TECHNICAL UNIVERSITY OF CATALONIA
Computer Science Department
Ph.D. Program: Artificial Intelligence

Ph.D. Thesis Dissertation

ProCLAIM: An Argument-Based Model for Deliberating Over Safety Critical Actions

Pancho Tolchinsky

Advisors: Prof. Ulises Cortés & Dr. Sanjay Modgil

May 2012

“Half of the people can be part right all of the time,
Some of the people can be all right part of the time,
But all of the people can’t be all right all of the time.
I think Abraham Lincoln said that. ”

Bob Dylan

Acknowledgments

First and foremost I must thank my wife for having been there throughout this very long journey. Secondly, I will again have to thank my wife for, fortunately, having done so much more than just being there.

To my friends Ulises and Sanjay, who also happen to be my advisors: Ulises, for better or worse, I do not believe I would have initiated this journey, let alone finished it, if it was not for you. You have made this masochistic experience as pleasant and enriching as possible. Sanjay, it was a very special gift to have met you along the way. I owe you more than you realise.

To my family, all included: to my father, who always tries to push us one degree higher; to my mother, for helping me, at the very beginning, when she showed me that papers can indeed have more red ink than black; to my brother, the useful doctor, for having mastered the patience of responding to my never-ending sequence of *what if* questions, with increasingly more ill patients queuing for more marginal organs.

To Katie and Peter, for being great research companions. Thank you for your insights, patience and friendship. To Matt, my coding fellow. To Antonio Loṕez Navidad and Francisco Caballero for not having harvested any of my vital organs throughout my long interviews. To Montse and the LEQUIA team for having dared to use *ProCLAIM* in the environmental context. To all the KEMLg team, in particular to Javi, for his generosity in sharing his time, knowledge, institutions and above all his latex files.

A very special thanks to all the members of the ASPIC project who also made this research possible, while creating a great atmosphere along the way, especially in the pubs. Thank you Sanjay, Henry, Peter, Matt, Trevor, Martin, Martin, Ioanna, David, Ivan, John, Ulises. . .

And to the rest, who perhaps I have forgotten to mention, feel free to consider yourselves thanked as well.

Abstract

In this Thesis we present an argument-based model – *ProCLAIM* – intended to provide a setting for heterogeneous agents to deliberate on whether a proposed action is safe. That is, whether or not a proposed action is expected to cause some undesirable side effect that will justify not to undertake the proposed action. This is particularly relevant in safety-critical environments where the consequences ensuing from an inappropriate action may be catastrophic.

For the practical realisation of the deliberations the model features a mediator agent with three main tasks: *1)* guide the participating agents in what their valid argumentation moves are at each stage of the deliberation; *2)* decide whether submitted arguments should be accepted on the basis of their relevance; and finally, *3)* evaluate the accepted arguments in order to provide an assessment on whether the proposed action should or should not be undertaken, where the argument evaluation is based on domain consented knowledge (*e.g* guidelines and regulations), evidence and the decision makers' expertise.

To motivate *ProCLAIM*'s practical value and generality the model is applied in two scenarios: human organ transplantation and industrial wastewater. In the former scenario, *ProCLAIM* is used to facilitate the deliberation between two medical doctors on whether an available organ for transplantation is or is not suitable for a particular potential recipient (*i.e.* whether it is safe to transplant the organ). In the later scenario, a number of agents deliberate on whether an industrial discharge is environmentally safe.

Contents

List of acronyms

AI	Artificial Intelligence
AEM	Argument Endorsement Manager
ASR	Argument Scheme Repository
CBR	Case-Based Reasoning
CBRc	Case-Based Reasoning component
COPD	Chronic Obstructive Pulmonary Disease
CQ	Critical Question
DA	Donor Agent
DCK	Donor Consented Knowledge
DOB	Degree Of Belief
DSS	Decision Support System
HA	Household Agent
HIV	Human Immunodeficiency Virus
HBV	Hepatitis B Virus
HCV	Hepatitis C Virus
IBIS	Issue Based Information System
InA	Industry Agent
ITA	Industrial Tank Agent
KB	Knowledge Base
MA	Mediator Agent
MAS	Multi-Agent System
MetA	Meteorologist Agent
NL	Natural Language
ONT	Organizacón Nacional de Transplantes
OCATT	Organizacó CATalana de Transplantaments
PA	Participant Agent
RA	Recipient Agent
RCA	River Consortium Agent
SA	Sewer Agent
sve	streptococcus viridans endocarditis
svi	streptococcus viridans infection
UCTx	Unidad Coordinadora de Transplantes
WTA	Wastewater Treatment Agent
WWTP	Wastewater Treatment Plant

Chapter 1

Introduction

Deciding whether a proposed action is safe is of particular value in safety-critical domains where the consequences ensuing from a *wrong* action may be catastrophic. Guidelines in such sensitive environments usually exist and are created in an attempt to minimise hazardous decisions, and thus direct decision makers on what to do. In some contexts however, decision makers are experts in the domain and may well propose actions that, although deviating from the guidelines, or common consented knowledge, are appropriate and thus, should be performed. Similarly, decision makers may prevent undertaking actions believed to be unsafe despite being compliant with guidelines. Furthermore, some scenarios require the participation of several agents, experts in different aspects of the problem, in deciding whether a proposed action should or should not be performed.

Despite the critical nature of the decisions to be taken, these deliberations among specialist agents are not always possible because of a lack of the appropriate support. We believe that Multi-Agent technology together with argumentation technics can make these deliberations less time consuming and enhance the possibility of a successful outcome, while accounting for the domain's consented knowledge.

