Hurwicz's optimism-pessimism rule [Hur51] and Savage's minimax regret rule [Sav51].

According to the maximin rule, when choosing between two decisions, an agent should favor the one whose worst possible outcome is most satisfying. Therefore, in my framework, this would amount to having function *satisf* return the lowest $Satisfaction$ among all potential $outcomes \in O$ encoded in the practical argument concerning the decision-goal pair under analysis (remember $PA = \langle S, D, G, O \rangle$, with O being a non-empty set of outcomes, each of which is a tuple of the form $\langle Satisfaction, Probability \rangle$). The maximin rule represents purely pessimistic thinking.

The maximax rule, representing purely optimistic thinking, amounts to favoring the decision whose best possible outcome is most satisfying. Therefore, *satisf* would now return the highest $Satisfaction$ among all potential outcomes encoded in the *PA* concerning the decision-goal pair under analysis.

Hurwicz's pessimism-optimism index is a value α between 0 and 1 representing how pessimistic the agent is: a value of 0 represents pure optimism, amounting to the maximax rule, while a value of 1 represents pure pessimism, amounting to the maximin rule. For each decision, one has to determine the best (highest satisfaction) and worst (lowest satisfaction) possible outcome, respectively denoted *max* and *min*. The overall satisfaction degree of a decision can be computed by the formula $\alpha \times min \; - \; (1-\alpha) \times max$. In my framework, *satisf* should determine the lowest and highest $Satisfaction$ among all potential outcomes encoded in the *PA* concerning the decision-goal pair under analysis and return the overall satisfaction value using the aforementioned formula.

The minimax regret rule is a moderately pessimistic approach. When choosing between two decisions, an agent should favor the one whose maximum possible regret among all potential outcomes is lowest. Regret, w.r.t. an outcome, is the difference between the satisfaction degree of the decision with highest satisfaction (among all decisions) assuming the occurrence of that outcome, and the satisfaction degree achieved by the decision under analysis (assuming the same outcome). This is analogous to the thinking of a gambler when reflecting what bet he should have made if he had known the results of the game beforehand. This approach is great when the set of potential decisions (and potential states of nature) is very well defined, as it focuses more on the relative strengths of the decisions than on their individual merits or shortcomings. Nevertheless, it can be problematic from a computational perspective. First of all, when computing the overall satisfaction of a goal w.r.t. a specific decision, it becomes necessary to consider the performance of all other decisions. Secondly, in rare cases, it may be that different decisions are dependent on different states of nature even when considering the same goal. For instance, decisions with a single (certain) outcome may be interleaved with decisions with two or more (uncertain) potential outcomes regarding the same goal. Considering this, it's not possible to generally apply the minimax regret rule in my framework, unless implementers enforce the restriction that practical arguments concerning the same goal must deal with an equal number of outcomes (which seems reasonable enough, given that it's always the proponent of a goal who provides its assessments). Also, notice that algorithm 2 expects that a decision will outperform another w.r.t. some goal if the return of function *satisf* is higher for that decision, since it was designed to deal with notion of utility, not regret. Therefore, in order to use the minimax regret rule correctly, in this case *satisf* must return a pair $\langle Satisfaction, MaxRegret \rangle$ where the value of $Satisfaction$ is the one associated to the outcome with maximum regret. $Satisfaction$ is then used in lines 9, 12 and 15 of algorithm 2, while the value of $MaxRegret$ is used in lines 10 and 12 of algorithm 3 (switching the "greater than" operator for the "lesser than" operator and vice-versa).

For decision-making under risk, the most widely used approach is the maximum expected utility [MVN53]. It requires knowledge of the probabilities of the different states of nature and of the utilities of the decisions' outcomes associated to each of those states. The expected utility of a decision is the sum of the utilities of each of its potential outcomes weighted by their respective probabilities. The concept of using a set of outcomes encoded as $\langle Satisfaction, Probability \rangle$ pairs in practical arguments is fully consistent with this theory. Thus, when evaluating decisions under risk using this rule, *satisf* should return the sum of the product $Satisfaction \times Probability$ of all $outcomes \in O$ encoded in the practical argument concerning the decision-goal pair under analysis (i.e. $\sum_{i=1}^{|O|} Satisfaction_i \times Probability_i$ for $PA = \langle S, D, G, O \rangle$ with $O_i = \langle Satisfaction_i, Probability_i \rangle$).

Finally, note that an agent should always use the same decision rule when comparing any two pairs of decisions; otherwise his reasoning would be considered flawed or biased. Whether different agents are allowed to use different decision rules is up to the initiator agent to define. For instance, by implementation one expert agent might be more optimistic than another, but if the initiator agent defines precisely which decision rule to use, the deliberation process won't be subject to variations in the agents' "character". This implies that the expert agents have to know *a priori* a basic set of different decision rules from which the initiator agent can select from, such as the ones proposed above (unless agents are capable of exchanging decision rules, but this goes beyond the scope of this section).

5.3 Establishing joint preference order

To compute a joint preference order over potential decisions, one may use an aggregation procedure that takes into account the same major assumption as for the individual preference orders, and a second assumption for breaking ties.

1. First of all, regardless of the preference orders established by each expert agent, one should favor decisions which violate as few (or, in other words, fulfill as many) minimum requirements of satisfaction as possible. In this case, it is necessary to consider all goals, i.e. the entire set of goals $Goals$ from the AFD.

2. In case of a tie (e.g. if two decisions are considered satisfying by all agents), one should favor the decision that ranks highest according to the junction of all individual preference orders. This implicates that it is necessary to somehow devise a way to join the preference orders of all expert agents. Social choice theory and, in a broader sense, voting theory in general attempt to deal with this problem, with ranked voting systems being particularly appropriate to deal with joining multiple preferences orders.

Once again, regarding the first assumption, one can immediately separate decisions into two groups from a joint perspective: *globally satisfying* and *globally unsatisfying*.

- *Globally satisfying* decisions are those that abide to the minimum requirements of satisfaction of all goals in *Goals*, i.e. the goals defined by all expert agents;

- *Globally unsatisfying* decisions are those which don't comply with the minimum satisfaction of at least one goal in $Goals$.

Thus the first and most important criterion for joint decision ranking is similar to that used in individual preference ordering:

- From a joint perspective, any decision D_1 is preferred to decision D_2 if the number of goals from set *Goals* minimally satisfied by D_1 is superior to that of D_2.

Now it is necessary to formalize the second assumption into a proper criterion for tiebreaking decisions that are equally preferred according to the first criterion. This could happen, for instance, when both decisions are globally satisfying, or when both are globally unsatisfying but nevertheless fulfill the minimum satisfaction requirements of the same number of goals. Note that, just as it was for individual preference orders, two decisions are considered equally preferred if they remain tied after application of both criteria.

I could tackle the issue of joining the individual decision preference orders in two ways: assigning varying degrees of importance to different agents, thus favoring the opinion of some agents over others (as in the CPAF framework); or assuming that the opinion of all agents is equally valid or important. I have been making the latter assumption during the entirety of this work, which is much more consistent with the values of democracy and fair participation that guide most decision-making, political and public participation processes among humans. Therefore, I'll keep assuming all expert agents are equally important. By no means does this assumption disempower the users of the system: if the decision maker is particularly interested in the opinion of a certain expert agent, the DSS can provide him with the agent's individual decision preference order. On the other hand, if the user considers that the agent's personal preference is irrelevant when computing joint preference order, the DSS should permit to ignore it, while keeping his epistemic knowledge valid.

We can resort to a variety of ranked voting systems to compute joint preference order according to the second criterion. Many different systems have been proposed in literature, such as the Borda count [dB81], Ranked Pairs [Tid87] and the Schulze method [Sch03]. Different systems satisfy different properties: it is up to implementers to define which system the initiator agent should use, because no voting system is considered to be fully satisfying, see for instance Arrow's impossibility theorem [Arr63]. Expert agents need not know what system is used by the initiator: they need only provide their individual preference orders.

As an example, I'll use the Borda count system, modified so as to allow for tied preferences, i.e. where individual agents may express the same amount of preference for certain decisions. The primary reason for demonstrating this system is that it is very simple to comprehend; in practice, nothing prevents implementers from using any ranked voting system, as long as it allows for tied preferences.

If we're given the preference order of an individual agent, we can compute the numerical rank of the decisions, from his perspective, by representing the preference order as an array, sorted by ascending order. Equally preferred decisions should rank as low as the array index of the lowest equivalently preferred decision (i.e. their rank should be the number of strictly less preferred decisions), while decisions with no ties should rank the same as their respective indexes in the array. For example, assume the preference order of an expert agent e is:

- $DecisionPrefs_e = \{D_1 > D_2 \approx D_3 \approx D_4 > D_5\}$

We can represent $DecisionPrefs_e$ as an array sorted by ascending order:

- $DecisionPrefsArray_e = [D_5, D_2, D_3, D_4, D_1]$

The decision ranks of agent e are:

- $rank_e(D_1) = 4, rank_e(D_2) = 1, rank_e(D_3) = 1, rank_e(D_4) = 1, rank_e(D_5) = 0$

One simply needs to compute the ranks of all decisions from the perspective of each agent, and then sum up those ranks. For instance, consider the ranks of the aforementioned agent e, plus those of an agent x.

- $DecisionPrefs_x = \{D_2 > D_3 > D_1 \approx D_5 > D_4\}$
- $DecisionPrefsArray_x = [D_4, D_1, D_5, D_3, D_2]$
- $rank_x(D_1) = 1, rank_x(D_2) = 4, rank_x(D_3) = 3, rank_x(D_4) = 0, rank_x(D_5) = 1$

Assuming only the existence of expert agents e and x, the joint decision ranks are:

- $rank(D_1) = 5, rank(D_2) = 5, rank(D_3) = 4, rank(D_4) = 1, rank(D_5) = 1$

Finally, assuming a tie in the first criterion, e.g. all five decisions are globally satisfying, the joint preference order is:

- $JointDecisionPrefs = \{D_1 \approx D_2 > D_3 > D_4 \approx D_5\}$

We can build an algorithm for comparing any pair of decisions according to the two criteria proposed above, which can be used to establish joint preference order (once again, clarity and simplicity were favored over efficiency). A pseudo-code version is provided in algorithm 4.

5.4 Analysis of joint preference computation

The issue of finding the candidate decision that best satisfies a group of people with widely different concerns and preferences is obviously quite complex and no theory provides a universally satisfying solution. This is especially true in the case of voting systems: most elections still use some variant of plurality voting, even with all its well-known disadvantages, not only because it is a relatively simple system, but also because no alternative is completely free of undesirable properties. Voting systems may be the only way to arrive at a reasonable compromise in scenarios with many participants, but here this is not the case: we have a few highly specialized artificial counselors and, unless otherwise indicated, the user wishes to follow the recommendations of all. Sometimes, the most satisfying decisions for a majority of experts can be considered catastrophic for another – in such cases, a voting system might result in the opinion of this single agent being "buried" under the weight of all the others, which is not acceptable in our case, as it might result in intolerable damages. After all, the user must have requested the expert's opinion for some reason.

All things considered, it would seem inappropriate to employ exclusively a ranked voting system, based on individual preference orders, to recommend decisions to the users. This is why I chose to use a criterion that precedes voting: first of all, prefer decisions that minimally satisfy all agents to the greatest possible extent[1]. A globally satisfying decision may not even be the most preferred from the perspective of any agent, but at least all goals are satisfied to a minimum

[1] In case the reader is wondering if the two criteria are equivalent, it is easy to come up with a counter-example. Consider, for instance, a scenario with two expert agents, e and x, each of which with five personal goals, and three potential decisions to choose from, A, B and C. Now consider that:

- A satisfies all 5 of $e's$ goals, and 2 of $x's$ goals;
- B satisfies 1 of $e's$ goals, and 4 of $x's$ goals;
- C satisfies none of $e's$ goals, and 3 of $x's$ goals.

The individual preference orders are therefore:

- $DecisionPrefs_e = \{A > B > C\}$
- $DecisionPrefs_x = \{B > C > A\}$

According to the first criterion, A is preferred to B because A satisfies 7 goals, while B satisfies 5 goals. According to the second criterion (using the previously described variation of Borda count), B is preferred to A because B ranks 3 points, while A ranks 2 points.

degree: no restrictions, legislations or other impositions are violated, and thus harm is avoided while potentially improving the state of the world to some extent. Only when decisions are tied according to the first criterion do we need to resort to voting: it would be impossible to employ a second aggregation criterion similar to the second and third criteria of the individual preference order computation, as it would be necessary to define a global preference order among the experts' personal goals, which goes against my underlying assumption of equal participation.

Now let's look at some of the desirable properties that we can observe in the joint preference computation system. First of all, it can be demonstrated that any joint preference order is total or complete, i.e. any pair of decisions can be compared.

- Since it is possible to compute the number of satisfied goals by any decision, the first criterion satisfies totality, as it is equivalent to a "is greater than or equal to" binary relation on the set of integer numbers, which is total (each decision is associated to an integer representing how many goals it satisfies);

- The second criterion satisfies totality as long as the underlying ranked voting system satisfies totality, as is the case of any well designed ranked voting system (any two pair of candidates can be compared). Note that this assumes that the individual preference orders which serve as input to the voting system are total, which is always achievable by definition of algorithm 2. The variation of the Borda count algorithm demonstrated in the previous section satisfies totality, as it is clearly possible to calculate the sum of the number of "points" of any decision by the aforementioned method, and because those numbers are comparable, the respective decisions are thus comparable as well.

Similarly, one can also demonstrate transitivity, i.e. if a decision *A* is preferred to decision *B* and *B* is preferred to decision *C*, then *A* is preferred to *C*.

- The first criterion, being equivalent to a "is greater than or equal to" binary relation on the set of integer numbers, obviously satisfies transitivity: if the number of goals satisfied by decision *A* is at least as great as for decision *B* and that of decision *B* is at least as great as for decision *C*, then the number of goals satisfied by *A* is at least as great as for *C*;

- The second criterion satisfies transitivity as long as the underlying ranked voting system satisfies transitivity. Many systems satisfy this property, Schulze in particular demonstrates it for his method [Sch03]. It is easy to see intuitively that my variation of the Borda count satisfies this property: if the total number of points attained by decision *A* is at least as great as for decision *B* and that of decision *B* is at least as great as for decision *C*, then the total number of points attained by *A* is at least as great as for *C*.

Pareto efficiency can also be demonstrated. A decision system is Pareto efficient if it is impossible to find any decision that makes any agent better off without making at least one agent worse off, comparatively to the winning decision(s) produced by the system.

- Consider, as a first case, decisions ranked according to the first criterion. I need to demonstrate that, for any decision A satisfying n goals, it is impossible to find a decision B satisfying less than n goals while making some agent better off and not making any agent worse off. Suppose the opposite. Remember that according to the criteria defined in section 5.1, if B makes no agent worse off, then the number of personal goals satisfied by B must be greater than or equal to A for all agents. If this is true, then the total number of goals satisfied by B must be at least as great as A, i.e. B satisfies at least n goals. Contradiction;

- The second criterion satisfies Pareto efficiency as long as the underlying ranked voting system satisfies Pareto efficiency. Many systems satisfy this property; once again Schulze is an example. Pareto efficiency can be reformulated for voting systems like this: when no agent strictly prefers decision B to decision A and at least one agent strictly prefers A to B, then decision B must not be jointly preferred to A. Once again, it can be demonstrated that my variation of the Borda count satisfies this property: if no agent strictly prefers B to A, then B attains at most as many points as A. Furthermore, if one agent strictly prefers A to B, then, by this method, A will score at least one more point than B. Therefore, if A has, in total, at least one more point, B can't possibly be jointly preferred to A.

5.5 Summary

This chapter provided a series of criteria for rank-ordering alternative decisions, connecting argumentation theory with classic decision theory criteria and some concepts from social choice theory, thus showing how accepted arguments can be used together with notions of personal values and goals to establish which decisions are most satisfying, both from individual and joint perspectives.

I started by compiling criteria for ranking decisions according to the preferences of individual expert agents. I made three major assumptions: firstly, that agents prefer decisions which minimally satisfy all of their personal goals; secondly, that in the event some personal goals can't be satisfied, agents prefer decisions which violate their least important personal goals; and lastly, that agents prefer decisions which maximally satisfy their most important personal goals. Afterwards, we saw how agents can employ classic theory for decision-making under risk and uncertainty for the purpose of computing the satisfaction of goals according to objectively accepted practical arguments under skeptically preferred semantics. I finished by providing criteria for aggregating preferences, based on two major assumptions: that decisions which minimally satisfy the personal goals of all agents are jointly preferred; and that in the case of a tie, agents should be given the ability to vote in accordance to their individual rank-ordering of decisions. I provided algorithms for these computations and an analysis of some desirable properties of my criteria.