The main objective of this work is to provide frameworks for the practical realisation of such deliberations, in order to;

- Make the deliberation effective and efficient by,
  - focusing on the relevant matters to be discussed, and
  - facilitating participation and exchange of arguments among agents, both human and artificial, where the latter may automate part or the totality of the process.
- Provide means to evaluate the outcome of the exchanged arguments on the basis of:
  - Their content: accounting for domain knowledge, *e.g.* guidelines.
  - Their use in previous deliberation: accounting for their associated empirical evidence.
  - Who endorse them: the role and/or reputation of the agents can bias the strength of the submitted arguments.

1.1 Deliberating Over Safety-Critical Actions

Safety critical actions such as transplanting an organ to a particular patient or to spill an industrial wastewater discharge require an extra obligation to ensure that no undesirable side effects will be caused, as these side effects may be the death of the patient or a severe impact on the effluvial ecosystem. To minimise harm, choice of safety-critical actions are usually governed by guidelines and regulations that direct decision makers on what to do. However, strict adherence to such domain consented knowledge may not always be convenient. For instance, in the transplant domain, strict adherence to conventional guidelines, regarding the criteria for donor and organ eligibility for transplantation, results in a progressive undersupply of available organs with the result of significantly extended waiting times and increased mortality of those on waiting lists [146]. Domains such as organ transplantation or wastewater management are highly complex and rapidly evolve, thus common consented knowledge cannot be expected to be always up-to date and account for all possible circumstances[1]. Hence, decision makers that are experts in these domains, should be able to deviate from guidelines, in so far as their decisions are well justified and supported by empirical evidence.

Furthermore, some safety-critical actions require the participation of several agents, experts in different aspects of the problem, for deciding whether or not their performance is safe. For example, an organ available for transplantation is better judged as suitable or not for a given recipient, if experts at the donor site jointly take a decision with the experts at the recipient site, which may be located in a different hospital [143]. Despite the added value of joint decision making among experts, this requirement cannot always be met. Without the appropriate support, the deliberation among experts on whether a proposed action is safe or not is time consuming and has no guarantee of a successful outcome. Thus, any decision support systems intended to assist experts in deciding whether a safety-critical action can be performed without causing severe undesirable side effects, must take into account that:

- Decisions on whether or not to perform a safety-critical action should be well justified.
- Guidelines and regulations are important, but strict adherence to them does not always warrant safety or determine the best decision.
- Empirical evidence plays an important role in safety-critical decision making.
- Decision makers may be experts in the domain. While their decision should be subjected to guidelines, they should be able to deviate from conventional guidelines in special circumstances.
- Several experts may be required to participate in the deliberation on whether or not the proposed action is safe, in which case one should take into account that:

[1] For example, Transplant organisations periodically publish the consented organ acceptability criteria. However, these criteria rapidly evolve because of the researchers' effort in extending them to reduce organ discards. Hence, the more advanced transplant professionals may deviate from consented criteria.

- Decision makers may be in disagreement about whether the action can safely be performed or not.
- Decision makers, especially human experts, may not be able to maintain long intensive deliberations. Furthermore, they may benefit from support in helping them consider all available information.
- Participant agents are expected to be heterogeneous. Some agent may be humans while others may be artificial. Furthermore, artificial agents may well be diverse in their implementation given that different agents may be implemented by different developers. It should be noted that by *heterogeneous* agents we do not imply the deliberation occurs in an open environment. Quite the opposite, we expect a highly regulated environment.
- Participants cannot be assumed to be skilled in argumentation.

In this work we propose an argumentation-based model *–ProCLAIM* – that provides a principled way to address the above introduced problem. That is, accounting for the above requirements, provide support to experts in deciding whether or not a safety-critical action can be performed without causing any undesirable side effect that would justify not undertaking the proposed action.

In the following section we provide a brief background to argumentation theory in Artificial Intelligence (AI). To latter, in §1.3, give a chapter by chapter description of the content, objectives and main contributions of this work. To finally, in §1.4 provide a summary of the main contributions of this Thesis.

1.2 Argumentation

The theory of argumentation is a rich, interdisciplinary area of research straddling philosophy, communication studies, linguistics, psychology and AI. Traditionally, the focus has been on *informal* studies of argumentation and its role in natural human reasoning and dialogue. More formal logical accounts of argumentation have recently been proposed by the AI community as a promising paradigm for modelling commonsense reasoning and communication between rational agents. Let us now consider an argument as a set of premises offered (informally) in support of a claim. For example:

Claim: The available `lung` should be transplanted to `John`.

Because:

> `John` is a patient whose quality of life may improve with the transplant of a suitable lung
>
> *and*
>
> `lung` is an available lung suitable for `John`
>
> *and*
>
> if an suitable lung is available for a patient then it should be transplanted.

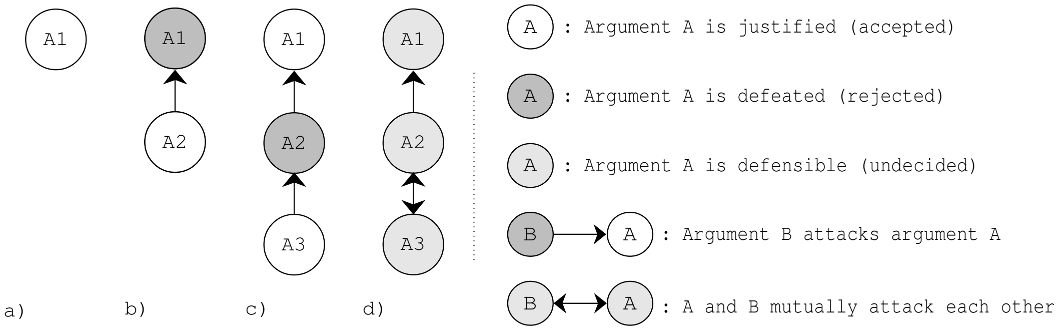

A1
A1
A2
A1
A2
A3
A1
A2
A3
a)
b)
c)
d)
A : Argument A is justified (accepted)
A : Argument A is defeated (rejected)
A : Argument A is defensible (undecided)
B A : Argument B attacks argument A
B A : A and B mutually attack each other

Figure 1.1: Graph of interacting arguments. According to the introduced example, $A1$ in an argument in favour of an organ transplant; $A2$ attacks $A1$ arguing the donor of that organ has Hepatitis C, which will cause the recipient to have Hepatitis C as well, which is undesirable. Now $A3$ defends $A1$ arguing that the potential recipient already has Hepatitis C.

Given this argument, which we can call $A1$, it seems reasonable to accept the transplant proposal. However, consider the counter-argument $A2$:

$A2$: The `lung` is not suitable for `John` *because* the donor of the lung has `Hepatitis C` which may be transmitted to `John` *and* `Hepatitis C` is a harmful for `John`

Furthermore, consider the counter-argument to $A2$:

$A3$: `Hepatitis C` is not harmful for `John` because `John` already has `Hepatitis C`.

Argumentation is the process whereby arguments are constructed and evaluated in light of their interactions with other arguments. So, in the above example, arguments $A1$, $A2$ and $A3$ have been constructed. $A3$ *attacks* $A2$ by contradicting a premise in $A2$, and $A2$ *attacks* $A1$ by contradicting a premise in $A1$. Given the arguments an their interaction, the winning arguments can then be evaluated (see Figure 1.1c). $A1$ is attacked by $A2$, but since $A2$ is itself attacked by $A3$, and the latter is not attacked, then $A1$ and $A3$ are the winning arguments. That is, the transplant can safely be performed despite the donor's Hepatitis C.

This example illustrates the modular nature of argumentation that most formal theories (models) of argumentation adopt [178]: 1) arguments are constructed in some underlying logic that manipulates statements about the world; 2) conflict-based interactions between arguments are defined; 3) given the network of interacting arguments, the winning arguments are evaluated.

The appeal of the argumentation paradigm resides in this intuitive modular characterisation that is akin to human modes of reasoning. Also, recent works in AI, and computer science community at large, have illustrated the potential for tractable implementations of logical models of argumentation [82, 78, 229], and the wide range of application of these implementations in software systems [50, 183, 185].

Furthermore, the inherently dialectical nature of argumentation models provide principled ways in which to structure exchange of, and reasoning about, arguments for proposals and/or statements between human and/or artificial agents.

Consider the above example where instead of a single agent engaging in its own internal argumentation to arrive at a conclusion, we now have two agents involved in a dialogue. Doctor Robert proposes argument $A1$, Doctor Dan argument $A2$, and then Doctor Robert counters with argument $A3$. Hence, this argument exchange may represent a dialogue between two doctors deliberating on whether or not an available lung is suitable for `John`.

Of course, dialogues introduce an added dimension, in the sense that realistic dialogues do not consist only in the exchange of arguments. Participants may challenge each others arguments, requesting, for example, for evidence that the donor does indeed has Hepatitis C. Also, participants may simply exchange contextual information that, although potentially relevant for the argumentation, may not be modelled as arguments. For example informing that the donor is a 65 years old male. Namely, different types of locutions may be exchanged and so, a protocol that defines the agents' interaction in the dialogue, what is usually term a *dialogue game*, has to be defined.

Yet another aspect in argumentation, is that it is not always trivial to compute which the winning arguments are, as in the network of arguments depicted in figure 1.1c. Suppose we define arguments $A2$ and $A3$ as mutually attacking each other, as depicted in figure 1.1d. Now we cannot conclude that $A1$ and $A3$ are the winning arguments. In a sense, this symmetrical attack between the two arguments indicates that it is a moot point as to whether the donor's Hepatitis C is or is not a contraindication for `John`. Typically, to resolve this impasse a preference relation between the mutually attacking arguments is assigned. So that for example, if $A2$ is preferred to $A3$, the donor's Hepatitis C is taken as a contraindication and thus, the transplant is deemed unsafe. However, if $A3$ is deemed preferred to $A2$, the transplant will be considered to be safe. In *ProCLAIM*, this preference relation will be derived from the domain consented knowledge, the confidence in the agents' knowledge about the domain and on empirical evidence. Thus, for example, whether the donor's Hepatitis C is a contraindication or not when the recipient already has Hepatitis C is decided on the basis of the medical guidelines, the reputation of the transplant professionals that endorse one or the other argument and on the basis of past recorded similar transplant cases.

Argumentation has shown to be particularly suitable for modelling commonsense reasoning, *i.e.* reasoning with conflicting, uncertainty and incomplete information [90, 135, 185]. These features are particularly relevant when reasoning over actions and their safety: as different experts may disagree with each other, as well as with the domain's guidelines, on whether or not an action is safe. As illustrated in the above example, Doctor Robert believes the transplant can safely be performed, while Doctor Dan believes the donor's Hepatitis C is a contraindication. The decision making is also pervaded with uncertainty, both on the effects (desirable and undesirable) of the proposed actions and on the circumstances in which the proposed action is performed (*e.g.* whether or not the donor and/or recipient do, in fact, have Hepatitis C). In the same way, some information, potentially relevant for the decision making may be missing, and even then, agents must still take a final decision.

Finally, argumentation allows for computing the winning arguments and thus propose a solution. The domain consented knowledge, the trust in the agents' knowledge about the domain and empirical evidence are taken into account in a natural way when assigning the preference relation between the mutually attacking arguments.