Algorithm 2: compareDecisions. Algorithm to compare the performance of any two decisions from the perspective of an expert agent, enabling the computation of the expert's individual rank-ordering on decisions

```
Data: Expert a, Decision d1, Decision d2
Result: Returns less than 0 if d1 is less preferred than d2; more than 0 if d1 is more
        preferred than d2; 0 if they are equally preferred, from the perspective of expert
        agent a
hash_map⟨Goal,double⟩ d1satisf, d2satisf /* Map each goal to a
satisfaction value for both decisions (using usual bipolar scale
of real numbers) */
int d1minSatisf = 0, d2minSatisf = 0 /* Keep track of number of minimally
satisfied goals for both decisions */
Goal[ ] unsatisfGoals /* Keeps track of unsatisfied personal goals by
either decision */
Goal[ ] goalPrefs_a = Goals.sortByAgentPrefsDesc(a) /* Compute the Goalprefs_a
array-style equivalent, see algorithm 1 */
foreach g in goalPrefs_a do
    d1satisf[g] = satisf(d1, g)
    d2satisf[g] = satisf(d2, g)
    if g ∈ Persgoals(a) then
        if d1satisf[g] ≥ 0 then
            ++d1minSatisf
        end
        if d2satisf[g] ≥ 0 then
            ++d2minSatisf
        end
        if d1satisf[g] < 0 OR d2satisf[g] < 0 then
            unsatisfGoals.append(g);
        end
    end
end
if d1minSatisf != d2minSatisf then // First criterion
    return d1minSatisf − d2minSatisf
else
    if unsatisfGoals.length > 0 then /* If decisions are unsatisfying,
    consider second criterion (tiebreakDecisions is algo. 3) */
        int secondCritResult = tiebreakDecisions(a, unsatisfGoals, d1satisf, d2satisf)
        if secondCritResult != 0 then
            return secondCritResult
        end
    end
    return tiebreakDecisions(a, goalPrefs_a, d1satisf, d2satisf) // Third criterion
end
```

Algorithm 3: tiebreakDecisions. This is an auxiliary algorithm to enable comparison of decisions according to second or third criterion

```
Data: Expert a, Goal[ ] sortedGoals, hash_map<Goal, double> d1satisf,
      hash_map<Goal, double> d2satisf
Result: Same as for algorithm 2, but considering only the second or third criterion
  /* Note that the computations are the same both for the second
  and third criteria: the only difference lies in the goals to be
  considered, which in the case of the second criterion are only
  the personal goals unsatisfied by either decision, and in the
  case of the third criterion are the goals of all agents. These
  goals, provided in the call from algorithm 2, are encoded in
  sortedGoals, sorted by descending preference order */
int d1BetterSatisfiedGoals = 0, d2BetterSatisfiedGoals = 0 /* Number of equally
  important goals more highly satisfied by each decision (in the
  set under consideration) */
for int i = 0; i < sortedGoals.length; ++i do
    if i > 0 AND compareGoals(a, sortedGoals[i-1], sortedGoals[i]) > 0 then /* If
      current goal is less important than previous goal, we are
      entering a new set of equally important goals */
        if d1BetterSatisfiedGoals != d2BetterSatisfiedGoals then
            return d1BetterSatisfiedGoals - d2BetterSatisfiedGoals
        else
            d1BetterSatisfiedGoals = d2BetterSatisfiedGoals = 0
        end
    end
    if d1satisf[sortedGoals[i]] > d2satisf[sortedGoals[i]] then
        ++d1BetterSatisfiedGoals
    else if d1satisf[sortedGoals[i]] < d2satisf[sortedGoals[i]] then
        ++d2BetterSatisfiedGoals
    end
end
return d1BetterSatisfiedGoals - d2BetterSatisfiedGoals
```

Algorithm 4: compareDecisionsJointly. Algorithm to compare the performance of any two decisions from an aggregated perspective, enabling the computation of the joint rank-ordering on decisions

```
Data: Decision d1, Decision d2
Result: Returns less than 0 if d1 is less preferred than d2; more than 0 if d1 is more
        preferred than d2; 0 if they are equally preferred, from the joint perspective
int d1minSatisf = 0, d2minSatisf = 0 /* Keep track of number of minimally
satisfied goals for both decisions */
foreach g in Goals do
    int d1GoalSatisf = satisf(d1, g), d2GoalSatisf = satisf(d2, g)
    if d1GoalSatisf ≥ 0 then
        ++d1minSatisf
    end
    if d2GoalSatisf ≥ 0 then
        ++d2minSatisf
    end
end
if d1minSatisf != d2minSatisf then // First criterion
    return d1minSatisf − d2minSatisf
else // Second criterion, using variation of Borda count
    int d1GlobalRank = 0, d2GlobalRank = 0
    foreach e in E do // Iterate over all expert agents
        Decision[] decisionPrefs_e = Decisions.sortByIndividPrefsAsc(e) /* Compute
        individual decision preference order of agent e, see
        algorithm 2 */
        for int i = 0, rank = 0; i < decisionPrefs_e.length; ++i do
            if i > 0 AND compareDecisions(e, decisionPrefs_e[i − 1], decisionPrefs_e[i]) < 0
            then
                rank = i  /* Update rank to current index if current
                decision is strictly more preferred than previous one
                for agent e */
            end
            if d1 == decisionPrefs_e[i] then
                d1GlobalRank += rank
            else if d2 == decisionPrefs_e[i] then
                d2GlobalRank += rank
            end
        end
    end
    return d1GlobalRank − d2GlobalRank
end
```

Criteria for choosing decisions

Chapter 6

Communication protocol

Over the last chapters, I have defined the key concepts of an argumentation model for deliberation, an abstract argumentation framework to compute acceptable arguments, and criteria for rank-ordering decisions from individual and joint perspectives. Now we need a protocol to allow agents to communicate information, exchange arguments, express preferences and, all in all, take advantage of the features designed over the previous chapters.

In this chapter, I will define a communication protocol loosely inspired by the DDF protocol [MHP07], with several adaptations to account for the different key concepts underlying this work, as well as to take advantage of argumentation and simulation, with the development of DSS in mind.

First, I'll start by making some general considerations about the architecture of the protocol, the types of actors involved and any underlying assumptions deemed relevant.

6.1 Outline of the architecture

Two kinds of agents participate in the dialogue. There is always one (and only one) initiator agent, and at least one expert agent (potentially many). The initiator agent acts on behalf of the human user. A DSS may consist of a GUI where the decision-maker defines the inputs of the problem to solve and any constraints on its resolution. The initiator acts as an interface with the experts agents: he takes the information provided by the user, communicates it to the experts and gathers their advice about potential decisions. This advice is then presented to the user via the GUI.

The deliberation dialogue begins with the initiator requesting assistance from the expert agents, who may accept or decline to help. Once the initiator agent is satisfied with the assembly of experts that he has gathered, he will proceed to define the initial circumstances (i.e. properties) of the problem at hand. He can do this in two ways, potentially mixed: by enumerating the relevant properties of the state of the world, or by providing a series of simulation files constituting a model representative of the state of the world, allowing the expert agents to extract relevant properties by themselves. The initiator may additionally set the decision criteria for computing goal satisfaction when reasoning under uncertainty.

Once the inputs are defined, the initiator agent needs to point at some specific issue in the world that requires attention, i.e. he must formulate the governing question that will constitute the problem subject to debate by the expert agents. Two approaches are possible: either the governing question is defined by the decision-maker, and thus the initiator merely informs the expert agents of the question to solve; or the initiator agent instructs experts to try to autonomously discover issues in the world according to their knowledge and report them back to the initiator, who in turn will present those issues (if any) to the user, allowing him to select an issue to pose as a governing question.

After the governing question is clearly defined, experts will commence an exploratory phase, where they propose elementary actions (and establish which actions are mutually exclusive) and personal goals to fulfill. Next, experts take turns exchanging practical arguments to assess the performance of potential decisions, as well as epistemic arguments to support or refute those assessments (the initiator may also contribute with trustworthy epistemic knowledge).

Once the discussion is over, the initiator agent gathers the decision preference order of each expert agent, computes joint preference order, and presents the information to the user, who may choose to redefine the inputs (thus restarting the debate) or to finish the dialogue.

At any moment during the dialogue, agents may see fit to retract previous locutions. The initiator agent, in particular, is capable of, and responsible for, retracting locutions uttered by other agents, provided he finds them to make little sense, to be semantically unsound, or to be in clear violation of the rules.

6.2 Stages

The dialogue is divided in seven stages, plus an optional stage. Each stage corresponds to some typical moment that you would expect in an ideal deliberation dialogue conducted under my protocol. As such, the types of locutions uttered by the agents also vary from stage to stage. This allows us to define a semi-formal structure of dialogue, so that implementers know what the agents should do at any given time to carry out their tasks and advance the discussion efficiently. In practice, however, unexpected situations might occur, so dialogues might have to switch back and forth between stages. As we'll see, there are some basic rules governing these transitions. Let's now look at the purpose of each stage.

1. Reunion

The dialogue begins with the initiator agent requesting assistance from expert agents. The first stage is all about gathering an assembly of experts before the actual discussion begins. Experts make their services known on the network so that other agents may contact them in search for advice. In a typical MAS setup, experts would probably publish their services using some variant of what is commonly called a "Yellow Pages" service, such as that implemented in the JADE platform [Tel].

The initiator agent consults the "Yellow Pages", finds expert agents concerned with different perspectives (such as Law, Environment, Economics, etc.), and makes contact with them. Experts are, of course, free to grant or deny assistance. In the latter case, they might provide a reason for acting so, such as being near operational limits.

Transition rules: once the initiator gathers experts specializing in all relevant concerns, the dialogue may move on to stage *Input definition*. Nevertheless, if an expert agent abandons the dialogue prior to its termination, the initiator might have to temporarily interrupt the debate and look for a replacement, no matter what stage the dialogue is in. The replacing expert agent should be informed by the initiator of all the relevant utterances exchanged in the dialogue up to that point.

2. Input definition

In this stage, the initiator defines the initial circumstances or properties of the state of the world that may be relevant to the discussion. The nature and structure of these properties were properly analyzed in section 3.2.

The initiator may define properties in two ways: by formally listing all properties explicitly by extension, or by providing a series of simulation files. These simulation files should enable expert agents to assemble an accurate model of the state of the world, so that they may have access to richer information and extract any relevant properties by themselves (after running the simulation locally). The initiator could provide access to those files as a series of FTP URLs, for instance. Additionally, he may combine the two modalities in order to supply additional properties not representable in the model.

At this stage, the initiator should also communicate any obligatory criteria for computing goal satisfaction when reasoning under risk or uncertainty/ignorance.

Transition rules: once all initial circumstances and obligatory decision criteria have been defined, the dialogue may move on to the stage *Issue discovery* or straight to stage *Question formulation*. Because the initial input greatly affects the outcomes of the debate, if the initiator agent wishes to communicate any retractions or additions to it at a later point, the dialogue should restart at the *Discussion* stage, in case that it has already moved past that stage. This may typically happen when the dialogue reaches the *Finalization* stage and the user wishes to know how the experts would solve the problem if different assumptions were made about the state of the world.

3. Issue discovery (optional)

After the *Input definition* stage, the initiator agent needs only define precisely what problem in the state of the world requires attention, so that expert agents may begin their discussion. If the governing question is determined by the user *a priori*, the dialogue may switch straight from stage *Input definition* to *Question formulation*. But rather than just solving problems, the user might be interested in using the DSS from the perspective of optimizing the performance of some real life infrastructure, as might happen in transportation planning. In this case, the user might take advantage of the expert agents' knowledge in order to find issues (or unexploited opportunities) in the state of the world, some of which he might not be aware of, that compromise some quality attributes.

Experts discover issues by extracting information from the initial circumstances, described in the previous stage, and comparing some attributes of the state of the world to their expectations of quality or goals, using metrics. Each expert should look for issues according to his area of expertise, for instance an environmentalist should look for environmental issues. In this scenario, it is particularly interesting to convey the input using simulation, since experts may extract a greater amount of information from the simulation models than from a mere list of properties.

Once the initiator gives the order to look for issues, the expert agents will analyze the input and take turns reporting issues to the initiator (the mechanism of turns is better explained in stage *Discussion*). Each issue should be accompanied by a rough measure of its severity. The initiator will then report those issues (if any) to the user, allowing him to select one to pose as the governing question.

Transition rules: once all experts have taken up one turn to report issues to the initiator agent, the dialogue may move on to stage *Question formulation*. Under normal circumstances, the dialogue shouldn't return directly to this stage, as there is no reason to rediscover issues once the governing question has been posed (unless the properties of the problem are changed). Nevertheless, the initiator may forcibly revert to this stage by retracting all utterances exchanged afterwards, as is the case with any stage.

4. Question formulation

This stage consists of a single utterance: the governing question, posed by the initiator agent. As explained in section 3.2, it points at some issue in the state of the world that requires attention. It is the problem that the user wants to solve, and as such the discussion by the expert agents should be centered on trying to solve this question. Governing questions typically should follow some decision-oriented formulation like *"What should be done about X?"* or *"What should be done to achieve Y?"*

As explained before, the dialogue can transit to this stage both from *Input definition* and *Issue discovery*.

Transition rules: once the governing question is posed, the dialogue may move on to stage *Exploration*. If the governing question has to be reformulated subsequently, the dialogue should restart at the beginning of stage *Exploration* (by retracting all utterances after the *Question formulation* stage). The only sensible reason for reformulating the governing question (other than having made a mistake in its original formulation) would seem to be when the user wishes to resolve more than one problem in the same session.

5. Exploration

In this stage, each expert agent takes up one turn to define his personal goals to fulfill (and its associated values), as well as proposing potential elementary actions that he believes might be of use to solve the question in discussion, and defining among the proposed actions which ones are mutually exclusive. This stage begins immediately after the initiator utters the governing question. It functions as a preamble to stage *Discussion*, being somewhat more loosely defined, with agents being able to go back and forth between the two.

In some scenarios, the user may wish to permit additional agents to join the dialogue autonomously (i.e. without prior invitation) if they desire to contribute to the discussion. This is only possible starting at this stage because the subject matter of the dialogue isn't defined until the end of stage *Question formulation*. If this is the case, the initiator agent may announce the running dialogue using the "Yellow Pages" service, and keep the announcement until the end of the *Discussion* stage.

Transition rules: once all experts have taken up one turn, the dialogue may move on to stage *Discussion*. Experts may find it necessary to add or retract goals or elementary actions during the *Discussion* stage. They may freely do so, provided that any arguments concerning eventually retracted goals or actions (or decisions incorporating newly discovered incompatible actions) are retracted as well, either voluntarily by their utterers or by the initiator agent.

6. Discussion

This is where the greatest part of the dialogue normally takes place. Expert agents take turns communicating. They propose decisions to be analyzed (decisions being sets of zero, one or several elementary actions, see explanation in section 3.2). The first decision to be proposed must be the *DN*, so that other decisions can be compared to just leaving things as they are (in other words, the current state of the world must be assessed first).

Experts exchange practical arguments to assess the performance of the proposed decisions, as well as epistemic arguments to support or refute those assessments. They also explicitly define the relations of attack and preference between epistemic arguments. Additionally, the initiator agent may contribute with unconditional epistemic knowledge to the discussion, functioning as absolute facts (presumably provided by the human user), i.e. epistemic arguments which may not be attacked and which are automatically preferred to all arguments they attack.

The order of turns is defined by the initiator agent. Once the stage begins, the initiator offers a turn to the first expert. Once the expert finishes his turn, he communicates to the initiator his intention to pass the turn to the next expert agent (alternatively, the initiator may forcibly end the expert's turn if he sees fit to do so). The initiator then instructs the next expert agent to speak, and so on. Additionally, the initiator agent may utter anything he wishes between turns. Experts may take as many turns as the initiator allows them to. Normally, it makes sense to finish the *Discussion* stage once no experts have anything to add over two consecutive turns (nevertheless, the initiator may finish the stage prematurely, for instance if he has to function under time constraints).

As explained in chapter 4, the combination of an expert agent's preferences over epistemic arguments must define a partial or total preorder, i.e. at the end of an expert agent's turn it must be possible to build a relation with the following properties (assume the existence of a relation $prefer$ over a generic set X, with $(x,y) \in prefer$ meaning that element x is **at least as preferred** as y):

- **Reflexivity**: $\forall x \in X, (x,x) \in prefer$;
- **Transitivity**: $\forall x,y,z \in X$, if $(x,y) \in prefer$ and $(y,z) \in prefer$, then $(x,z) \in prefer$.

Furthermore, two elements x and y are considered **equivalent** if both $(x,y) \in prefer$ and $(y,x) \in prefer$, and **incomparable** if $(x,y) \notin prefer$ and $(y,x) \notin prefer$ (in which case, concerning epistemic arguments, they are assumed to be of equivalent strength until declared otherwise). Also, x is **strictly preferred** to y if $(x,y) \in prefer$ and $(y,x) \notin prefer$.

Finally, notice that, in chapter 4, we implicitly assumed that the set of goals $Goals$, the function mapping goals to values Val, the set of elementary actions $ELAC$ and the binary relation representing incompatible actions I were given *a priori*. Naturally, I provide the necessary mechanisms in the communication protocol for agents to construct that information during the *Exploration* and *Discussion* phases. But a problem may arise from letting expert agents conduct this process: it could be that there is disagreement about elementary things such as what personal goals are legitimate, if goals are associated to the proper values, if an action is technically feasible or whether some pairs of actions are incompatible. Such discussions fall outside the scope of my theory of argumentation (though it could easily be adapted for this purpose with some minor modifications), so experts may have to conduct embedded dialogues of persuasion in order to convince others of the validity of such utterances, or to be convinced to retract them.

Transition rules: the initiator agent explicitly defines when to move on to stage *Evaluation* via a dedicated locution. Once the dialogue has moved past the *Discussion* stage, it may not revert to this or previous stages unless the initiator agent forcibly retracts subsequent utterances, as explained for previous stages.

7. Evaluation

Experts take up one turn each to define their individual preference order over all the proposed decisions. The combination of an agent's preferences over decisions must define a total preorder, i.e. it at the end of the agent's turn it must be possible to build an individual preference relation with the same properties as explained in stage *Discussion* for preference over epistemic arguments, plus the requirement of totality. In a total preorder, no two elements are incomparable.

- **Totality**: $\forall x,y \in X, (x,y) \in prefer$ or $(y,x) \in prefer$ (i.e. at least one, possibly both are defined).

As explained before, experts may only base their evaluation on objectively accepted practical arguments under skeptically preferred semantics. Additionally, the criteria for computing goal satisfaction under uncertainty may be enforced by the initiator agent.

Once the initiator agent gathers the individual preference order of all expert agents, he is finally able to compute the joint decision preference order and present the results to the user.