Hence, *in theory*, argumentation could readily be used to achieve the objectives of this Thesis. However, a number of pragmatic issues must be solved when proposing an argumentation-based deliberation for a, real-life, practical use.

Firstly, it cannot be assumed that participants are experts in argumentation, namely, that they are able to

- *efficiently* identify ALL the lines of reasoning relevant for the problem at had. That is, identify the relevant replies to previous submitted dialogue moves. And so, in particular, ignore those replies that, although legal (valid) from the viewpoint of the underlying model of argumentation, are irrelevant for the dialogue. Thus, for example, Doctor Dan should be able to efficiently identify all the relevant replies to Doctor Robert's submitted arguments $A1$ and $A3$. And ignore all the irrelevant lines of reasoning, *e.g.*, questioning whether `John` is indeed a donor.

- *efficiently* construct an argument that, *effectively*, embodies the desired reply. Suppose Doctor Dan disagrees with argument $A3$, and so, wants to submit an argument $A4$ in reply, such that it attacks $A3$. He should be able to construct argument $A4$, with minimum overhead, and such that the argument contains no more and no less than required for the deliberation.

Thus, while we do expect participants to be experts in the domain, in so far participants are assumed to also be skillful in argumentation, the burden of the success of the deliberation is placed on the agents' argumentation ability.

Secondly, participant agents are expected to be heterogeneous. In particular, some may be human while others artificial agents. Therefore, the adopted underlying model of argumentation, must not impose on the agents any specific way of reasoning and/or knowledge representation. This would otherwise compromise the agents' heterogeneity, and thus, the practical realisation of the deliberation.

And thirdly, if a preference relation is used to identify the winning arguments, there should be a clear account of where these preference relation comes from and what do they stand for.

These, more pragmatical issues, that allows for the practical realisation of the deliberation, are only weakly addressed in the current AI literature. And thus, they are the main questions this Thesis aims to address in presenting the *ProCLAIM* model.

In the following section we introduce *ProCLAIM* and how it addresses the above practical issues. We do so by providing a short abstract of the content of each chapter and its contributions.

1.3 Towards a Practical Realisation of Deliberations over Safety-Critical Actions

1.3.1 Chapter 2: Background for the Argumentation

In this chapter we introduce the argumentation theory the *ProCLAIM* model is based on, reviewing the most relevant works relative to our Thesis. We start by introducing the notion of argument schemes, typically used in the informal logic literature as a method for argument representation. Strongly influential in dialectical argumentation is the work of Walton [231], which proposes a classification of different types of arguments that embody stereotypical patterns of reasoning. Instantiations of argument schemes can be seen as providing a justification in favour of the conclusion of the argument. To each scheme are associated critical questions, so that for the presumptions to stand, satisfactory answers must be given to any such questions that are posed in the given situation. This work has motivated numerous works in dialectical argumentation [225, 194, 101] where particularly relevant for our current study is the work of Atkinson *et al.* [34], in which a dialogue game protocol for arguing over proposals for action is proposed. The argumentation evolves around a single argument scheme for practical reasoning:

> $AtkAS$:
> In the circumstances R
> we should perform action A
> to achieve new circumstances S
> which will realise some goal G
> which will promote some value V[2].

Where, if the variables R, A, S, G and V are instantiated appropriately, an argument in favour of action A is constructed. To this argument schemes are associated sixteen critical questions that address three different type of possible disagreements: *What is true* (*e.g.* –*Questioning the description of the current circumstances*–), *what is best* (*e.g.* –*Questioning whether the consequences can be realised by some alternative action*–) and *representational inconsistencies* (*e.g.* –*Questioning whether the desired features can be realised*–). Each such critical question can be instantiated by a scheme which in turn can be further questioned by its own critical questions. Following this idea a dialogue game protocol for multi-agent argument for proposals over action is proposed in [32]. As we discuss in §6 we take Atkinson *et al.* work on schemes and critical question as a starting point for formulating *ProCLAIM*'s protocol-based exchange of arguments tailored for reasoning over whether a proposed action can a safely be performed.

Whilst argument schemes provide us with a means to generate arguments and question them, we also need a mechanism that will enable us to automatically evaluate the arguments and challenges generated in order to determine the ones that are acceptable. For this we make use of Dung's abstract argumentation theory [80] which has proven to be an influential approach to conflict resolution and non-monotonic reasoning over the past decade. Dung's argumentation framework consists of a tuple $< \mathbf{A}, \mathbf{R} >$, where $\mathbf{A}$ is a set of arguments and $\mathbf{R}$ is a binary relation of attack between arguments. That is, if $A1, A2 \in \mathbf{A}$ and $(A2, A1) \in R$ then, $A1$ and $A2$ are arguments and $A2$ attacks $A1$. Broadly speaking, an argument $A1$ in $\mathbf{A}$ is defeated (*i.e.* it is not a winning argument) if there is an argument $A2$ in $\mathbf{A}$ such that $A2$ is a winning argument and $(A2, A1) \in \mathbf{R}$. An argument $A1$ is an justified (winning) argument if there is no other winning argument $A2$, $A2 \in \mathbf{A}$, such that $(A2, A1) \in \mathbf{R}$.

[2] In this sense values represent the social interests promoted through achieving the goal. Thus they are qualitative, as opposed to quantitative, measures of the desirability of a goal.

Thus, for example, in a framework $< \{A1, A2, A3\}, \{(A2, A1), (A3, A1)\} >$, the winning arguments are $A1$ and $A3$, and $A2$ is defeated argument (defeated by $A3$). Argument $A3$ defends $A1$ from the attack of argument $A2$, see figure 1.1c.

Dung's Argumentation Framework has become a standard for modelling both agents' (monolithic) reasoning [58, 119, 181] and for evaluating arguments constructed and submitted during the course of a dialogue [27, 46, 176]. Indeed, as stated above, arguments $A1$, $A2$ and $A3$ may have well been the result of a single agent's reasoning to conclude that the transplant is safe, despite the donor having Hepatitis C. Or, it may be the result of a deliberation dialogue between two (or even three) agents. Be as it may, the submitted arguments conform an interacting graph of arguments, to which, a deliberating agent can still add further arguments that attack or defend the justification given for or against the proposed action's safety. In our case, each such argument instantiates a scheme, the critical questions then help identify relevant reasoning lines by which to attack any previously submitted argument.

We continue by discussing Walton and Krabbe's influential classification of human dialogues [232]. This work helps clarifying the context and purpose of the dialogue which is helpful for its formalisation. Thus, we analyse the characteristics of the different argumentation dialogues types, such as persuasion, negotiation or deliberation, to conclude that for the kind of collaborative decision making *ProCLAIM* is intended for, *deliberation dialogues* are most appropriate. Deliberation dialogues involve participants deciding what action should be undertaken in a given situation. Typically participants in such a dialogue believe that they share a responsibility for deciding what to do, which provides a collaborative context to the dialogue. Thus, as opposed to, for instance, persuasion, characterised by a proponent and an opponent, participant in a deliberation dialogue do not have any initial commitment with respect to the basic subject matter of the dialogue. In particular, in this way the dialogue focuses on *what to do* rather than who is right and who is wrong. Negotiation, on the other hand, involves dividing some scarce resource and thus, it defines an adversarial situation which of course deviates from the purpose of *ProCLAIM*'s dialogues.

In order to implement these dialogues we must define protocols for the agents' interactions, usually termed dialogue games. Dialogue games are interactions between two or more participants who *move* by uttering locutions, according to certain rules. A dialogue game may be specified by listing the legal locutions, together with the rules which govern the utterance of these locutions, the opening and termination of dialogues, and the rules for manipulation of any dialogical commitments incurred by the participants during a dialogue [153]. Numerous dialogue games have been proposed for argumentation dialogues, most of which model persuasion [38, 52, 176] and negotiation dialogues [207, 156, 184] and very few actually address deliberation dialogues [150, 132]. The most elaborated dialogue game intended for deliberation is the framework for ideal Deliberation Dialogues proposed by McBurney, Hitchcock and Parsons in [150]. In that framework, deliberation dialogues may proceed through eight successive stages: *Open*, *Inform*, *Propose*, *Consider*, *Revise*, *Recommend*, *Confirm* and *Close Stages*. The goals of participants in these dialogues change according to the stage of the dialogue. It is for this reason that stages are marked explicitly, so as to better enable participants to know what is expected of them at each stage. In this framework, there are some constraints on the order of the stages, and some stages may be repeated, in an iterative fashion, before the dialogue is completed. As we discuss in §5, This approach is taken as a starting point for defining *ProCLAIM*'s dialogue game where a great difference between the two approaches is that while McBurney's *et al.* dialogue game is intended for an open (ideal) deliberation about what to do, *ProCLAIM*'s dialogue game is specialised for deliberating over the safety of a proposed action where great part of the interaction is shaped (and constraint) by the argument schemes the participant agents can instantiate at each stage of the deliberation.

We conclude this chapter by discussing the fruitful relation between argumentation and human-computer interaction and reviewing other works proposing argumentation-based systems intended for safety critical domains and discuss their relation with the *ProCLAIM* model. In this last section we contextualise the proposed model within the tradition of qualitative approaches to address decision making under uncertainty [169], as an alternative to classical decision theory based on probabilistic inference [227, 187], specially in practical situations where it is difficult or simply not possible to quantify uncertainty. In the following chapter (in §3) we motivate the need for this qualitative approach for addressing our two case studies. In particular, we motivate the requirement for a deliberation of the kind provided by *ProCLAIM*.

1.3.2 Chapter 3: Two Safety Critical Scenarios

To motivate the need for a deliberation among experts on whether a safety-critical action is safe or not, two safety critical scenarios are introduced. The first and main case study relates to human organ transplantation while the second, complementary case study, is related to industrial wastewater discharges in riverbasins, and thus of environmental impact.

One of the main problems faced in the transplant domain is the scarcity of human organs for transplantation. Despite this scarcity, an important number of organs, available for transplantation, are being discarded as being deemed not suitable for that purpose. Currently, deciding whether to offer an available organ is based exclusively on the assessment of experts at the donor site, ignoring the fact that *1)* expert may disagree on the viability of an organ, and *2)* organs are rarely viable or non-viable *per se*; rather, assessment of viability should depend on both the donor and potential recipient characteristics as well as on the courses of action to be undertaken during transplantation. Thus, we propose the use of the *ProCLAIM* model to coordinate joint deliberation on organs' viability between transplant professionals on the donor site with professionals at the recipient site. The purpose of this deliberation is to, while ensure that the proposed transplant is safe, prevent discarding organs due to the application of conventional medical guidelines that are too strict. Therefore, these deliberations have the potential of increasing the availability of human organs for transplantation; which, in turn, may help reduce the increasing disparity between demand for and supply of organs.

As to the environmental scenario, the focus is on industrial wastewater discharge in riverbasins. Industries that produce and discharge wastewater, often meet critical situations in which an efficient and informed decision must be taken in order to minimise the negative environmental impact on the river. For an environmentally safe discharge, decision makers must account for the different actors involved in the fluvial ecosystems, such as the sewer systems, storing tanks, wastewater treatment plants, while also accounting for external factors, such as meteorological conditions or other concurrent industrial discharges. In this scenario, *ProCLAIM* is proposed to coordinate joint deliberations among the various and heterogeneous agents, in order to decide whether an industrial wastewater discharge environmentally safe or not.