Transition rules: once all experts have taken up one turn, the dialogue may move on to stage *Finalization*. Reverting directly to stage *Evaluation* after reaching stage *Finalization* makes no sense, since it would amount to repeating the same computations (naturally, as explained before, it might make sense to revert further back in the dialogue and, in that case, go through *Evaluation* again).

8. Finalization

After computing joint preference order and presenting to the user all the relevant information acquired during the deliberation, the dialogue may either be restarted at a previous stage or terminated.

Some potential reasons for reverting to previous stages have been explained throughout their respective descriptions. For example, the user may be unsatisfied with the results of the dialogue and thus choose to refine the problem inputs so that agents have more detailed information to work on. When reverting to a previous stage, the initiator agent must retract all relevant locutions uttered after that stage (if he wants to start prior to its termination) or since its beginning (if he wishes to start at its beginning; note that this distinction doesn't make sense for every stage). Some locutions are merely auxiliary (such as requesting expert assistance) and can't be retracted, while others add proper content to the discussion and thus may be retracted. This distinction will be made clearer in the next section.

Transition rules: the dialogue ends when the initiator agent utters a dedicated locution. The initiator agent may restart the dialogue at a previous stage by retracting locutions.

6.3 Locutions

Now I need to formally define the locutions that allow agents to carry out a dialogue in accordance with the stages presented over the previous section. Each locution serves a different purpose and is appropriate at different moments in the dialogue. While one might be able to devise alternative ways to formalize the dialogue, I think the locutions presented here strike the right balance between taking full advantage of the features presented in earlier chapters and providing enough freedom for implementers to adapt the protocol to different situations.

Before articulating each locution, it is necessary to introduce some concepts that are common to all. All utterances should keep the following metadata:

- an unique identifier;

- a timestamp;
- the sender;
- the list of intended recipients.

Locutions are split into two categories: auxiliary and primary. Auxiliary locutions add no proper content to the dialogue: their only function is to provide the necessary mechanisms to conduct it, such as requesting expert agent assistance, retracting prior utterances, providing synchronization, etc. Primary locutions add actual content to the dialogue, like properties, arguments, goals, preferences, etc.

At any given time, an agent may need to know what utterances were previously exchanged. Therefore, agents need to keep a *Conversation Log*, or *CL* for short. The *CL* is a collection of the valid primary locutions uttered by all agents ordered from the earliest to the most recent[1]. Auxiliary locutions aren't kept because they contain no essential information[2]. While expert agents should always try to store the *CL* locally for performance reasons, the only source guaranteed to be completely reliable in tracking utterances is the initiator agent, as he is present during the whole dialogue (should he leave, the dialogue is automatically terminated, no matter if it is on purpose or accidentally)[3]. Consequently, when an expert agent joins a dialogue that is already underway, he should ask the initiator for synchronization.

I have defined 28 locutions in total: 14 auxiliary and 14 primary. Appendix A provides a detailed explanation of the purpose and rules of each locution. In this section, we need only understand the general functionality of each locution. Tables 6.1 and 6.2 provide a list of the auxiliary and primary locutions, respectively, each accompanied by a short description. Within each table, the locutions are roughly in the same order by which they would normally first appear in the dialogue.

6.4 Analysis of the communication protocol

Now that my communication protocol has been defined, I need to provide an assessment of its quality in light of the desirable features it should exhibit. In order to establish what those features are, I turn to the work of authors experienced in building agent protocols for multiple purposes. McBurney et al. [MPW02] have proposed a set of desiderata for argumentation protocols, drawing on elements of various research domains such as argumentation and political theory. In total, they establish 13 desiderata, which I'll briefly introduce and analyze in light of my protocol, one by one. If the reader is more interested in an assessment of my theory in light of its outcomes, please refer to section 5.4.

[1] Under special conditions, primary locutions *prefer* and *say* may not be added to the *CL*, see their definitions for details.

[2] An agent may nevertheless keep track of every utterance, primary or not, for later analysis, e.g. for purposes of debugging or performance testing.

[3] Naturally, if the initiator drops out of the dialogue accidentally, he might recover it by keeping the *CL* in non-volatile memory (as he should, at least until the dialogue is terminated on purpose).

Table 6.1: Quick reference list of auxiliary locutions

Name of the locution	**Description**
request_assistance	Uttered by the initiator agent to request assistance from some expert agent.
grant_assistance	Signifies readiness of the sender to participate in the dialogue. Normally uttered by an expert in response to a *request_assistance* locution. Alternatively, the sender may utter this locution to express willingness to participate without previous request.
deny_assistance	Uttered by an expert in response to a *request_assistance* locution. Signifies that the expert declines to participate in the dialogue.
withdraw_dialogue	Uttered by an expert. Signifies that the expert no longer wishes to participate in the dialogue. This means that the agent abandons the dialogue midway.
request_sync	Uttered by an expert agent to request an utterance of *provide_sync* by the initiator agent back to the expert.
provide_sync	Uttered by the initiator agent in response to a *request_sync*. Together with *request_sync*, this locution allows expert agents to synchronize *CL* with the initiator agent.
retract	Allows agents to retract a previous primary locution.
retract_bulk	Allows the initiator to retract a locution and all locutions uttered afterwards. This is particularly useful to revert to a previous stage in the dialogue. It is equivalent to calling *retract* multiple times. Promotes efficient communication.
dispute	Allows an expert to ask for justification of some primary locution of type *define_goal*, *propose_action* or *incompatible_actions* uttered previously by another expert, whose discussion falls outside the scope of my theory of argumentation (see description of stage *Discussion*).
take_turn	Allows the initiator to instruct an expert to begin his turn, in accordance with the rules of the current stage (either *Issue discovery*, *Exploration*, *Discussion* or *Evaluation*).
end_turn	Ends the turn of the recipient of the last *take_turn* utterance, allowing other agents to take their turns.
discover_issues	Uttered by the initiator to instruct the experts to look for issues in the state of the world, as defined by its properties (*define_property*) and/or simulation model files (*define_model*).
report_issue	Used by experts to report an issue to the initiator after a *discover_issues* request. Must be uttered once for each separate issue.
end_dialogue	Signifies normal termination of the dialogue by the initiator (normally uttered at the *Finalization* stage, but this is not required).

Table 6.2: Quick reference list of primary locutions

Name of the locution	**Description**
define_property	The initiator agent utters this locution to define a property of the state of the world.
define_model	Allows the initiator agent to provide experts with access to the files necessary for simulating the state of the world. There may be at most one definition of a simulation model at any given time during the dialogue.
enforce_criterion	Allows the initiator agent to enforce the criterion to be used when computing overall goal satisfaction in decision-making under risk or uncertainty/ignorance, as defined in section 5.2.
define_question	The initiator agent uses this locution to define the dialogue's governing question.
define_goal	Allows an expert agent to define a personal goal.
propose_action	Experts utter this locution to propose elementary actions, which may later be used for proposing potential decisions, i.e. combinations of zero, one or several elementary actions.
incompatible_actions	Experts utter this locution to point out incompatibility between two elementary actions.
propose_decision	Experts utter this locution during the *Discussion* stage to propose decisions they would like to see considered in the debate.
assert	Agents utter this locution to put forward an epistemic argument.
attack	Agents utter this locution to express a relation of attack between two epistemic arguments.
assess_decision	Expert agents may utter this locution to put forward a practical argument. The parameters of this locution are coincident with the definition of practical arguments from section 4.1.2.
prefer	Expert agents may utter this locution to express preference between epistemic arguments during the *Discussion* stage, or between decisions during the *Evaluation* stage.
evaluate	Uttered by the initiator to order the experts to move from the *Discussion* stage to the *Evaluation* stage.
say	This locution allows agents to exchange any kind of information that falls outside the regular scope of the dialogue, for example in order to intertwine other types of dialogues (e.g. negotiation dialogues).

Let's then proceed to assessing my protocol in light of the 13 desiderata, in the same order as they are presented by McBurney et al.

1. **Stated Dialogue Purpose**
   Description: "A dialectical system should have one or more publicly-stated purposes, and its locutions and rules should facilitate the achievement of these".
   Assessment: The purpose of the protocol is clear: enabling deliberation dialogues among computational agents, with a strong focus on assisting human decision-makers using a combination of DSS, argumentation, simulation and potentially other technologies. The specific purpose of each dialogue is stated via the *define_question* locution. Additionally, the protocol supports an innovating feature that allows agents to autonomously discover and suggest problems to discuss.

2. **Diversity of Individual Purposes**
   Description: "A dialectical system should permit participating agents to achieve their own individual purposes consistent with the overall purpose of the dialogue".
   Assessment: This is one of the major underlying assumptions of my theory, and is extensively discussed in practically all sections. The initiator agent's purpose is to act on behalf of the user's interests, while each expert agent will incorporate different goals to uphold and variations in preference between goals, values and arguments, which will translate into different preferences for decisions.

3. **Inclusiveness**
   Description: "A dialectical system should not preclude participation by any potential agent which is qualified and willing to participate".
   Assessment: There are no limitations on what expert agents can participate in the dialogue, other than that they know the protocol and adhere to its rules, obviously. Both participation by invitation and spontaneous participation modes are implemented, as per definition of locutions *request_assistance*, *grant_assistance* and *deny_assistance*.

4. **Transparency**
   Description: "Participants to a dialogue should know the rules and structure of the dialectical system prior to commencement of the dialogue".
   Assessment: The rules of the protocol are transparent: the set of admissible locutions and their structure, preconditions and effects are all articulated in detail in appendix A.

5. **Fairness**
   Description: "A dialectical system should either treat all participants equally, or, if not, make explicit any asymmetries in their treatment".
   Assessment: The protocol rules are asymmetrical because the initiator agent has different rights and obligations from them expert agents, but these differences are known to all participants. All expert agents are treated equally.

6. **Clarity of Argumentation Theory**
   Description: "A dialectical system should conform, at least at the outset, to a stated theory of argument".
   Assessment: This is evident; see theory of argument in chapter 4.

7. **Separation of Syntax and Semantics**
   Description: "The syntax of a dialectical system should be defined separately from its semantics".
   Assessment: As the reader is invited to verify in appendix A, preconditions are never formulated in terms of semantics, i.e. the definition of a locution's syntax never relies on the particular meaning of its arguments (parameters) or on the internal state of any agent.

8. **Rule-Consistency**
   Description: "The locutions and rules of a dialogue system should together be internally consistent; that is, they should not lead to deadlocks (where no participant may utter a legal locution), nor infinite cycles of repeated locutions".
   Assessment: It is clear from the rules that, in every stage, at least one agent (the initiator) and potentially all agents (speaking in their turn) are free to add more information to the debate: no preconditions or effects lead to deadlocks, but even if they did, they could always be settled by retracting locutions. There appears to be no reason for infinite cycles either, as preconditions prevent adding repeated content. Naturally, it is always possible that any of these problems might occur, not as a result of the protocol's rules, but due to incorrect agent implementations (i.e. internally, an agent may be in a deadlock or infinite cycle situation). The initiator is responsible for dealing with agents exhibiting systematically erroneous behavior.

9. **Encouragement of Resolution**
   Description: "Resolution of each dialogue (normal termination) should be facilitated, and not precluded, by the locutions and rules of a dialectical system".
   Assessment: The rules make it clear how to transition between stages, and mechanisms for orderly and premature termination are also provided. Furthermore, the initiator agent can force the dialogue to advance by ending expert's turns prematurely, and he may choose to finish the *Discussion* stage at any time by utterance of the *evaluate* locution.

10. **Discouragement of Disruption**
    Description: "Normally, the rules of a dialectical system should discourage or preclude disruptive behavior, such as uttering the same locution repeatedly".
    Assessment: Yes, the preconditions of the locutions prevent most forms of disruptive behavior. Furthermore, retraction (voluntary or forcible) is possible, and the initiator agent has the power to deal with systematically disruptive behavior by expelling the abuser from the dialogue and requesting the assistance of another agent to take his place.

11. **Enablement of Self-Transformation**
    Description: "A dialectical system should permit participants to undergo self-transformation in the course of a dialogue".
    Assessment: This is supported in many ways, the most obvious of which being the ability to exchange arguments, retract locutions and dispute goals, all of which have the potential to make agents change their preferences about decisions.

12. **System Simplicity**
    Description: "The locutions and rules of a dialectical system should be as simple as possible, consistent with the eleven criteria above. In particular, each locution should serve a specific and stated function in the dialogue, and the protocol rules should lead to efficient achievement of the dialogue purposes".
    Assessment: There are no extraneous locutions other than *retract_bulk* (which is equivalent to multiple utterances of *retract*), which is justified to promote efficient communication. Although the protocol has 28 locutions, which may seem to render it complex, it should be noted that many locutions are accessory and serve mostly to aid setup and control flow (many authors don't consider this), such as *request_assistance*, *grant_assistance*, *deny_assistance*, *withdraw_dialogue*, *request_sync*, *provide_sync*, *retract_bulk*, *take_turn*, *end_turn*, *end_dialogue* and also *say*, because its purpose is generic. That leaves us with 17 locutions serving the purpose of the debate per se, which seems quite reasonable, considering the richness of the underlying theory. Furthermore, some locutions could easily be condensed into one, for instance *define_property* and *define_model*, or *propose_action* and *propose_decision*, but it was preferred to keep them separate to simplify the articulation of the rules and facilitate implementation.

13. **Computational Simplicity**
    Description: "A dialectical system should be designed to minimize any computational demands on its participants, and on the system itself, consistent with the twelve criteria above".
    Assessment: The running-time of a debate conducted under this protocol varies greatly according to many implementation-defined factors, such as the complexity of the problem to solve, the number of experts, the depth of their base knowledge, the discrepancies between their beliefs, the variety of goals, the complexity of their metrics and the number of decisions to consider. Each stage has a small impact on a dialogue's running time, but by far the stage that should contribute most to the debate's computational complexity is *Discussion*. Computing the objectively accepted arguments under skeptically preferred semantics is an expensive process, but novel SAT-based approaches [CDGV14] are generally capable of calculating the preferred extensions for argumentation frameworks with more than a hundred arguments in less than a second on ordinary hardware.

In light of this assessment, overall I consider my protocol to exhibit all desirable features that one may reasonably expect. All 13 desiderata are fulfilled, though desiderata 8 (Rule-Consistency) and 13 (Computational Simplicity) are dependent on the specifics of each implementation to a certain extent. I also believe that my protocol strikes a better balance between generality of application and practical usefulness than most other works in literature, as it exhibits an orderly structure of conversation where each stage serves a very well defined purpose, and is grounded on a clear theory of decision-making and argumentation, thus greatly facilitating the work of implementers. At the same time, novel features are presented, such as the possibility of using simulation, requesting issue discovery, allowing multiple ways to enter the dialogue, using different criteria, representing complementary and incompatible actions and expressing individual preferences over decisions, among others.

6.5 Summary

Combining all the theory developed throughout chapters 3, 4 and 5, I devised a protocol for agent communication grounded on the features of my argumentation model.

I started by providing an overview of the protocol's architecture. Then I divided the dialogues conducted under my protocol in eight major stages, and carefully explained the purpose of each: ***Reunion***, during which the initiator agent gathers an assembly of expert agents; ***Input definition***, during which the initiator conveys the inputs of the problem to the experts; ***Issue discovery***, an optional stage allowing experts to report issues in the representation of the state of the world back to the initiator; ***Question formulation***, where the initiator clearly defines the question to be resolved; ***Exploration***, during which experts define their personal goals and propose elementary actions; ***Discussion***, where experts exchange epistemic and practical arguments about decisions; ***Evaluation***, during which experts rank-order decisions according to individual preferences and accepted arguments; and ***Finalization***, which allows the initiator to restart the dialogue at a previous stage with different inputs, or to dismiss the experts, thus ending the dialogue. I also introduced the locutions that allow agents to conduct dialogues in accordance with the protocol (which are explained in detail in appendix A), and provided an assessment of the protocol in light of some desirable properties.

Chapter 7

An example dialogue

I have established the key concepts underlying my theory, laid down the rules of my argumentation framework, suggested some criteria for choosing decisions, and built a protocol to take advantage of all those features plus some more. Now it's time to put everything together and demonstrate how my theory can be used in a real-life scenario.

I'll show an example dialogue in the context of planning improvements for an urban transportation network. As I've explained before, the topic of transportation planning is too vast and complex to be fitted into this work with the all justice it deserves. In section 2.3, we only saw a small selection of introductory theory. Should the reader wish to know more about this topic, I reiterate my recommendation for specialized literature such as the Highway Capacity Manual [HCM10]. The example provided in this chapter will only deal with topics that should be reasonably familiar to the general public.

7.1 The scenario

Before we can begin to explore the example dialogue, I need to introduce the scenario we'll be dealing with during the rest of this chapter. Figure 7.1 shows a simplified illustration of a small section of an urban traffic network[1] (created with the Edraw Max software). The scenario is focused on three streets, which intersect near a middle school. Streets A and B are connected to two major residential areas. Street C connects to a highway access, which in turn allows access to the industrial and shopping districts, where most adult citizens spend their workdays. Vehicles travel on the right side of the road, as shown by the white arrows on the start of each street. Vehicles arriving at the intersection from street A must give way because of the stop sign. The school has two gates, placed near crosswalks on streets A and B. All streets have sidewalks on both sides, plus A and B feature some parking spaces (free of charge). There are no traffic lights of any kind. Some private lands, composed mostly of uninhabited fields, surround the region.

[1] This example is loosely based on a real-life scenario in Portugal.

To simplify our scenario, let's assume that most citizens living in residential areas A and B commute to work via the highway, and that the most direct path to it goes through street C (coming from A or B depending on their area of residence). Furthermore, assume that the residential areas aren't connected to each other by any streets other than A and B (if this sounds implausible, imagine that other paths exist but are far too circuitous).