The transplant scenario is taken to illustrate the different aspects of the *ProCLAIM* model throughout this Thesis, in §7.2 we present a running example, in §10.1 we present an implemented version of this scenario and later in §11.1 we discuss our experience and lessons learned in its development. The environmental scenario was primarily developed by environmental engineers. We used this experience to test the applicability of *ProCLAIM* in novel scenarios, and in particular to learn which are the main challenges for developing a *ProCLAIM*-based application by developers who are not familiar with Argumentation. We describe this experience and the lessons learned in the development of the environmental scenario in §11.2.

The work in this chapter is address in the following publications [16, 12, 11, 15, 9, 10, 17, 13] addressing the transplant scenario and [3, 6, 2, 1] which address the environmental scenario.

It is worth noting two important outcomes of these two scenarios. For the transplant scenario we developed an software application within the FP6-European Project Argumentation Service Platform with Integrated Components (ASPIC)[3] that was used as the main large scale demonstrator of the argumentation technologies developed within the this project (this is further discussed in §10.1). An important outcome of the environmental scenario is Aulina's PhD Thesis in Environmental Engineering [39] that gravitates around the results obtained with *ProCLAIM*.

1.3.3 Chapter 4: *ProCLAIM*'s Framework

In this chapter we provide a first overview of *ProCLAIM*, identifying the model's components and their role within the deliberation. The main component featured by *ProCLAIM* is a Mediator Agent (MA), that defines a centralised medium through which participants interact. The MA's task is to ensure the success of the deliberation's progress, as well as, when the deliberation concludes, proposes a solution on the basis of the exchanged arguments and accounting for the domain consented knowledge, the empirical evidence that support the arguments and the confidence in the participants' knowledge in the domain. To do so, the MA references four knowledge resources defined by *ProCLAIM*, which are briefly described below:

Argument Scheme Repository: Encodes the scenario specific argument schemes and their associated critical questions. Referenced by the MA in order to direct the participant agents in the submission and exchange of arguments.

[3] http://www.argumentation.org

Domain Consented Knowledge: Encodes the scenario's domain consented knowledge. Referenced by the MA in order to account for the domain's guidelines, regulations or any knowledge that has been commonly agreed upon.

Case-Based Reasoning component: Stores past cases and the arguments given to justify the final decision. Referenced by the MA in order to evaluate the arguments on an evidential basis.

Argument Endorsement Management: This component manages the confidence in the participants' knowledge of the domain. It is referenced by the mediator agent in order to bias the strength of the arguments on the basis of the agents that endorse them.

This chapter concludes with an illustration of how *ProCLAIM* is instantiated in the transplant and environmental scenarios. The main contribution introduced in this chapter is *ProCLAIM*'s principled way to address the intended problem. Which involves: Using the scenario specific patterns of reasoning as a means to elicit all the relevant factors for deciding whether a proposed action can safely be performed. This results in a tree of interacting arguments. The given arguments are evaluated on the basis of their content, on who endorses them, and on the basis of their associated evidential support. The result of this argumentative process is a comprehensible assessment over whether it is safe to perform the proposed action. If there is enough available knowledge, this assessment can be regarded as a justification to why the proposed action can or cannot be safely performed, we continue this discussion in §1.3.7.

The work in this chapter is address in the following publications [18, 5, 11, 16]

1.3.4 Chapter 5: *ProCLAIM*'s Deliberation Dialogue

In this chapter we describe the interaction protocol that governs *ProCLAIM*'s dialogue. Since the agents interaction is centralised by the Mediator Argent we split the dialogue game in two levels. One in which we define a shallow inform-reply interaction between a participant agent and the MA, which we call the proxy dialogue game. This interaction level prevents participant agent from receiving any disruptive message, including arguments that are too weak to be accounted for. On a second level we define the locutions intended for the actual deliberation. We organise this deliberation dialogue game into three stages: *Open*, *Deliberation* and *Resolutions*. Tthe *Deliberation Stage* can in turn be subdivided in three layers: *Argumentation*, *Context* and *Endorsement* layers. The moves in these three layers may happen in parallel and the moves at one layer may have an effect on another layer. We define yet another interaction layer, called *Information layer* in which the participant agents can request the MA for updates regarding the ongoing deliberation:

- *Open Stage*: In which participant enter the dialogue, provide and receive the basic required information for their participation.

- *Deliberation Stage*:

  - *Context Layer*: At this layer participant agents assert (resp. propose) or retract facts (resp. actions) they believe to be potentially relevant for the deliberation.
  - *Argumentation Layer*: This is where the actual argumentation takes place. At this level we only define the locution (the wrappers) for submitting arguments and challenges. The actual argumentation, defined in terms of a circuit of schemes and critical questions, is described in §6.
  - *Endorsement Layer*: In this layer, the participant agents indicate which are the arguments they endorse. These endorsements are taken into account when assigning a preference relation between mutually attacking arguments.

- *Resolution Stage*: At this stage agents seek to conclude the deliberation, in which case the MA proposes a solution on the basis of the submitted arguments, accounting for the provided contextual information and the arguments' endorsements.

- *Inform Layer*: At this layer participant agents can recover any lost information relevant for the deliberation.

This dialogue game is thus similar to McBurney *et al.*'s, deliberation framework [150], in that it is defined in terms of stages, however, we take these stages as conceptual, in the sense that they only help organise the game's definition, participant agents need not know which locution corresponds to each stage and locutions of different stages may be submitted in parallel defining a very liberal dialogue game. What makes the deliberation highly focused occurs at the *Argumentation Layer* as we discuss in §6.

An important contribution of this dialogue game is that it decouples the resolution of *what is the case*, which happens at the *Context Stage*, and the deliberation over the actions' safety, which occurs at the *Argumentation Layer*. Firstly, this gives priority to the main question:–*Is the action safe in current circumstances?*-so that, for example, questioning the current circumstances (whether a stated fact holds) is licensed only if this challenges the action's safety. Secondly, as we show in §6, it allows addressing, in a relatively simple fashion, problems such as incomplete or uncertain information within the scope defined by *ProCLAIM*.

The work in this chapter is address in [16].

1.3.5 Chapter 6: *ProCLAIM*'s Argumentation Layer

In this chapter we focus on the deliberation's *Argumentation Layer*, that is, here we define what types of arguments participants can exchange and following what rules. This is defined in terms of a structured set (a *circuit*) of schemes and critical questions which conform a *protocol-based exchange of arguments*. These schemes are specialised for deliberating over whether or not current circumstances are such that the proposed action can be performed without causing any side effect. Any argumentation move is directed towards elucidating this question. In fact, the deliberation can be regarded as an argumentative process for eliciting knowledge from the participants, as opposed to defining a strategic dialogue in which a better choice of arguments may better serve the agents' individual goals.

We start by introducing the structure of *ProCLAIM*'s arguments, which is based on Atkinson *et al.*'s scheme for action proposal $AtkAS$ (introduced above in §1.3.1). We narrow the argumentation possibilities by assuming that a proposed action is in default circumstances desirable. The circuit of schemes and critical questions is designed then to direct participants to consider, incrementally, additional factors (facts or complementary courses of actions) that have an impact on the action's safety, that is, that make the main proposed action safe or unsafe by being or not being the case (performed or not, in the case of complementary actions). The result of this process is an organised tree of arguments that highlights all the relevant factors for the decision making indicating why these factors are relevant.

The main contribution in this chapter resides in the specialised nature of the schemes and critical questions for reasoning over action safety. While most approaches focus on generality (*e.g.* [34, 101]), with the underlying assumption of an ideal discussion scenario, here we take an almost opposite approach motivated by *ProCLAIM*'s decision making context. We explicitly limit the argumentation to the immediate relevant aspects of the problem at hand in order to produce highly focused deliberations. The intended outcome is a set of schemes that not only help agents construct structured arguments (as traditionally scheme do), but schemes which instantiation involves no overhead for laymen in argumentation and can thus be used as a mechanism for eliciting knowledge from domain experts without being disruptive. To deliver such schemes, however, the schemes and critical questions defined in this chapter need to be further specialised for the target application, this we describe in §7.

An additional contribution of the schemes defined in this chapter is that the relevant factors for the decision making are explicitly singled out. This facilitate the task of comparing past deliberations, following the basic intuition that if two deliberations share all their relevant factors they are then similar. This is of course central to the implementation of *ProCLAIM*'s Case-Based Reasoning component defined in §9.

The work in this chapter is address in the following publications [16, 5, 17].

1.3.6 Chapter 7: Argumentation into Practice

In this chapter we intend to provide a clear view of how the *ProCLAIM*'s Mediator Agent can guide the participant agents (human or artificial) at each stage of the deliberation on what can be argued and how, and so, providing a setting for an efficient and effective deliberation among heterogeneous agents. Central to this task is *ProCLAIM*'s Argument Scheme Repository. For this reason we start by describing a step by step procedure to construct the application-specific schemes and critical questions that conform the ASR. The constructed schemes are defined both in a formal representation useful for computational uses (*e.g.* for artificial agents), and in a natural language representation useful for human agents. Later, by means of an example, we show how the MA can guide both human and artificial agents in their argumentation by referencing the ASR and following the dialogue game introduced in §5. As part of the mechanisms provisioned by *ProCLAIM* to promote a highly focused deliberation, is the validation process. In this process the MA checks whether each submitted argument by the participant agents is not only a well formed argument but whether it is a relevant argument for the deliberation. To this end, the MA references the three knowledge resources: Domain Consented Knowledge (DCK), Case-Based Reasoning component (CBRc) and the Argument Endorsement Manager (AEM). Thus, a submitted argument may be rejected by the MA if it is not validated by the DCK and the CBRc has no record of this argument being successful in the past. Nonetheless, the argument may exceptionally be accepted if the AEM deems the submitter agent to be sufficiently reliable regarding the particular aspects addressed by the submitted argument. In this way the MA prevents disrupting the deliberation with spurious arguments, or arguments that are too weak to be considered in the particular circumstances.

The main contribution in this chapter is that we show *ProCLAIM*'s argumentation into practice. In other words, we show how a deliberation dialogues between agents (human or software) over the safety critical actions can be fully or partially automatised, in a manner which is structured and orderly, and which elicits all the information needed to make such decisions jointly and rationally, even when this information is possessed only by some of the participating agents.

The work in this chapter is address in the following publications [16, 11, 5, 17].

1.3.7 Chapter 8: *ProCLAIM*'s Argument Evaluation

As soon as participant agents inform they have no more moves to submit (or a timeout is triggered) the MA proceeds to evaluate the submitted arguments organised as a tree of interacting arguments. This evaluation involves three main steps for the MA: *1)* Reference the Domain Consented Knowledge (DCK) and the Case-Baser Reasoning component (CBRc) to check whether there are additional arguments that must be submitted as being deemed relevant by the DCK and/or by the CBRc. *2)* Reference the DCK, CBRc and the Argument Endorsement Manager (AEM) in order to assign a preference relation between the arguments that mutually attack each other. Finally, *3)* apply Dung's theory in order to identify the winning arguments so as to propose a solution which will help decide whether the main proposed action should or should not be undertaken.