Many parents drop their children off at school on their way to work. As a consequence, during rush hour, vehicles form long queues before the crosswalks near the school gates. Traffic in street A is particularly slow due to the stop sign near the intersection.

During school intervals and lunchtime, many children cross street A to go to the cake shop (and back to school). Because traffic is relatively fluid during those hours, drivers become less cautious and accidents sometimes happen. This isn't so much the case with street B, since most children cross the street only once when they arrive at school (during rush hour) coming from residential area B in the northbound direction.

The volume of traffic around the school has risen significantly over the past few years. Some citizens expressed their concerns w.r.t. impacts on children's health. This led authorities to install an air monitoring station near the school fence to provide regular measurements of air quality.

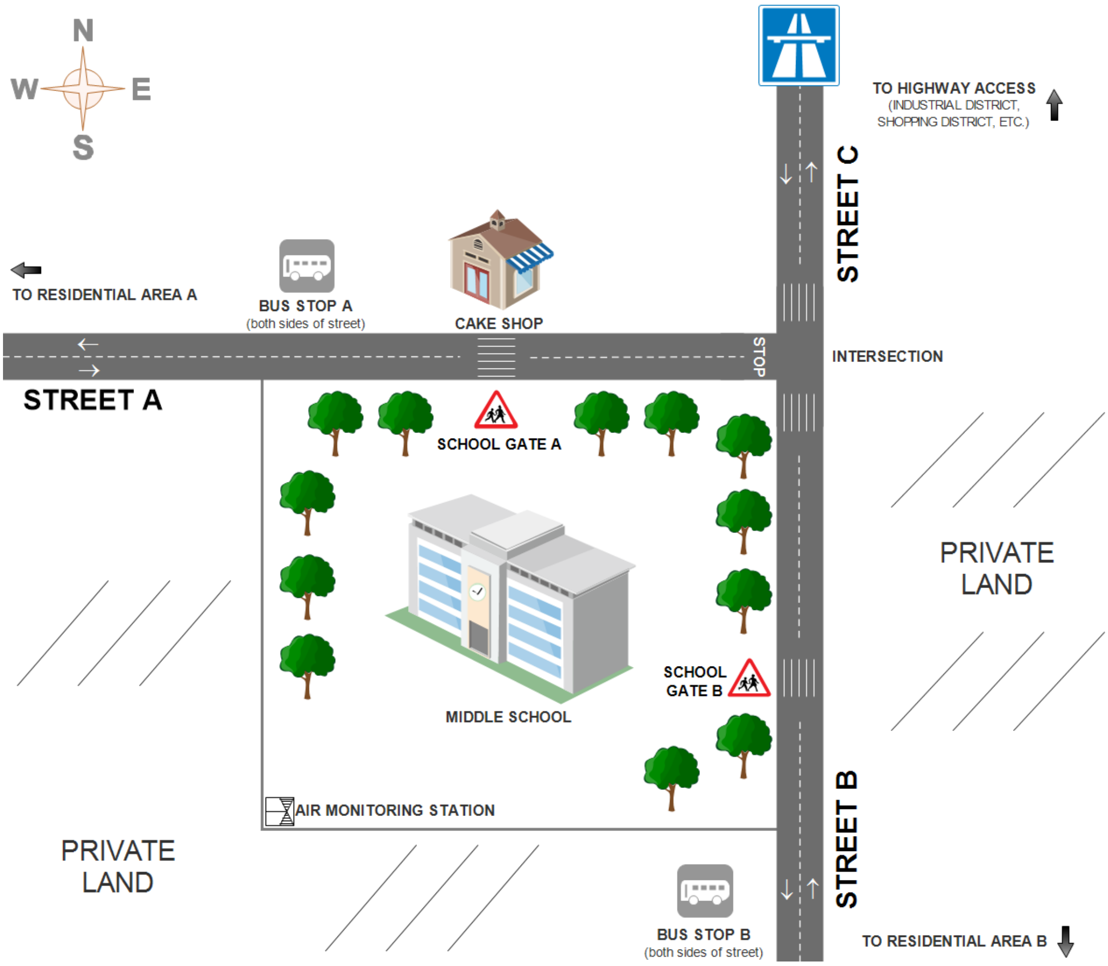

Figure 7.1: An illustration of the scenario

Now that we are acquainted with the traffic network and the problems surrounding it, let's move on to the example dialogue. In the next section, we'll see how a group of expert agents can be solicited to study potential measures in order to improve the situation.

7.2 Requesting help and conveying the situation

Let's assume the existence of an initiator agent who will act on our behalf. From now on, I'll designate him *Ini*. First, he has to gather an assembly of experts. We'll request the assistance of three experts with different roles:

- **Mobility expert**: studies improvements to the traffic network in order to increase its safety and capacity, so as to serve demand more satisfactorily. I'll designate him *Mob*;
- **Environment expert**: ensures that air pollutant and sound emissions are kept within acceptable levels for human health. I'll designate him *Env*;
- **Finances expert**: examines implementation costs, so as to not let the debate degenerate into unrealistic propositions in light of the available budget. I'll designate him *Fin*.

Ini utters his first locution, and thus the dialogue starts, triggering the *Reunion* stage. He requests assistance from *Mob*.

U1: [*Ini* -> *Mob*] **request_assistance** ( )

Hopefully, this representation of utterances should be intuitive. First we have the identifier, **U1**, followed by the sender, *Ini*, and the recipient(s), in this case *Mob* (when dealing with utterances targeted at all participants, the recipients list will only contain the word *All*). Next we have the type of the locution, *request_assistance*, followed by the list of arguments within parentheses, which in this case is empty. To keep it simple, we won't see in this dialogue any utterances of types *deny_assistance, withdraw_dialogue, request_sync, provide_sync, retract_bulk, dispute* or *say*, as they would add little to the example.

U2: [*Mob* -> *Ini*] **grant_assistance** ( )

Mob accedes to *Ini's* request. The same process happens with *Env* and *Fin*.

U3: [*Ini* -> *Env*] **request_assistance** ( )

U4: [*Env* -> *Ini*] **grant_assistance** ( )

U5: [*Ini* -> *Fin*] **request_assistance** ( )

U6: [*Fin* -> *Ini*] **grant_assistance** ( )

Now we can move on to stage *Input definition*. Let's assume that we have a simulation model of the scenario, including information about the routes of every vehicle passing in the network during a normal weekday (both passenger vehicles and buses), as well as pedestrians. Furthermore, assume this model is built as a series of SUMO [KEBB12] files:

- **network.net.xml**: Models the configuration of the network (e.g. streets, junctions and crosswalks);
- **bus_stops.add.xml**: Indicates the location of the bus stops;
- **routes.rou.xml**: Contains collected information about the routes of all vehicles (passenger vehicles, buses, etc.) and pedestrians in a full weekday. Each vehicle is associated to a type defined in file *types.add.xml*;
- **types.add.xml**: Contains information about the characteristics of the different types of vehicles and pedestrians passing in the network (for instance, vehicles may comply with different emission norms, like Euro 2 and Euro 3);
- **detectors.add.xml**: Defines the location and type of different detectors in the network, which may be used, for example, to compile statistics on traffic, emissions and noise.
- **run.sumo.cfg**: The configuration file, referencing all previous files, which must be loaded in SUMO.

These simulation files are made available to the expert agents as a series of FTP links via the *define_model* locution, triggering the *Input definition* stage.

U7: [*Ini -> All*] **define_model** (*[ftp://193.136.28.205/scenario1/network.net.xml ,*
ftp://193.136.28.205/scenario1/bus_stops.add.xml ,
ftp://193.136.28.205/scenario1/routes.rou.xml ,
ftp://193.136.28.205/scenario1/types.add.xml ,
ftp://193.136.28.205/scenario1/detectors.add.xml ,
ftp://193.136.28.205/scenario1/run.sumo.cfg])

Not every piece of information can be represented within the simulation model. Therefore, some additional properties can only be defined separately. What follows are a few examples of such properties. Because this is an example dialogue for human readers, I'll only employ sentences in natural language.

U8: [*Ini -> All*] **define_property** (*"Budget for immediate improvements in the region is 1 million euros"*)

U9: [*Ini -> All*] **define_property** (*"Predicted average population growth of 0.8% per year during the next 10 years"*)

U10: [*Ini -> All*] **define_property** (*"Last year, there were 3 accidents involving pedestrians near school gate A"*)

U11: [*Ini -> All*] **define_property** (*"Street A provides 12 free parking spaces in eastbound direction, 6 in westbound direction; street B provides 14 free parking spaces in northbound direction, 8 in southbound direction"*)

U12: [*Ini -> All*] **define_property** (*"Parking spaces are often occupied early in the day by teachers and school staff, so there is little opportunity for dropping people off without disturbing traffic flow in the same direction"*)

U13: [*Ini -> All*] **define_property** (*"Recent measurements from air monitoring station yielded values of particulate matter nearing limits established by the European Comission"*)

U14: [*Ini -> All*] **define_property** (*"Traffic noise levels measured recently at the school yard were within acceptable values, with* $L_{eq(24)}$ *= 50 dBA, EPA recommends* $L_{eq(24)} \leq$ *55 dBA"*)

While we're at it, let's enforce the maximum expected utility criterion for decision-making under risk, and the maximin rule (pessimism) when dealing with uncertainty.

U15: [*Ini -> All*] **enforce_criterion** (*"RISK", "MAXIMUM_EXPECTED_UTILITY"*)

U16: [*Ini -> All*] **enforce_criterion** (*"UNCERTAINTY", "MAXIMIN"*)

After the introduction to the scenario, some problems and opportunities for improvement should be pretty obvious to the reader. In any case, one could ask the experts to discover issues on their own and report them back, triggering the *Issue discovery* stage. The first parameter of the *report_issue* locution is a real number between 0 and 1, representing the issue's level of severity from the perspective of the expert.

U17: [*Ini -> All*] **discover_issues** ( )

U18: [*Ini -> Mob*] **take_turn** ( )

U19: [*Mob -> Ini*] **report_issue** (*1, "Major safety issues near school gates, with vast room for improvement"*)

U20: [*Mob -> Ini*] **report_issue** (*0.75, "Intersection provides inefficient distribution of traffic flow, with vast room for improvement"*)

U21: [*Mob -> Ini*] **report_issue** (*0.75, "Congestions near school gates at rush hour due to lack of dedicated traffic facilities, some room for improvement"*)

U22: [*Mob -> Ini*] **report_issue** (*0.5, "Lack of efficient, alternative highway accesses"*)

U23: [*Mob -> All*] **end_turn** ( )

U24: [*Ini -> Env*] **take_turn** ( )

U25: [*Env -> Ini*] **report_issue** (*0.5, "Observed levels of particulate matter emissions could be hazardous to children's health if waiting in the street during long periods of time, some room for improvement"*)

U26: [*Env -> Ini*] **report_issue** (*0.25, "Projected levels of oxides of nitrogen emissions, using simulation, reveal higher than average values, with potential risks to vegetation and human health if traffic increases substantially"*)

U27: [*Env -> All*] **end_turn** ( )

U28: [*Ini -> Fin*] **take_turn** ( )

U29: [*Fin -> Ini*] **report_issue** (*0.25, "Plenty of unexploited opportunities to foster economic development in the region, given high volume of traffic and uninhabited lands"*)

U30: [*Fin -> All*] **end_turn** ( )

Now we have to define the governing question of the debate, triggering the *Question formulation* stage. Looking at the issues reported by the experts, its clear that the safety issues near the school gates should be the most serious immediate concern, and one that could be easily ameliorated, if not resolved altogether. Let's focus on discussing potential decisions addressing that issue, while also attempting to improve traffic flow.

U31: [*Ini -> All*] **define_question** (*"What should be done to improve safety around the school gates, while attempting to boost traffic flow?"*)

7.3 Defining goals and actions

Moving on to the *Exploration* stage, it is time for experts to define their personal goals and propose actions to deal with the situation. I'll keep the order of turns *Mob->Env->Fin* for the rest of the dialogue. Just remember that *Ini* is allowed to speak at any time between the experts' turns.

U32: [*Ini -> Mob*] **take_turn** ( )

U33: [*Mob -> All*] **define_goal** (*"Reduce accidents to the greatest possible extent, especially those involving pedestrians", "Safety"*)

U34: [*Mob -> All*] **define_goal** (*"Ensure low waiting times for vehicles at street junctions", "Mobility"*)

U35: [*Mob -> All*] **define_goal** (*"Ensure acceptable traffic flow in street segments at all times", "Mobility"*)

U36: [*Mob -> All*] **propose_action** (*"Place traffic lights on crosswalks near school gates"*)

U37: [*Mob -> All*] **propose_action** (*"Replace crosswalks near school gates with pedestrian bridges"*)

U38: [*Mob -> All*] **incompatible_actions** (*"Place traffic lights on crosswalks near school gates", "Replace crosswalks near school gates with pedestrian bridges"* [2])

U39: [*Mob -> All*] **propose_action** (*"Place traffic lights on intersection"*)

U40: [*Mob -> All*] **propose_action** (*"Replace intersection with roundabout"*)

U41: [*Mob -> All*] **incompatible_actions** (*"Place traffic lights on intersection", "Replace intersection with roundabout"*)

U42: [*Mob -> All*] **propose_action** (*"Substitute parking places in street A eastbound and street B northbound with separate lanes for dropping people off and build park in land adjacent to school"*)

U43: [*Mob -> All*] **end_turn** ( )

To evaluate the fulfillment of his goals, we'll assume that *Mob* uses some of the metrics presented in section 2.3.3, in particular those related to LOS at street segments and junctions. The same principle applies to *Env* regarding metrics for air pollution and noise. Naturally, some goals are more difficult to evaluate on the basis of a standard procedure (such as those related to safety issues), but we'll ignore the details of their assessment.

U44: [*Ini -> Env*] **take_turn** ( )

U45: [*Env -> All*] **define_goal** (*"Maintain emissions of air pollutants within tolerable levels with regard to the health of humans and vegetation", "Environment"*)

U46: [*Env -> All*] **define_goal** (*"Ensure that environmental noise, including that generated by traffic, is kept within acceptable values for human health", "Environment"*)

U47: [*Env -> All*] **propose_action** (*"Enforce law to retrofit older vehicles with emission control equipment using economic incentives"*)

[2] Notice how I referred to the actions explicitly, instead of using the identifiers of their respective utterances. The only reason for this is to facilitate the reader's comprehension. Between computational entities, it would certainly seem more efficient to simply refer to the identifiers. The protocol does not specify this requirement, so as to not delve into semantic details. Nevertheless, it is advisable that utterances referring to content from previous utterances should cite their identifiers directly. This applies to *retract, retract_bulk, dispute, incompatible_actions, propose_decision, attack, assess_decision* and *prefer*, and possibly *say*, depending on its contents.

U48: [*Env* -> *All*] **propose_action** (*"Prohibit circulation of vehicles not conforming with Euro 3 standard around school area"*)

U49: [*Env* -> *All*] **propose_action** (*"Decrease prices of public transportation offers"*)

U50: [*Env* -> *All*] **end_turn** ( )

U51: [*Ini* -> *Fin*] **take_turn** ( )

U52: [*Fin* -> *All*] **define_goal** (*"Ensure decisions don't exceed budget", "Finances"*)

U53: [*Fin* -> *All*] **end_turn** ( )

It is now possible to calculate the preference orders between goals. First of all, we need the agents' orders on values.

- ***Mob***: *Safety > Mobility > Environment > Finances*
- ***Env***: *Safety > Environment > Mobility > Finances*
- ***Fin***: *Safety > Finances > Mobility > Environment*

Now we need the agents' orders on personal goals. I'll precede each goal with an identifier, so as to make subsequent references less verbose. Because *Fin* only defined one goal - *"Ensure decisions don't exceed budget"* - I'll omit him from this list (his goal shall be identified as G_6).

- ***Mob***: *"G_1 - Reduce accidents to the greatest possible extent (...)" > "G_2 - Ensure low waiting times for vehicles at street junctions" ≈ "G_3 - Ensure acceptable traffic flow in street segments at all times"*
- ***Env***: *"G_4 - Maintain emissions of air pollutants within tolerable levels (...)" > "G_5 - Ensure that environmental noise (...) is kept within acceptable values"*

Now we can finally define *Goalprefs*, i.e. the complete goal preference order of each expert agent.

- ***Mob***: $G_1 > G_2 \approx G_3 > G_4 \approx G_5 > G_6$
- ***Env***: $G_1 > G_4 > G_5 > G_2 \approx G_3 > G_6$
- ***Fin***: $G_1 > G_6 > G_2 \approx G_3 > G_4 \approx G_5$

7.4 Debating potential decisions

Now the dialogue can move on to the *Discussion* stage, with experts proposing decisions and exchanging arguments. In order not to extend our example more than necessary, I won't cover every possible decision (to do so, we'd need to consider every possible unique combination of non-mutually exclusive elementary actions). For simplicity's sake, we'll consider each decision at a time, starting with the *DN*, which amounts to a proposal with an empty list of actions.

U54: [*Ini -> Mob*] **take_turn** ( )

U55: [*Mob -> All*] **propose_decision** (*[]*)

I'll structure epistemic arguments as a list of comma-separated premises, followed by *"->"*, followed by a conclusion. The support set of the practical arguments will always concern at least three aspects: the relevant properties of the world, the foreseeable consequences of decisions given those properties and the impact of those consequences on goals.

U56: [*Mob -> All*] **assert** (*"Positive population growth -> Traffic not likely to decline"*)

U57: [*Mob -> All*] **assert** (*"Traffic not likely to decline, 3 accidents near school gates last year, Nothing is done -> Frequency of accidents will stay similar or increase"*)

U58: [*Mob -> All*] **assert** (*"3 accidents near school gates last year, Frequency of accidents will stay similar or increase -> Impossible to fulfill aim of reducing accidents if nothing is done"*)

U59: [*Mob -> All*] **assess_decision** (*[U56, U57, U58], DN, "Reduce accidents(...)", [<-1, 1>]*)

Thus the dialogue's first practical argument concerns *DN* and the goal of reducing accidents. The support set consists of the epistemic arguments from utterances U56, U57 and U58, and *Mob* scores the sole outcome with value -1 (which is considered unsatisfactory, as it falls below 0), with probability 1 (100% certainty). Let's assume that *Mob* ranks decisions w.r.t. this particular goal in three possible ways: -1 when the frequency of accidents is presumed to remain the same or increase, 0 when the frequency of accidents is presumed to decrease and 1 when it is presumed highly unlikely than any more accidents will occur.

U60: [*Mob -> All*] **assert** (*"Simulation reveals waiting times averaging 40 seconds at intersection during rush hour for vehicles coming from street A, Traffic not likely to decline, Nothing is done -> Waiting times at intersection will stay the same"*)

U61: [*Mob -> All*] **assert** (*"Waiting times at intersection will stay the same -> Intersection has LOS E for vehicles coming from street A if nothing is done"*)

Because LOS has six levels, *Mob* can convert it to a numeric scale with values -1, -0.6, -0.2, 0.2, 0.6 and 1, corresponding to levels F to A.

U62: [*Mob -> All*] **assess_decision** (*[U60, U61], DN, "Ensure low waiting times (...) at street junctions", [<-0.6, 1>]*)

U63: [*Mob -> All*] **assert** (*"Simulation reveals average speed at street A of 25% of free flow speed during rush hour in eastbound direction, Simulation reveals average speed at street B of 45% of free flow speed during rush hour in northbound direction, Traffic not likely to decline, Nothing is done -> Traffic flow on all streets will stay the same"*)

U64: [*Mob -> All*] **assert** (*"Traffic flow on all streets will stay the same -> Street A has LOS F and street B has LOS D if nothing is done"*)

U65: [*Mob -> All*] **assess_decision** (*[U63, U64], DN, "Ensure acceptable traffic flow(...)", [<-0.6, 1>]* [3])

U66: [*Mob -> All*] **end_turn** ( )

Let's now see *Env's* assessment of *DN*. Assume that *Env's* interpretation of the current state of air pollution in the region is that it merits regular observation, but there is no immediate danger to health, though some measures could be taken to prevent the escalation of the problem. Furthermore, *Env* calculates satisfaction of the environmental noise goal as a simple subtraction between the limit equivalent sound level for outdoor measurements, recommended by the EPA (55 dBA), and the projected value after the decision is taken. Remember that SUMO contains models both for predicting air pollutant emissions (HBEFA) and noise emissions (Harmonoise) generated by traffic. *Env* can take advantage of these features to perform his assessments.

U67: [*Ini -> Env*] **take_turn** ( )

U68: [*Env -> All*] **assert** (*"Traffic not likely to decrease, Nothing is done -> Air pollutant emissions not likely to decrease"*)

U69: [*Env -> All*] **assert** (*"Observed emissions of particulate matter nearing EU limits, Air pollutant emissions not likely to decrease -> Emissions of particulate matter are not an immediate cause for concern but escalation should be prevented"*)

U70: [*Env -> All*] **assess_decision** (*[U68, U69], DN, "Maintain emissions of air pollutants within tolerable levels(...)", [<0, 1>]*)

U71: [*Env -> All*] **assert** (*"Traffic not likely to decrease, Nothing is done -> Noise emissions not likely to decrease"*)

U72: [*Env -> All*] **assert** (*"Noise emissions not likely to decrease, Measured values outdoors are of $L_{eq(24)}$ = 50 dBA, Measured values comfortably within conservative limit of 55 dBA -> Noise is not a cause for concern if nothing is done"*)

U73: [*Env -> All*] **assess_decision** (*[U71, U72], DN, "Ensure that environmental noise(...)is kept within acceptable values", [<5, 1>]*)

U74: [*Env -> All*] **end_turn** ( )

Finally, it's *Fin's* turn to assess *DN*, which should be pretty straightforward. *Fin* scores decisions based on the budget's balance after consideration of all potential costs.

[3] Assume that *Mob* averages the scores of streets A and B, which are -1 and -0.2 respectively

U75: [*Ini -> Fin*] **take_turn** ( )

U76: [*Fin -> All*] **assert** (*"Budget for immediate improvements in the region is 1 million euros, No immediate costs if nothing is done -> Positive balance of 1 million euros if nothing is done"*)

U77: [*Fin -> All*] **assess_decision** (*[U76], DN, "Ensure decisions don't exceed budget", [<1000000, 1>]*)

U78: [*Fin -> All*] **end_turn** ( )

Now that we have studied the basic decision of simply doing nothing, let's see how the experts could debate other proposals.[4]

U79: [*Ini -> Mob*] **take_turn** ( )

U80: [*Mob -> All*] **propose_decision** (*["Replace crosswalks near school gates with pedestrian bridges"]*)

U81: [*Mob -> All*] **assert** (*"Pedestrian bridges separate vehicle and pedestrian circulation -> No more accidents involving pedestrians near school gates"*)

U82: [*Mob -> All*] **assert** (*"No more accidents involving pedestrians near school gates -> Aim of reducing accidents is completely fulfilled"*)

U83: [*Mob -> All*] **assess_decision** (*[U81, U82], U80, "Reduce accidents(...)", [<1, 1>]*)

U84: [*Mob -> All*] **assert** (*"Simulation reveals that eliminating pedestrian traffic near school gates improves average speed at street B to 60% of free flow speed during rush hour in northbound direction -> Street B has LOS C with pedestrian bridges implemented"*)

U85: [*Mob -> All*] **assert** (*"Simulation reveals that pedestrian bridges don't improve traffic flow in street A probably due to faster flow in street B and need to give way due to stop sign -> Street A keeps LOS F with pedestrian bridges implemented"*)

U86: [*Mob -> All*] **assess_decision** (*[U84, U85], U80, "Ensure acceptable traffic flow(...)", [<-0.4, 1>]*)

U87: [*Mob -> All*] **end_turn** ( )

U88: [*Ini -> Env*] **take_turn** ( )

[4] Notice that experts no longer have to assess each potential decision in light of every single goal: when it is obvious that a decision provokes no effect on a certain goal, the assessment provided for *DN* applies. An example is the goal concerning noise emissions, which isn't likely to be affected by any combination of actions proposed by the agents.

U89: [*Env -> All*] **assert** (*"Studies suggest that pedestrian bridges in urban locations are often avoided by impatient people due to taking more time to cross the street, Children are typically less patient -> Accidents involving pedestrians near school gates are still a possibility"*)

U90: [*Env -> All*] **attack** (*U89, U81*)

U91: [*Env -> All*] **attack** (*U81, U89*)

U92: [*Env -> All*] **prefer** (*U89, U81*)

U93: [*Env -> All*] **attack** (*U89, U82*)

U94: [*Env -> All*] **end_turn** ( )

Through U89, *Env* attacked the conclusions of argument U81 and the premises of argument U82 by citing knowledge from an external source. Because the conclusions of U81 and U89 contradict each other, *Env* recognizes that the attack is mutual, but expresses preference for his own argument (as will *Mob* and *Fin* in subsequent turns). *Mob* will deal with this new information after *Fin's* cost assessment.

U95: [*Ini -> Fin*] **take_turn** ( )

U96: [*Fin -> All*] **prefer** (*U89, U81*)

U97: [*Fin -> All*] **assert** (*"Cost of a building pedestrian bridge across a street of width similar to A and B is around €250K -> Cost of building pedestrian bridges in streets A and B combined amounts to €500K"*)

U98: [*Fin -> All*] **assert** (*"Cost of building pedestrian bridges in streets A and B combined amounts to €500K -> Positive balance of €500K if bridges are built"*)

U99: [*Fin -> All*] **assess_decision** (*[U97, U98], U80, "Ensure decisions don't exceed budget", [<500000, 1>]*)

U100: [*Fin -> All*] **end_turn** ( )

U101: [*Ini -> Mob*] **take_turn** ( )

U102: [*Mob -> All*] **prefer** (*U89, U81*)

Recognizing the attack on some of its assumptions, *Mob* retracts practical argument U83 and rebuilds it in light of new information.

U103: [*Mob -> All*] **retract** (*U83*)

U104: [*Mob -> All*] **assert** (*"Pedestrian bridges separate vehicle and pedestrian circulation -> Building a pedestrian bridge has the potential of completely preventing accidents involving pedestrians as long as the bridge is adopted"*)

U105: [*Mob -> All*] **assert** (*"There is a possibility that some children will avoid the pedestrian bridge -> Full adoption of the bridge cannot be guaranteed"*)

U106: [*Mob -> All*] **assert** (*"Full adoption of the bridge cannot be guaranteed -> Rate of accidents should definitely decrease but might be sub-optimal without awareness campaigns"*)

U107: [*Mob -> All*] **assess_decision** (*[U104, U105, U106], U80, "Reduce accidents(...)", [<0, ?> , <1, ?>]*)

Through his new assessment, *Mob* expresses uncertainty about the outcomes of the decision w.r.t. to the goal of reducing accidents. He encodes this uncertainty as two potential outcomes with unknown probability and scores 0 and 1, corresponding, respectively, to the case that some children ignore the pedestrian crossing, and the case that it is widely adopted.

Recognizing that an awareness campaign among school students could help to reduce safety issues, *Mob* proposes a new elementary action.

U108: [*Mob -> All*] **propose_action** (*"Raise awareness of school students about safety issues"*)

Since we can't explore every possible decision in this example, I'll finish the *Discussion* stage by showing *Mob* propose a decision that is likely to address all goals at a minimally satisfactory level. It is composed of five separate actions.

U109: [*Mob -> All*] **propose_decision** (*["Place traffic lights on crosswalks near school gates", "Raise awareness of school students about safety issues", "Replace intersection with roundabout", "Substitute parking places in street A eastbound and street B northbound with separate lanes for dropping people off and build park in land adjacent to school", "Enforce law to retrofit older vehicles with emission control equipment using economic incentives"]*)

U110: [*Mob -> All*] **assert** (*"Traffic lights significantly decrease drivers' disrespect for crosswalks, Awareness campaigns at school should encourage safe habits for children -> Rate of accidents very likely to decrease"*)

U111: [*Mob -> All*] **assess_decision** (*[U110], U109, "Reduce accidents(...)", [<0, 1>]*)

U112: [*Mob -> All*] **assert** (*"Simulation reveals that average waiting times at roundabout from all directions are inferior to 15 seconds -> Roundabout LOS is B"*)

U113: [*Mob -> All*] **assess_decision** (*[U112], U109, "Ensure low waiting times (...) at street junctions", [<0.6, 1>]*)

U114: [*Mob -> All*] **assert** (*"Creation of separate lanes for dropping people off eliminates unnecessary congestions, Simulation shows that eliminating congestions near school gates plus construction of roundabout assures smooth traffic flow averaging 75% of free flow speed in all streets -> Streets A and B have LOS B"*)

U115: [*Mob -> All*] **assess_decision** (*[U114], U109, "Ensure acceptable traffic flow(...)", [<0.6, 1>]*)

U116: [*Mob -> All*] **end_turn** ( )

U117: [*Ini -> Env*] **take_turn** ( )

U118: [*Env -> All*] **assert** (*"Nearly 30% of vehicles in the region are not fitted with up-to-date emission control equipment, Full application of law will take over a year -> Enforcing law will impact nearly 30% of vehicles after a year"*)

U119: [*Env -> All*] **assert** (*"Enforcing law will impact nearly 30% of vehicles after a year, Simulation reveals that decreasing emissions of 30% of vehicles to the standards of modern vehicles is sufficient to ward off air pollution concerns in the medium term -> Positive effect on reducing air pollution"*)

U120: [*Env -> All*] **assess_decision** (*[U118, U119], U109, "Maintain emissions of air pollutants within tolerable levels(...)", [<1, 1>]*)

U121: [*Env -> All*] **end_turn** ( )

U122: [*Ini -> Fin*] **take_turn** ( )

U123: [*Fin -> All*] **assert** (*"Traffic light installation on each street costs at least €150K -> Traffic light installation on streets A and B costs approximately €300K"*)

U124: [*Fin -> All*] **assert** (*"Cost per parking space for surface car park averages €5K including land cost and construction, 50 spaces should be enough to accommodate demand, Costs of converting former street parking spaces to new lanes can be considered irrelevant -> Total cost of building car park is approximately €250K"*)

U125: [*Fin -> All*] **assert** (*"Intersection connects 3 streets -> Replacing 3-way intersection with roundabout costs at least €250K"*)

U126: [*Fin -> All*] **assert** (*"Economic incentives to retrofit vehicles with emission control equipment are fully subsidized by EU -> No cost to retrofit vehicles with emission control equipment"*)

U127: [*Fin -> All*] **assert** (*"Traffic light installation on streets A and B costs approximately €300K, Total cost of building car park is approximately €250K, Replacing 3-way intersection with roundabout costs at least €250K, No cost to retrofit vehicles with emission control equipment, Add margin of 10% for unforeseen costs -> Total cost of decision U109 is approximately €880K leading to positive balance of €120K"*)

U128: [*Fin -> All*] **assess_decision** (*[U123, U124, U125, U126, U127], U109, "Ensure decisions don't exceed budget", [<120000, 1>]*)

U129: [*Fin -> All*] **end_turn** ( )

Now we end the *Discussion* stage by having *Ini* tell the experts to move on to stage *Evaluation*.

U130: [*Ini -> All*] **evaluate** ( )

7.5 Gathering suggestions

Before computing the individual and joint preference orders over the three decisions proposed by the expert agents, we need to know what practical arguments are objectively accepted under skeptically preferred semantics, and then, for each of those arguments, we need to compute overall goal satisfaction in light of the criteria enforced by *Ini*. The latter is rather simple to do, because the only practical argument that deals with reasoning under uncertainty is U107. Because *Ini* enforced the maximin rule when dealing with uncertainty (all the way back in utterance U16), the result of the *satisf* function w.r.t. U107 is 0, as it amounts to picking the lesser value out of the two potential outcomes (0 and 1). For the remaining practical arguments, *satisf* yields the score of the sole outcome.

Concerning the calculation of objective acceptance, only epistemic arguments U81, U82 and U89 and practical argument U83 need to considered, as they are the only ones affected by relations of attack and preference. Figure 7.2 shows these relations. Practical argument U83 includes both U81 and U82 in its support set. Epistemic arguments U89 and U81 attack each other, and U89 unilaterally attacks U82. Additionally, all experts share the same preference relation, which dictates solely that U89 is preferred to U81. Therefore, by the rules of my argumentation framework, U89 defeats both U81 and U82, and by consequence also defeats the practical argument U83, as it includes at least one defeated epistemic argument in its support set. Considering these defeat relations, all epistemic and practical arguments except U81, U82 and U83 are objectively accepted under skeptically preferred semantics. Note that *Mob* understood this situation during the course of the dialogue and retracted the practical argument U83 immediately, so it wouldn't even be necessary to compute its acceptance at this stage were it not for the purpose of the example.

Now we can compute the decision preference orders. It is helpful to recall all the decisions proposed by the experts over the course of the debate and assign identifiers to them.

- **D1** (proposed in U55): *DN* (do nothing)

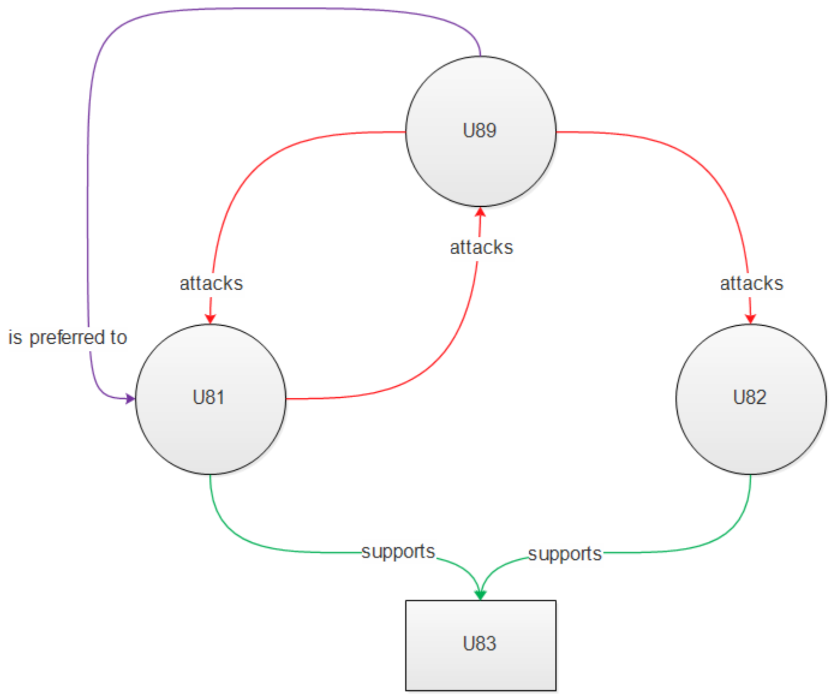

Figure 7.2: An illustration of the relations between arguments

- **D2** (proposed in U80): *"Replace crosswalks near school gates with pedestrian bridges"*
- **D3** (proposed in U109): *"Place traffic lights on crosswalks near school gates", "Raise awareness of school students about safety issues", "Replace intersection with roundabout", "Substitute parking places in street A eastbound and street B northbound with separate lanes for dropping people off and build park in land adjacent to school", "Enforce law to retrofit older vehicles with emission control equipment using economic incentives"*

Let's start off with *Mob*. His individual preference order is the easiest to compute. Remember that the first criterion is the number of personal goals that are minimally satisfied. *Mob's* personal goals are G_1 (reducing accidents), G_2 (minimizing junction delays) and G_3 (maximizing traffic flow). D1 satisfies none of *Mob's* goals, as can easily be seen by the various *assess_decision* locutions uttered by this expert. D2 satisfies only G_1. D3 satisfies all of *Mob's* goals (i.e. D3 is a satisfying decision from his perspective), as can be verified by utterances U111, U113 and U115. Therefore, it is clear that *Mob* prefers D3 to D2, and D2 to D1.

U131: [*Ini* -> *Mob*] **take_turn** ( )

U132: [*Mob* -> *Ini*] **prefer** (*U109, U80*)

U133: [*Mob* -> *Ini*] **prefer** (*U109, U55*)

U134: [*Mob* -> *Ini*] **prefer** (*U80, U55*)

U135: [*Mob* -> *All*] **end_turn** ( )

Env also happens to have the same preferences about decisions as *Mob*, but for different reasons. First of all, it is important to consider that all decisions minimally satisfy *Env's* personal goals (all decisions are considered satisfying from *Env's* perspective). To tiebreak these decisions, *Env* has to resort to the third criterion, i.e. preferring the decisions that better satisfy the most important goals (the second criterion only deals with unsatisfying decisions). We saw earlier that *Env* regards G_1 as more important than G_4 (his personal goal regarding air pollution); and G_4 as more important than G_5 (his personal goal regarding noise pollution). It happens that, according to the maximin rule, G_1 is equally well satisfied by both D2 and D3 (satisfaction degree 0), and unsatisfied by D1 (satisfaction degree -1). Therefore, both D2 and D3 are automatically preferred to D1. Additionally, D3 satisfies G_4 better than D2 (satisfaction degrees 1 and 0, respectively). Therefore, *Env* also prefers D3 to D2, and D2 to D1.

U136: [*Ini* -> *Env*] **take_turn** ( )

U137: [*Env* -> *Ini*] **prefer** (*U109, U80*)

U138: [*Env* -> *Ini*] **prefer** (*U109, U55*)

U139: [*Env* -> *Ini*] **prefer** (*U80, U55*)

U140: [*Env* -> *All*] **end_turn** ( )

Finally, all decisions satisfy *Fin's* sole personal goal, G_6 (monetary costs). *Fin* prefers D2 and D3 to D1 for the same reasons as *Env*. Unlike the other experts, *Fin* prefers D2 to D3, because G_6 is better satisfied by the former than the latter (satisfaction degrees 500000 and 120000, respectively), and *Fin* regards G_6 as the most important goal after G_1. Therefore, *Fin* prefers D2 to D3, and D3 to D1.

U141: [*Ini* -> *Fin*] **take_turn** ( )

U142: [*Fin* -> *Ini*] **prefer** (*U80, U109*)

U143: [*Fin* -> *Ini*] **prefer** (*U80, U55*)

U144: [*Fin* -> *Ini*] **prefer** (*U109, U55*)

U145: [*Fin* -> *All*] **end_turn** ( )

After the last utterance, the dialogue automatically moves to stage *Finalization*. Before closing it, we can compute joint preference order. Fortunately, this is rather easy to do, since all decisions rank differently according to the first criterion, i.e. the total number of goals that are minimally satisfied. D1 satisfies three goals: G_4 (air pollution), G_5 (noise pollution) and G_6 (monetary costs). D2 satisfies the same goals plus one - G_1 (reducing accidents). D3 is the only globally satisfying solution, as it satisfies all the six goals proposed by the agents. Therefore, D3 is jointly preferred to D2; and D2 is jointly preferred to D1. Now we can end the dialogue.

U146: [*Ini* -> *All*] **end_dialogue** ( )

7.6 Summary

In this chapter, we saw an example of how my theory can be used to solve problems of the real world. We started by setting up a scenario related to the domain of urban transportation networks, and saw how the inputs of a problem can be conveyed to a group of expert agents using simulation models and manually defined properties. We then requested discovery of issues in the network, and chose which ones to address more specifically.

The experts defined goals in accordance with their values and responsibilities, and proposed potential decisions to fulfill those goals, so as to attempt to solve the network's issues. After exchanging arguments and assessing the performance of each decision, the experts declared their individual preferences and we computed the joint decision preference order, before terminating the dialogue in orderly fashion.

Chapter 8

Conclusions

I conclude this thesis by providing a recap of the contributions of my theory and suggesting some possible directions for future work.

8.1 Contributions

Back in chapter 1, I described the aim of this thesis as *"to develop an argumentation model for deliberation in heterogeneous multiagent systems"*, and then I went on to propose five major goals through which I could achieve my aim. Let's analyze each of those goals in order.

The first goal was to *"determine the high-level key concepts that are common to any deliberation dialogue, independently of the context on which it might take place"*, which was addressed in chapter 3. The key concepts that I described were the *governing question* surrounding a problem, the *initial circumstances* or *properties* of the problem, the *base knowledge* of the expert agents, potential *actions* and *decisions* to solve the question, *consequences* of such actions, *goals* to be fulfilled, *metrics* to assess the fulfillment of those goals and *values* that serve as the underlying motivation of each expert. All of these concepts, I think, serve a very specific role within a deliberation dialogue. Only the concept of base knowledge is fuzzier, but it is needed to understand the origin and structure of the information possessed by each expert regarding actions, consequences, goals, metrics and other things. I believe that the elicitation of these structural concepts was greatly useful in guiding the rest of the work, as I predicted from the beginning. There is, however, a more important benefit: it clarifies the major aspects that developers have to deal with when implementing expert agents, and thus, I hope, becomes a reference in how to approach the high-level design of a rational deliberative agent.

The second goal was to *"establish an abstract argumentation framework that, based on the previously defined key concepts, allows deliberating agents to represent interactions between arguments and compute which ones are acceptable"*, which was addressed in chapter 4. This abstract argumentation framework drew on the foundational work of Dung and other authors and expanded upon it to provide a theory aimed at decision-making and deliberation dialogues, incorporating most of the aforementioned key concepts. I discussed the separate roles and importance of epistemic and practical arguments and the relationships between them, and made some considerations about the nature of epistemic argument preferences and attacks. Then I built a notion of practical arguments based on assessing the fulfillment of goals, allowing the incorporation of uncertainty and risk. Thus practical arguments effectively become a major factor in establishing a preference order between candidate decisions. Among other things, I also discussed the issue of computing defeat relations between arguments, and established rules for acceptability: only objectively accepted practical arguments under skeptically preferred semantics should be considered when establishing preference between decisions. I believe this framework provides an interesting contribution by itself, in that it shows how a theory of argumentation can incorporate many concepts relevant to decision-making and deliberation dialogues. Furthermore, it serves as the starting point for the development of the other two major components of my theory: the criteria for choosing decisions and the communication protocol.

My third goal was to *"establish a set of criteria that allows agents to determine a preference order for decisions, based on personal values and acceptable arguments in favor and against such decisions"*, and this I tackled in chapter 5. This relates to the concepts of decision calculi and social choice theory, which haven't yet been able to produce a consensual method. But by embracing this challenge, I demonstrated how one can compile a set of very reasonable criteria for rank-ordering potential decisions based on acceptable arguments and on the importance of different goals, from the perspective of individuals and groups, while incorporating techniques from decision-making under risk and uncertainty. I also provided algorithms for this purpose and an analysis of my criteria, showing that it exhibits some desirable properties like Pareto efficiency.

My fourth goal was to *"build a communication protocol that allows agents to conduct deliberation dialogues in accordance with my argumentation model"*, which I attended to in chapter 6. This was where the whole theory came together, combining the usefulness of an abstract argumentation framework with criteria for rank-ordering decisions and a protocol for multiagent interaction, drawing from multiple fields like decision-making, argumentation theory, deliberation dialogues, social choice theory and simulation. I devised a protocol divided in eight stages, each with a well-defined purpose: *Reunion*, *Input definition*, *Issue discovery*, *Question formulation*, *Exploration*, *Discussion*, *Evaluation* and *Finalization*. I also developed a series of locutions with carefully designed rules, which are explained in detail in appendix A. To demonstrate the suitability of my protocol, I provided an assessment of its merits in light of a set of desirable properties (or *desiderata*) suggested by influential authors, all of which were fulfilled. Many efforts were made to assure that my protocol strikes a good balance between generality of application and practical usefulness.

My fifth and final goal was to *"demonstrate the capabilities and potential of my theory with an example inspired by a complex problem from the real world"*, which I addressed in chapter 7. To achieve this, I presented a scenario loosely based on a real-life situation, in the context of planning improvements for an urban transportation network. Using this scenario, I created a fictional dialogue, carried out in accordance with the rules of my communication protocol, in which I attempted to briefly exhibit all the important features of my theory. Although the example addresses a very specific scenario, I believe it is adequate enough to demonstrate how my argumentation model can be used in practice to solve problems of the real world. And so I am convinced that all my goals have been achieved.

All things considered, I believe my thesis provides an interesting account of how argumentation theory can be used in combination with MAS to aid in the resolution of decision-making problems, which was the aim of this project. Furthermore, I think the multifaceted nature of the approach and the application of theory from different fields of knowledge brings additional insight into how we can improve the tools that help people make decisions.

8.2 Possibilities for future work

Using this thesis as a starting point, one can imagine a few directions for future work. The most obvious would be to implement a full-blown DSS employing this argumentation model. Though I focused on exemplifying the capabilities of my work in the context of urban transportation planning, I believe most practical decision-making problems are a natural fit. Given the detailed exposition of the features presented throughout these chapters, I think one could accommodate my theory to different circumstances with little or no modifications. One of my major concerns in designing my model was to try to put myself in the place of implementers, so as to provide a practical way to employ argumentation in real contexts without having to battle an army of intractable complications. With this principle in mind, it would be interesting to assess if there is anything that can be done to further improve the practical usefulness of my model, and perhaps develop a new AOSE methodology based on argumentative agents.

Regardless of the suitability of an abstract framework or a communication protocol, there is always one great source of difficulty in implementing argumentation-based MAS, stemming from the huge data modeling task that is required to encode expert knowledge. As I mentioned before, I think the key concepts described in this work can be most helpful when designing the logic of an expert agent; nevertheless, the representation of knowledge is usually a laborious task. This could be facilitated by incorporating recent approaches allowing human specialists to input their knowledge into the system using some method akin to building mind-maps, which could be automatically converted to ready-to-use agent knowledge at a second step of the process.

My next suggestion is somewhat related to the previous one. In this work, I have assumed that the user acts merely as a provider of input. It would be interesting to follow a more hybrid approach where humans could have an impact in the agents' discussion in real-time. A user could, for instance, point out inaccuracies in the experts' arguments, request the dialogue to branch in two under the assumption of different values of some property, etc. Foreseeing this scenario, I have already implemented some basic mechanisms to allow this interaction, such as putting forward incontestable arguments and attacks via the initiator agent, the ability to retract immediately to previous points in the dialogue, and the ability to accommodate extensions to the dialogue via the *say* locution.

Another interesting follow-up would be to investigate ways to demonstrate and extend the usefulness of simulation in combination with my model. I have already discussed some of the benefits of integrating simulation with argumentation to improve the quality of the system's suggestions: it provides access to richer models of the problems in discussion; it allows a greater degree of predictability w.r.t. potential decisions, thus favoring more tentative approaches; and it allows agents to integrate arguments with observations in the simulation environment in order to explore new opportunities and reinforce or discard each other's stances. This is already fully supported by my model, but I did not specify how expert agents would go about studying the effects of potential actions in the simulator at run-time. Obviously, this has no bearing on the argumentation model itself; besides, it is wholly dependent on the context and the simulation software. Nevertheless, it would be useful to develop software packages with this aim in mind: easily allowing an agent to make changes to a simulation model autonomously in order to test the effects of different actions. I have actually done some work in this area by implementing a traffic agent and an environmentalist agent that interact with the SUMO simulator at run-time, providing different inputs and extracting results, but given its technical difficulty, this work is still in early stages.

Finally, there are some minor aspects in my model that could be improved. I don't consider them essential and I doubt they would be useful in most contexts, but they might still be worth considering. As an example, the protocol could support alternative voting systems. Right now, only ranked-voting systems are allowed, but the user might be more interested in enforcing a plurality voting system when many expert agents are involved, or even only allowing the debate to finish once experts achieved a consensus. It would be difficult to achieve this in a generic way without violating some assumptions. Forced consensus, in particular, would be highly problematic in systems working under time constraints, and it would probably require fitting experts with the ability to make concessions (in this case, it might be preferable to engage in an embedded negotiation dialogue). Another example would be the somewhat restrictive nature of only considering objectively accepted arguments under skeptically preferred semantics. For instance, if two epistemic arguments attack each other, one is only considered valid if it is unanimously preferred to the other by all experts. While it is generally desirable that all experts agree about what is true (i.e. the facts of the problem), it might be sufficient, for instance, that an argument is agreed upon only be the absolute majority of the experts. This would implicate the need to enforce a well-established criterion before the discussion started, similar to what is done in the case of decision-making under risk or uncertainty.

To sum it up, this thesis allows several possibilities for future work, some in the form of extensions, others as complements to my theory. Nevertheless, I believe this work, on its own and in its current state, is already plenty useful in a variety of situations. It aimed to show how one can implement a multiagent system to support decision-makers, involving groups of heterogeneous agents that deliberate about decisions using argumentation. The result is a comprehensive model that achieves this purpose.

References

[AARR14] Tiago M. L. Azevedo, Paulo J. M. De Araújo, Rosaldo J. F. Rossetti, and Ana Paula C. Rocha. JADE, TraSMAPI and SUMO : A tool-chain for simulating traffic light control. In *Proceedings of 8th International Workshop on Agents in Traffic and Transportation*, pages 14–21, Paris, 2014.

[ABC07] Katie Atkinson and Trevor Bench-Capon. Practical reasoning as presumptive argumentation using action based alternating transition systems. *Artificial Intelligence*, 171:855–874, 2007.

[AC02] Leila Amgoud and Claudette Cayrol. A reasoning model based on the production of acceptable arguments. In *Annals of Mathematics and Artificial Intelligence*, volume 34, pages 197–215, 2002.

[Ale90] Robert Alexy. A theory of practical discourse. *The communicative ethics controversy*, pages 151–190, 1990.

[AP94] Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1):39–59, 1994.

[APP00] Leila Amgoud, Simon Parsons, and Laurent Perrussel. An argumentation framework based on contextual preferences. In *International Conference on Formal and Applied and Practical Reasoning (FAPR'00)*, pages 59–67, London, 2000.

[Arr63] Kenneth J. Arrow. *Social Choice and Individual Values*. Cowles Foundation Monographs Series. Yale University Press, 1963.

[BAT10] Jean-Rémi Bourguet, Leila Amgoud, and Rallou Thomopoulos. Towards a unified model of preference-based argumentation. *Foundations of Information and Knowledge Systems*, 5956:326–344, 2010.

[BB01] David Banister and Yossi Berechman. Transport investment and the promotion of economic growth. *Journal of Transport Geography*, 9(3):209–218, September 2001.

[BBG12] M. Barbati, G. Bruno, and A. Genovese. Applications of agent-based models for optimization problems: A literature review. *Expert Systems with Applications*, 39(5):6020–6028, April 2012.

[BC03] Trevor Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.

[BG09] Pietro Baroni and Massimiliano Giacomin. Semantics of abstract argument systems. In *Argumentation in artificial intelligence*, pages 25–44. Springer, 2009.

[BKM+14] Laura Bieker, Daniel Krajzewicz, Antoniopio Morra, Carlo Michelacci, and Fabio Cartolano. Traffic simulation for all: a real world traffic scenario from the city of Bologna. In *SUMO2014 Modeling Mobility with Open Data*, pages 19–26, Berlin, 2014. Deutsches Zentrum für Luft- und Raumfahrt (DLR).

[BPG+04] Paolo Bresciani, Anna Perini, Paolo Giorgini, Fausto Giunchiglia, and John Mylopoulos. Tropos: An Agent-Oriented Software Development Methodology. *Autonomous Agents and Multi-Agent Systems*, 8(3):203–236, May 2004.

[BTMA13] Jean-Rémi Bourguet, Rallou Thomopoulos, Marie Laure Mugnier, and Joël Abécassis. An artificial intelligence-based approach to deal with argumentation applied to food quality in a public health policy. *Expert Systems with Applications*, 40(11):4539–4546, September 2013.

[CDGV14] Federico Cerutti, Paul E. Dunne, Massimiliano Giacomin, and Mauro Vallati. Computing preferred extensions in abstract argumentation: A SAT-based approach. In *Lecture Notes in Computer Science*, volume 8306 LNAI, pages 176–193. Springer, 2014.

[CHCF02] John W. Coffey, Robert R. Hoffman, Alberto J. Cañas, and Kenneth M. Ford. A concept map-based knowledge modeling approach to expert knowledge sharing. *Proceedings of IKS*, pages 212–217, 2002.

[CSMC+00] Ulises Cortes, Miquel Sanchez-Marre, Luigi Ceccaroni, Ignasi R-Roda, and Manel Poch. Artificial Intelligence and Environmental Decision Support Systems. *Applied Intelligence*, 13:77–91, 2000.

[Dam03] Khanh Hoa Dam. Evaluating and Comparing Agent-Oriented Software Engineering Methodologies. Master's thesis, RMIT University, Australia, 2003.

[dB81] Jean-Charles de Borda. Memoire sur les elections au scrutin. *Histoire de l'Academie Royale de Science, Paris*, pages 657–664, 1781.

[DMT07] Phan Minh Dung, Paolo Mancarella, and Francesca Toni. Computing ideal sceptical argumentation. *Artificial Intelligence*, 171:642–674, 2007.

[DPH07] Paul Davidsson, Jan A. Persson, and Johan Holmgren. On the Integration of Agent-Based and Mathematical Optimization Techniques. In Ngoc Thanh Nguyen, Adam Grzech, Robert J. Howlett, and Lakhmi C. Jain, editors, *Agent and Multi-Agent Systems: Technologies and Applications*, volume 4496 of *Lecture Notes in Computer Science*, pages 1–10. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[Dun95] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77:321–357, 1995.

[Eur02] Council of the European Union European Parliament. Directive 2002/49/EC of the European Parliament and of the Council of 25 June 2002 relating to the assessment and management of environmental noise. *Official Journal of the European Union*, (L 189), 2002.

[Eur07] Council of the European Union European Parliament. REGULATION (EC) No 715/2007 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 20 June 2007 on type approval of motor vehicles with respect to emissions from light passenger and commercial vehicles (Euro 5 and Euro 6) and on access to vehicle repair and maintenance. *Official Journal of the European Union*, (L 171), 2007.

[Eur08] Council of the European Union European Parliament. Directive 2008/50/EC of the European Parliament and of the Council of 21 May 2008 on ambient air quality and cleaner air for Europe. *Official Journal of the European Union*, 51(L 152):1–44, 2008.

[Eur09] Council of the European Union European Parliament. REGULATION (EC) No 443 2009 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 23 April 2009 setting emission performance standards for new passenger cars as part of the Community's integrated approach to reduce CO2 emissions from light-duty vehicles. *Official Journal of the European Union*, (L 140), 2009.

[FERO08] Paulo A. F. Ferreira, Edgar F. Esteves, Rosaldo J. F. Rossetti, and Eugénio C. Oliveira. A cooperative simulation framework for traffic and transportation engineering. In *5th International Conference on Cooperative Design, Visualization, and Engineering*, pages 89–97, 2008.

[FIP02] FIPA (Foundation for Intelligent Physical Agents). FIPA ACL Message Structure Specification. Available at http://www.fipa.org/specs/fipa00061/SC00061G.html, 2002. Accessed: 2014-12-30.

[GHMV03] R. Girle, D. Hitchcock, P. McBurney, and B. Verheij. Decision Support for Practical Reasoning: a theoretical and computational perspective. In C. Reed and T. J. Norman, editors, *Argumentation Machines. New Frontiers in Argument and Computation*, pages 55–84. Kluwer Academic Publishers, Dordrecht, 2003.

[Got98] Jonathan Gottsegen. Using argumentation analysis to assess stakeholder interests in planning debates. *Computers, Environment and Urban Systems*, 22(4):365–379, July 1998.

[GPW07] Thomas F. Gordon, Henry Prakken, and Douglas Walton. The Carneades model of argument and burden of proof. *Artificial Intelligence*, 171:875–896, 2007.

[HCM10] *Highway Capacity Manual 2010, Volumes 1 - 4*. Transportation Research Board, Washington, DC, 2010.

[Hit91] David Hitchcock. Some principles of rational mutual inquiry. In *Proceedings of the Second International Conference on Argumentation*, pages 236–243. Amsterdam: SICSAT, 1991.

[HJBJ13] Stella Heras, Jaume Jordán, Vicente Botti, and Vicente Julián. Argue to agree: A case-based argumentation approach. *International Journal of Approximate Reasoning*, 54(1):82–108, January 2013.

[Hur51] Leonid Hurwicz. A class of criteria for decision-making under ignorance. *Cowles Commission Discussion Papers*, 356, 1951.

[INF] INFRAS. Handbook Emission Factors for Road Transport (HBEFA). Available at http://www.hbefa.net/. Accessed: 2014-12-03.

[Ins] Institute of Transportation Systems at the German Aerospace Center. Vehicular mobility trace of the city of Cologne, Germany. Available at http://kolntrace.project.citi-lab.fr/. Accessed: 2014-12-01.

[iTE10] iTETRIS Project Consortium. iTETRIS - An Integrated Wireless and Traffic Platform for Real-Time Road Traffic Management Solutions. Available at http://www.ict-itetris.eu/, 2010. Accessed: 2014-12-12.

[Jen00] Nicholas R. Jennings. On agent-based software engineering. *Artificial Intelligence*, 117(2):277–296, March 2000.

[JG09] Yan Jin and Mathieu Geslin. Argumentation-based negotiation for collaborative engineering design. *International Journal of Collaborative Engineering*, 1(1-2):125–151, 2009.

[Joh00] Ralph H. Johnson. *Manifest rationality: A pragmatic theory of argument*, chapter Context: Argumentation as a Cultural Practice. Routledge, 2000.

[JSW98] Nicholas R. Jennings, Katia Sycara, and Michael Wooldridge. A Roadmap of Agent Research and Development. *Autonomous Agents and Multi-Agent Systems*, 1(1):7–38, 1998.

[KEBB12] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent Development and Applications of SUMO – Simulation of Urban MObility. *International Journal on Advances in Systems and Measurements*, 5(3):128–138, 2012.

[KvdT08] Souhila Kaci and Leendert van der Torre. Preference-based argumentation: Arguments supporting multiple values. *International Journal of Approximate Reasoning*, 48:730–751, 2008.

[LL92] Magnus Ljungberg and Andrew Lucas. The OASIS air traffic management system. 1992.

[MF14] Jörg P. Müller and Klaus Fischer. Application Impact of Multi-agent Systems and Technologies: A Survey. In *Agent-Oriented Software Engineering: Reflections on Architectures, Methodologies, Languages, and Frameworks*, pages 27–53. Springer Berlin Heidelberg, 2014.

[MHP07] Peter McBurney, David Hitchcock, and Simon Parsons. The eightfold way of deliberation dialogue. *International Journal of Intelligent Systems*, 22:95–132, 2007.

[MM12] Maxime Morge and Paolo Mancarella. Arguing over Goals for Negotiation: Adopting an Assumption-Based Argumentation Decision Support System. *Group Decision and Negotiation*, 23(5):979–1012, December 2012.

[Mod49] Franco Modigliani. Papers and proceedings of the sixty-first annual meeting of the american economic association. *The American Economic Review*, 39(3):205, May 1949.

[MPW02] Peter McBurney, Simon Parsons, and Michael Wooldridge. Desiderata for agent argumentation protocols. *Proceedings of the first international joint conference on Autonomous agents and multiagent systems part 1 - AAMAS '02*, page 402, 2002.

[MVN53] Oskar Morgenstern and John Von Neumann. Theory of games and economic behavior. 1953.

[PCS10] David Pardoe, Doran Chakraborty, and Peter Stone. Tactex09: A champion bidding agent for ad auctions. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1*, pages 1273–1280. International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[Pol87] John L. Pollock. Defeasible Reasoning. *Cognitive science*, 11(4):481–518, 1987.

[Pra06] Henry Prakken. Combining sceptical epistemic reasoning with credulous practical reasoning. In *COMMA*, number 144, pages 311–322, 2006.

[PRG11] Lúcio S. Passos, Rosaldo J. F. Rossetti, and Joaquim Gabriel. An agent methodology for processes, the environment, and services. In *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pages 2124–2129, 2011.

[PS04] Anna Perini and Angelo Susi. Developing a decision support system for integrated production in agriculture. *Environmental Modelling & Software*, 19(9):821–829, September 2004.

[Rai02] Howard Raiffa. *Negotiation Analysis - the science and art of collaborative decision making*. Harvard University Press, Cambridge, MA, USA, 2002.

[RBB+02] Rosaldo J. F. Rossetti, Rafael H. Bordini, Ana L. C. Bazzan, Sergio Bampi, Ronghui Liu, and Dirck Van Vliet. Using BDI agents to improve driver modelling in a commuter scenario. *Transportation Research Part C: Emerging Technologies*, 10(5):373–398, 2002.

[RCS06] Jean-Paul Rodrigue, Claude Comtois, and Brian Slack. *The Geography of Transport Systems*, chapter Transport planning and policy. Routledge, 2006.

[RL05] Rosaldo J. F. Rossetti and Ronghui Liu. An agent-based approach to assess drivers' interaction with pre-trip information systems. In *Intelligent Transportation Systems*, volume 9, pages 1–10. Taylor & Francis, 2005.

[ROB07] Rosaldo J. F. Rossetti, Eugénio C. Oliveira, and Ana L. C. Bazzan. Towards a specification of a framework for sustainable transportation analysis. In *Workshop on Artificial Intelligence Applied to Sustainable Transportation Systems, 13th Portuguese Conference on Artificial Intelligence*, 2007.

[RR04] Chris Reed and Glenn Rowe. Araucaria: software for argument analysis, diagramming and representation. 2004.

[RS09] Iyad Rahwan and Guillermo R. Simari. *Argumentation in Artificial Intelligence*, volume 171, chapter Argumentation for Decision Making. Springer US, Boston, MA, July 2009.

[Sav51] Leonard Savage. The theory of statistical decision. *Journal of the American Statistical association*, 46(253):55–67, 1951.

[SB75] Edward H. Shortliffe and Bruce G. Buchanan. A model of inexact reasoning in medicine. *Mathematical biosciences*, 23(3):351–379, 1975.

[Sch02] Ralf Schleiffer. Intelligent agents in traffic and transportation. *Transportation Research Part C: Emerging Technologies*, 10(5-6):325–329, October 2002.

[Sch03] Markus Schulze. A new montonic and clone-independent single-winner election method. *Voting Matters*, 17(1):9–19, 2003.

[SG05] Linda Steg and Robert Gifford. Sustainable transportation and quality of life. *Journal of Transport Geography*, 13(1):59–69, March 2005.

[SM99] Andy Stirling and Sue Mayer. *Rethinking Risk: a pilot multicriteria mapping of a genetically modified crop in agricultural systems in the UK.* SPRU, 1999.

[SWH06] Weiming Shen, Lihui Wang, and Qi Hao. Agent-Based Distributed Manufacturing Process Planning and Scheduling: A State-of-the-Art Survey. *IEEE Transactions On Systems, Man and Cybernetics - Part C: Applications and Reviews*, 36(4):563–577, 2006.

[TARO10] Ivo J. P. M. Timoteo, Miguel R. Araujo, Rosaldo J. F. Rossetti, and Eugenio C. Oliveira. TraSMAPI: An API oriented towards Multi-Agent Systems real-time interaction with multiple Traffic Simulators. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 1183–1188. IEEE, September 2010.

[Tel] Telecom Italia SpA. Jade Site | Java Agent DEvelopment Framework. Available at http://jade.tilab.com/. Accessed: 2014-11-09.

[Tid87] T. Nicolaus Tideman. Independence of clones as a criterion for voting rules. *Social Choice and Welfare*, 4(3):185–206, 1987.

[TKGD13] Josep Tomàs Vergès, Daniel Krajzewicz, Dominik Grether, and Kai Dr. Nagel. Analysis and simulation of traffic management actions for traffic emission reduction. Master's thesis, Universitat Politècnica de Catalunya, 2013.

[Tou58] Stephen E. Toulmin. *The Uses of Argument*. Cambridge University Press, Cambridge, UK, 1958.

[Tra07] Transportation Planning Capacity Building Program. *The Transportation Planning Process: Key Issues*. Federal Highway Administration, Federal Transit Administration, 2007.

[U.S78] U.S. Environmental Protection Agency. Protective Noise Levels - Condensed Version Of EPA Levels Document. Technical report, 1978.

[VEG04] Frans Van Eemeren and Rob Grootendorst. *A systematic theory of argumentation: The pragma-dialectical approach*, volume 14. Cambridge University Press, 2004.

[Ver09] Bart Verheij. The toulmin argument model in artificial intelligence. In *Argumentation in artificial intelligence*, pages 219–238. Springer, 2009.

[Wal49] Abraham Wald. Statistical decision functions. *The Annals of Mathematical Statistics*, 20(2):pp. 165–205, 1949.

[Wal95] Douglas Walton. *Commitment in dialogue: Basic concepts of interpersonal reasoning*, chapter Dialogues: Types, Goals and Shifts. SUNY press, 1995.

[Wal09] Douglas Walton. Argumentation theory: A very short introduction. In *Argumentation in artificial intelligence*, pages 1–22. Springer, 2009.

[Wal13] Douglas Walton. *Argumentation schemes for presumptive reasoning*. Routledge, 2013.

[WJK00] Michael Wooldridge, Nicholas R. Jennings, and David Kinny. The Gaia Methodology for Agent-Oriented Analysis and Design. *Autonomous Agents and Multi-Agent Systems*, 3(3):285–312, 2000.

[Woo02] Michael Wooldridge. *An Introduction to MultiAgent Systems*, chapter Reaching Agreements. John Wiley & Sons, 1st edition, 2002.

[WTK01] Thomas Webler, Seth Tuler, and Rob Krueger. What is a good public participation process? Five perspectives from the public. *Environmental Management*, 27(3):435–450, 2001.

[WWV+09] Minhong Wang, Huaiqing Wang, Doug Vogel, Kuldeep Kumar, and Dickson K. W. Chiu. Agent-based negotiation and decision making for dynamic supply chain formation. *Engineering Applications of Artificial Intelligence*, 22(7):1046–1055, October 2009.

[Zad65] Lotfi A. Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

REFERENCES

Appendix A

Locution rules

This appendix provides the detailed rules of each of the 28 locutions defined in my communication protocol in chapter 6. It is split into two sections: one for auxiliary locutions and one for primary locutions. The order of the locutions within each section are the same as those presented in tables 6.1 and 6.2, which is roughly the order by which they would normally first appear in the dialogue. Each description contains seven fields: the name of the locution, permitted senders, permitted recipients, a list of its parameters, a description of its purpose (similar to those presented in tables 6.1 and 6.2 or extended if necessary), the preconditions that must be respected to permit its utterance and the effects of its utterance.

A.1 Auxiliary locutions

request_assistance

- **Sender**: The initiator agent.
- **Recipient**: An expert agent.
- **Parameters**: None.
- **Description**: Uttered by the initiator agent to request assistance from some expert agent. The first utterance in the dialogue is a locution of this kind. It is prevalent in the *Reunion* stage, but may be uttered at any moment in the dialogue.
- **Preconditions**: None.
- **Effects**: First instance of this locution triggers *Reunion* stage.

grant_assistance

- **Sender**: An expert agent.
- **Recipient**: The initiator agent.

- **Parameters**: None.
- **Description**: Signifies readiness of the sender to participate in the dialogue. Normally uttered by an expert agent in response to a *request_assistance* locution. Alternatively, the sender may utter this locution to express willingness to participate without previous request.
- **Preconditions**: None.
- **Effects**: The sending agent joins the dialogue. He may leave by normal dialogue termination (see locution *end_dialogue*), purposefully by uttering *withdraw_dialogue* while the dialogue is underway, or abruptly, potentially for accidental reasons (in which case the initiator may wait temporarily for a reconnection).

deny_assistance

- **Sender**: An expert agent.
- **Recipient**: The initiator agent.
- **Parameters**:
  1. (Optional) The reason for denying assistance.
- **Description**: Uttered by an expert agent in response to a *request_assistance* locution. Signifies that the expert agent declines to participate in the dialogue. Note that an expert agent may be assumed to deny assistance if he does not answer within a certain period after *request_assistance* if uttered.
- **Preconditions**: There must have been a prior utterance of *request_assistance* addressed to the sender of this locution.
- **Effects**: None.

withdraw_dialogue

- **Sender**: An expert agent.
- **Recipient**: The initiator agent.
- **Parameters**:
  1. (Optional) The reason for leaving the dialogue.
- **Description**: Uttered by an expert agent. Signifies that the expert agent no longer wishes to participate in the dialogue. This means that the agent abandons the dialogue midway. This withdrawal may be permanent or temporary, according to the reason provided by the agent. If no reason is provided, it is assumed to be permanent.
- **Preconditions**: Sender must have joined the dialogue (i.e. there must have been a valid previous utterance of *grant_assistance* by the sender of this locution). This precondition is obvious, so it will be omitted in subsequent locutions.

- **Effects**: If the withdrawal is permanent, the initiator agent might see fit to retract some of the expert's locutions (including, for instance, personal goals and practical arguments concerning such goals), depending on the stage of the dialogue. If temporary, the initiator agent might perform the same actions if the expert does not rejoin after a timeout.

request_sync

- **Sender**: An expert agent.
- **Recipient**: The initiator agent.
- **Parameters**: None.
- **Description**: Uttered by an expert agent to request an utterance of *provide_sync* by the initiator agent back to the expert. May be uttered at any moment during the dialogue.
- **Preconditions**: None.
- **Effects**: None.

provide_sync

- **Sender**: The initiator agent.
- **Recipient**: An expert agent.
- **Parameters**:
  1. A list of all locutions in the *CL*, ordered from the earliest to the most recent.
- **Description**: Uttered by the initiator agent in response to a *request_sync*. Together with *request_sync*, this locution allows expert agents to synchronize *CL* with the initiator agent.
- **Preconditions**: Normally, there must have been a prior utterance of *request_sync* by the recipient, but the initiator may see fit to synchronize *CL* without prior request (for instance, if an agent joins the dialogue midway).
- **Effects**: Recipient's *CL* becomes equal to initiator's *CL* at the moment of uttering.

retract

- **Sender**: Any agent (special rules apply to expert agents, see preconditions).
- **Recipients**: All other agents.
- **Parameters**:
  1. The (identifier of the) locution to be retracted.

- **Description**: Allows agents to retract a previous primary locution.[1]
- **Preconditions**: The locution to be retracted must be in the *CL*, i.e. it must be a valid primary locution. Furthermore, if the sender is an expert agent, the locution to be retracted must have been uttered by him. The initiator may utter *retract* at any moment, but he should not do it when an expert is taking his turn. Experts may utter *retract* within their turns during the *Exploration*, *Discussion* and *Evaluation* stages, but during the *Evaluation* stage they may not retract locutions uttered during previous stages (so as to not invalidate other expert's evaluations).
- **Effects**: If rules are followed, the locution is retracted from the *CL*. Note that retracting a locution might implicate other locutions become invalid (e.g. retracting a personal goal invalidates practical arguments concerning that goal), which must be retracted as well, either autonomously or by the initiator agent. Retracting locutions might trigger a reversal to a different stage.

retract_bulk

- **Sender**: The initiator agent.
- **Recipients**: All other agents.
- **Parameters**:
  1. The first locution to be retracted, by temporal order.
- **Description**: Allows the initiator agent to retract a locution and all locutions uttered afterwards. This is particularly useful to revert to a previous stage in the dialogue. It is equivalent to calling *retract* multiple times, starting at the specified locution. The rationale for *retract_bulk* is merely to promote efficient communication.
- **Preconditions**: The first locution to be retracted must be in the *CL*, i.e. it must be a valid primary locution.
- **Effects**: The specified locution and all locutions uttered afterwards by temporal order are removed from the *CL*. Care must be taken not to leave the *CL* in a state of rule violation (read description of the effects of *retract*), which is guaranteed not to happen if the dialogue reverts to the beginning or end of a stage.

dispute

- **Sender**: An expert agent.
- **Recipients**: All agents.

[1] Note that it isn't necessary to retract defeated epistemic arguments, as the framework's rules will invalidate them by default (this should actually be avoided when there is still a chance the argument may be defended against its attackers). Nevertheless, epistemic arguments which are obviously incorrect may be retracted for performance reasons.

- **Parameters**:
  1. A previously uttered primary locution of type *define_goal*, *propose_action* or *incompatible_actions*, sent by another expert agent, which is to be disputed.
- **Description**: Allows an expert agent to ask for justification of some primary locution of type *define_goal*, *propose_action* or *incompatible_actions* uttered previously by another expert agent. As explained before (see description of stage *Discussion*), this may be useful to dispute the legitimacy of personal goals, associations of goals to values, the technical feasibility of actions or declarations of incompatible actions, whose discussion falls outside the scope of the theory of argumentation presented in chapter 4.
- **Preconditions**: The locution to be disputed must be in the *CL*, i.e. it must be a valid primary locution of one of the aforementioned types, and it must have been uttered by a different expert agent. The dialogue must be either in stage *Exploration* or *Discussion*, which means that *define_question* must have been uttered already and *evaluate* must not have been uttered yet. Furthermore, the sending expert agent must be in his turn to speak.
- **Effects**: An embedded persuasion dialogue may take place, according to implementation-defined rules. Two outcomes are possible: either the sender of the disputed locution proves its validity and nothing else happens, or the disputed locution is proved invalid and subsequently retracted, either voluntarily by its sender, or forcefully by the initiator agent.

take_turn

- **Sender**: The initiator agent.
- **Recipient**: An expert agent.
- **Parameters**: None.
- **Description**: Instructs the expert agent to begin his turn, in accordance with the rules of the current stage (either *Issue discovery*, *Exploration*, *Discussion* or *Evaluation*).
- **Preconditions**: Dialogue must be in one of the aforementioned stages. Must not have existed a prior utterance of *take_turn* without corresponding *end_turn*.
- **Effects**: Recipient gains the right to speak until the next utterance of *end_turn*.

end_turn

- **Sender**: Any agent (special rules apply to expert agents, see preconditions).
- **Recipients**: All other agents.
- **Parameters**: None.

- **Description**: Ends the turn of the recipient of the last *take_turn* utterance, allowing other agents to take their turns.
- **Preconditions**: There must have been a prior utterance of *take_turn* with no associated *end_turn*. Expert agents may only end their own turns, i.e. they may only utter *end_turn* after a *take_turn* of which they were the recipient. The initiator agent may forcibly end any turn.
- **Effects**: Recipient of last *take_turn* loses right to speak. May trigger stages *Discussion* or *Finalization* if it ends the turn of the last expert agent in stages *Exploration* or *Evaluation* respectively.

discover_issues

- **Sender**: The initiator agent.
- **Recipients**: All other agents.
- **Parameters**: None.
- **Description**: Instructs the expert agents to look for issues in the state of the world, as defined by its properties (*define_property*) and/or simulation model files (*define_model*).
- **Preconditions**: Either *define_model* or one instance of *define_property* must have been previously uttered. *define_question* must not have been previously uttered, as it triggers the *Question formulation* stage.
- **Effects**: Triggers *Issue discovery* stage. Must be followed by a *take_turn* so that the first expert may start uttering *report_issue*.

report_issue

- **Sender**: An expert agent.
- **Recipient**: The initiator agent.
- **Parameters**:
  1. An estimate of the issue's severity in a scale between 0 and 1, with 1 being the most severe.
  2. A description of the issue.
- **Description**: Used by expert agents to report an issue to the initiator after a *discover_issues* request. Must be uttered once for each separate issue.
- **Preconditions**: *discover_issues* must have been previously uttered, and agent must be in his turn to speak (must have received a *take_turn* order, with no corresponding *end_turn* yet).
- **Effects**: None.

end_dialogue

- **Sender**: The initiator agent.
- **Recipients**: All other agents.
- **Parameters**: None.
- **Description**: Signifies normal termination of the dialogue, and cessation of the experts' responsibility to participate. Normally uttered at the *Finalization* stage, but this is not a requirement.
- **Preconditions**: None, other than that the dialogue has been started.
- **Effects**: The dialogue terminates.

A.2 Primary locutions

define_property

- **Sender**: The initiator agent.
- **Recipients**: All other agents.
- **Parameters**:
  1. An initial circumstance of the state of the world, as described in section 3.2.
- **Description**: The initiator agent utters this locution to define a property of the state of the world. This locution is predominant in the *Input definition* stage, but may be used before the dialogue moves on to the *Evaluation* stage.
- **Preconditions**: The dialogue must not have moved past the *Discussion* stage, i.e. the initiator agent must not have uttered *evaluate* (preferably, the dialogue should be in the *Input definition* stage so as to preserve normal flow). Additionally, *define_property* should not be uttered during the *Issue discovery* stage, otherwise experts may base their judgment on incomplete information.
- **Effects**: Triggers *Input definition* stage if there was no previous utterance of *define_property*, *define_model* or *enforce_criterion*.

define_model

- **Sender**: The initiator agent.
- **Recipients**: All other agents.
- **Parameters**:
  1. A list of URIs, giving access to the necessary resources to build a simulation model of the state of the world.

- **Description**: Allows the initiator agent to provide experts with access to the files necessary for simulating the state of the world. There may be at most one definition of a simulation model at any given time during the dialogue.
- **Preconditions**: Same as for *define_property*. Additionally, there must have been no prior utterance of *define_model*, i.e. if the initiator wants to redefine the model, the previous utterance of *define_model* must be retracted first.
- **Effects**: Triggers *Input definition* stage if there was no previous utterance of *define_property* or *enforce_criterion*.

enforce_criterion

- **Sender**: The initiator agent.
- **Recipients**: All other agents.
- **Parameters**:
  1. Whether the criterion to be enforced applies to reasoning under risk or uncertainty/ignorance. Must be either "RISK" or "UNCERTAINTY".
  2. The criterion to be enforced.
- **Description**: Allows the initiator agent to enforce the criterion to be used when computing overall goal satisfaction in decision-making under risk or uncertainty/ignorance, as defined in section 5.2. All agents must follow this criterion when computing decision preference order, unless it is later retracted.
- **Preconditions**: Same as for *define_property*. Additionally, there must have been no previous utterance of *enforce_criterion* for the same decision-making type (i.e. with the first argument being equal).
- **Effects**: Triggers *Input definition* stage if there was no previous utterance of *define_property*, *define_model* or *enforce_criterion*.

define_question

- **Sender**: The initiator agent.
- **Recipients**: All other agents.
- **Parameters**:
  1. The governing question, as described in section 3.2.
- **Description**: The initiator agent uses this locution to define the dialogue's governing question.
- **Preconditions**: There must have been no prior utterance of *define_question*.
- **Effects**: Triggers the *Question formulation* stage, which ends immediately, triggering the *Exploration* stage.

define_goal

- **Sender**: An expert agent.
- **Recipients**: All other agents.
- **Parameters**:
  1. The goal, as described in section 3.2.
  2. The value associated with the goal, as described in section 3.2.
- **Description**: Allows an expert agent to define a personal goal.
- **Preconditions**: The dialogue must be either in stage *Exploration* or *Discussion* (preferably *Exploration* to preserve normal dialogue flow), and the agent must be speaking in his turn. If the same goal has been previously defined, the locution is considered invalid.
- **Effects**: The expert agent commits to upholding the specified goal, i.e. for a sender called e, e declares that the new goal belongs to *Persgoals(e)*.

propose_action

- **Sender**: An expert agent.
- **Recipients**: All other agents.
- **Parameters**:
  1. An elementary action, as described in section 3.2.
- **Description**: Experts utter this locution to propose elementary actions, which may later be used for proposing potential decisions, i.e. combinations of zero, one or several elementary actions. This locution is predominant in stage *Exploration*, but may be used in stage *Discussion* as well.
- **Preconditions**: The dialogue must be either in stage *Exploration* or *Discussion* (preferably *Exploration* to preserve normal dialogue flow), which means that *define_question* must have been uttered already and evaluate must not have been uttered yet. Furthermore, the sending expert agent must be in his turn to speak. If the elementary action has been proposed before, the locution is considered invalid.
- **Effects**: The proposed elementary action may be used when proposing decisions, as long as it is not incompatible with any other elementary actions in that decision.

incompatible_actions

- **Sender**: An expert agent.
- **Recipients**: All other agents.

- **Parameters**:
  1. An elementary action that has been previously proposed, incompatible with the second.
  2. Another elementary action that has been previously proposed, incompatible with the first.
- **Description**: Experts utter this locution to point out incompatibility between two elementary actions. The combination of these locutions can be used to form a symmetric, not necessarily transitive relation, as explained in chapter 4. Notice that it isn't necessary to utter *incompatible_actions* again with reversed argument order because symmetry is always assumed.
- **Preconditions**: The dialogue must be either in stage *Exploration* or *Discussion* (preferably *Exploration* to preserve normal dialogue flow), which means that *define_question* must have been uttered already and *evaluate* must not have been uttered yet. Furthermore, the sending expert agent must be in his turn to speak. The two arguments (parameters) may not refer to the same elementary action. Both elementary actions must have been proposed before via *propose_action*.
- **Effects**: The two elementary actions may not be used simultaneously when proposing decisions. If a decision under such conditions has already been proposed, either that or this locution must eventually be retracted.

propose_decision

- **Sender**: An expert agent.
- **Recipients**: All other agents.
- **Parameters**:
  1. A decision to be considered, i.e. a (possibly empty) list of compatible elementary actions, as described in section 3.2.
- **Description**: Experts utter this locution during the *Discussion* stage to propose decisions they would like to see considered in the debate.
- **Preconditions**: The dialogue must be in the *Discussion* stage, i.e. *define_question* must have been uttered already, the expert must have already taken up his initial turn to propose actions and goals (corresponding to the *Exploration* stage) and *evaluate* must not have been uttered yet. Furthermore, the sending expert agent must be in his turn to speak. If, for any pair of elementary actions x, y contained in the list, *incompatible_actions*(x, y) or *incompatible_actions*(y, x) has been previously uttered, this locution is considered invalid. If there are no prior instances of *propose_decision*, the decision under proposal must be the *DN*, i.e. an empty set of elementary actions. The locution is also considered invalid if the same decision has been proposed before.

- **Effects**: The proposed decision may be used in *assess_decision* and *prefer* locutions.

assert

- **Sender**: Any agent (special rules apply).
- **Recipients**: All other agents.
- **Parameters**:
  1. An epistemic argument.
- **Description**: Agents utter this locution to put forward an epistemic argument, i.e. a defeasible piece of information from the agent's base knowledge, which might concern, for example, the consequences of an action, the effects of such consequences on certain goals, etc. (see section 3.2 and chapter 4).
- **Preconditions**: The dialogue must be in the *Discussion* stage, as explained for *propose_decision*. Obviously, if the epistemic argument has been asserted before, the locution is considered invalid. There are no rules *a priori* concerning the concrete structure of the argument.
- **Effects**: The epistemic argument may be employed in the support set of a practical argument, i.e. may be referred to subsequently when uttering an *assess_decision* locution, as well as *attack* and *prefer*. If the sender is the initiator agent, the argument may not be attacked and is automatically preferred to any arguments it attacks (it is considered an absolute fact).

attack

- **Sender**: Any agent.
- **Recipients**: All other agents.
- **Parameters**:
  1. An epistemic argument that has been previously asserted. This is the attacker.
  2. Another epistemic argument that has been previously asserted. This is the argument that is attacked.[2]
- **Description**: Agents utter this locution to express a relation of attack between two epistemic arguments, more concretely an attack from the first argument on the second one.
- **Preconditions**: The dialogue must be in the *Discussion* stage, as explained for *propose_decision*. As explained before, if the sender of the assert locution corresponding to the second argument is the initiator agent, this locution is considered invalid. For

[2]Read *attack*(x,y) as "epistemic argument x attacks epistemic argument y" (the reverse may or may not apply and therefore cannot be assumed unless explicitly stated via *attack*(y,x)).

simplicity's sake, it should be generally assumed that the sender of this locution is also the sender of the assert locution corresponding to the first argument (each agent would thus have the responsibility to define the attacks executed by his arguments), but this is not required.

- **Effects**: The attack may result in expert-specific defeat(s) depending on the epistemic argument preference relations (expressible with *prefer* locutions). This in turn may result in changes in the set of objectively accepted practical arguments under skeptically preferred semantics. See chapter 4 for details.

assess_decision

- **Sender**: An expert agent.
- **Recipients**: All other agents.
- **Parameters**:
  1. The practical argument's support set, i.e. a non-empty list of previously asserted epistemic arguments.
  2. A previously proposed decision.
  3. A previously defined goal.
  4. A set of outcomes, each of which is a pair containing a satisfaction level of the goal and probability of realization, see definition of practical arguments in chapter 4.
- **Description**: Expert agents may utter this locution to put forward a practical argument. The parameters of this locution are coincident with the definition of practical arguments in chapter 4.
- **Preconditions**: The dialogue must be in the *Discussion* stage, as explained for *propose_decision*. All referenced epistemic arguments, as well as the decision and goal must have been previously defined. No prior instance of *assess_decision* simultaneously referencing the same decision and goal must have been uttered. The sender of this locution must be the same as the sender of the *define_goal* locution corresponding to the referenced goal.
- **Effects**: If, by the end of the *Discussion* stage, the practical argument is objectively accepted under skeptically preferred semantics, it becomes a factor for consideration when computing decision preference orders.

prefer

- **Sender**: An expert agent.
- **Recipients**: Varies (see preconditions).

- **Parameters**:
  1. Either a previously uttered epistemic argument or decision.
  2. Either a previously uttered epistemic argument or decision (read *prefer*(x,y) as "x is at least as preferred as y").
- **Description**: Expert agents may utter this locution to express preference between epistemic arguments during the *Discussion* stage, or between decisions during the *Evaluation* stage.
- **Preconditions**: Either both arguments (parameters) reference epistemic arguments, or they both reference decisions. If they reference epistemic arguments, the dialogue must be in the *Discussion* stage, as explained for *propose_decision*; if they reference decisions, the dialogue must be in the *Evaluation* stage, i.e. *evaluate* must have been previously uttered and the sender must be speaking in his turn. As explained in section 6.2, the rules of the respective binary relations must be respected by the end of the agent's turn. Furthermore, if dealing with epistemic argument preferences, the recipients of the locution must be all agents, whereas if dealing with decision preferences, the recipient must be only the initiator agent (so as to permit blind voting), and as such the locution is not added to the *CL*.
- **Effects**: When dealing with epistemic arguments, effects may be similar to those described in *attack*. When dealing with decisions, may affect individual preference order, and consequently the joint preference order as well.

evaluate

- **Sender**: The initiator agent.
- **Recipients**: All other agents.
- **Parameters**: None.
- **Description**: Orders the expert agents to move from the *Discussion* stage to the *Evaluation* stage. This is the only primary locution that doesn't add any content to the dialogue, but is added to the *CL* anyway. It is necessary to establish progression: it makes expert agents understand that arguments should no longer be exchanged, and preferences over decisions should now be established. Normally, it makes sense to finish the *Discussion* stage once no experts have anything to add over two consecutive turns. Nevertheless, the initiator may choose to finish the stage prematurely, for instance if he has to function under time constraints.
- **Preconditions**: The dialogue must be in the *Discussion* stage, as explained for *propose_decision*, and no expert agent must be in his turn to speak.
- **Effects**: Triggers *Evaluation* stage.

say

- **Sender**: Any agent.
- **Recipients**: Any number of agents (subject to rules, see preconditions).
- **Parameters**:
  1. A boolean indicating whether the locution should be incorporated in the *CL*.
  2. A textual designation of the type of locution.
  3. Any number of additional parameters.
- **Description**: This locution allows agents to exchange any kind of information that falls outside the regular scope of the dialogue, for example in order to intertwine other types of dialogues (e.g. negotiation dialogues).
- **Preconditions**: If the sender does not wish to incorporate the locution in the *CL*, there are no preconditions *a priori*. Otherwise, the initiator agent must be included in the list of recipients, and the locution may only be uttered during the *Exploration*, *Discussion* or *Evaluation* stages (furthermore, if sent by an expert agent, he must be in his turn to speak; if sent by the initiator, no expert may be in his turn to speak). Other preconditions may be added by implementers, depending on the contents of the locution.
- **Effects**: Undefined.