However, as we discuss in this chapter, a clear, one dimensional solution, cannot always be provided. Not only participant agents may disagree on what to do, but also the different knowledge resource may disagree on which arguments are preferred. Furthermore, uncertainty and incomplete information need also to be taken into account. Thus, the task of the MA includes, organising the available information derived from domain consented knowledge, previous similar deliberations, and expert opinions' in order to deliver a comprehensible solution proposal, which in its best case is a justification to why the proposed action can safely be performed or why it is not. However, when there is not enough knowledge to do so, the proposed solution identifies what pieces of knowledge are missing and indicates what are the risks involved in performing the proposed action under these circumstances.

The main contribution of this chapter is the integration of the diverse and complementary assessments derived from *ProCLAIM*'s knowledge resources into a comprehensible assessment over the action safety. This integrated assessment provides decision makers a better description of the circumstances in which the action is to be performed and what are the risks involved in so doing.

The work in this chapter is address in the following publications [16, 18].

1.3.8 Chapter 9: *ProCLAIM*'s Case-Based Reasoning Component (under development)

In this chapter we will introduce *ProCLAIM*'s Case-Based Reasoning component (CBRc), which allows to evaluate a target problem on the basis of previously resolved deliberations. The CBRc allows to:

- **Resolve symmetrical attacks into asymmetrical attacks**. Thus, it assigns a preference relation between mutually attacking arguments on evidential basis. For example, suppose we take the Hepatitis C example, with the target deliberation being of the form $< \{A1, A2, A3\}, \{(A2, A1), (A3, A2), (A2, A3)\} >$, and so, with arguments $A2$ and $A3$ mutually attacking each other. Suppose that the similar deliberations, retrieved by the CBRc, of the form: $< \{A1, A2, A3\}, \{(A2, A1), (A3, A2)\} >$ where $A3$ attacks asymmetrically $A2$, significantly outnumbers those of the form $< \{A1, A2, A3\}, \{(A2, A1), (A2, A3)\} >$ where $A2$ asymmetrically attacks $A3$. In that case, the CBRc would propose to prefer $A3$ over $A2$. Thus, indicating that there is evidence that the donor's Hepatitis C is not a contraindication within these circumstances, and so, the transplant can safely be performed.

- **Submit additional arguments**, deemed relevant in previous similar deliberation, though not accounted for by the participants in the current deliberation. Returning to the Hepatitis C example, suppose that the retrieved similar deliberations are now of the form $< \{A1, A2, A3, A4\}, \{(A2, A1), (A3, A2), (A4, A3)\} >$ with $A4$ being an argument that indicates that, because the donor and recipient had both Hepatitis C, the recipient resulted[4] having a severe infection. Thus, the CBRc would submit the additional argument $A4$, that attacks argument $A3$, to the current deliberation. Hence, the CBRc would indicate that there is evidence that, in current circumstances, the transplant is not safe.

As mentioned above, the way the model's schemes and critical questions are defined they allow for the reuse of the knowledge encoded in the deliberations. In particular, the CBRc's memory is organised on the basis of the Argument Scheme Repository. Thus, the scenario specific schemes are essential for the implementation of the CBRc reasoning cycle.

The main contribution in this chapter the definition of the CBRc by which we shown how past stored deliberation can be reused in order to resolve similar target problems. Three fundamental aspects in CBRc' definition are: *1)* the trees of arguments produced in a *ProCLAIM* deliberation encode most of what is relevant for the decision making; *2)* the use of the specialised arguments schemes of the ASR to organise the CBRc's memory, which allows to efficiently retrieve the relevant cases while makes it easy to retain the new resolved target cases; and *3)* a retained tree of arguments embeds the outcome of the proposed action, if performed.

The work in this chapter is address in the following publications [18, 16, 11].

[4] Note that, since the retrieved deliberations were already resolved, the retrieved arguments are no longer presumptive but are explanatory in nature.

1.3.9 Chapter 10: Software Implementations

In this chapter we present four pieces of software that address different aspects of *ProCLAIM*. The focus in this chapter is to discuss the model's applicability, contributions and limitations on the basis of our experience in its implementation.

The first and more important result, regarding *ProCLAIM*'s implementation, is the large scale demonstrator for the medical transplant scenario. This demonstrator was developed within the FP6-European Project Argumentation Service Platform with Integrated Components (ASPIC)[5], and was recognised by the project reviewers as one of the main contributions and achievements of the ASPIC project and, in general, as one of the most sophisticated argument-based systems. The main goals the ASPIC project were 1) to develop a solid theoretical foundations for the Argumentation Theory in AI; 2) based on the theoretical work, develop practical-software components that embody standards for the argumentation-based technology (*inference*, *decision-making*, *dialogue* and *learning*); and *3)* In order to test these components develop two large scale demonstrators. The main large scale demonstrator was based on the transplant scenario introduce in this Thesis. This medical demonstrator was implementation using the *ProCLAIM* model as well as two ASPIC components: a Dung-based argumentation engine and a dialogue manager. The former component was used by the artificial agents in their reasoning and the latter component was used by the mediator agent in order to implement the transplant scenario's interaction protocol.

In this chapter we review three additional prototypes: a web-based argument scheme repository browser that allows domain experts to interact with the encoded knowledge in the Argument Scheme Repository; a web-based implementation that, on the basis of the model's abstract dialogue game, facilitates the construction of scenario specific Argument Scheme Repositories. The third prototype is an implementation of *ProCLAIM*'s Case-Based Reasoning component introduced in §9.

The main contributions introduced in this chapter are the actual implementation of the above mentioned tools and applications.

The work in this chapter is address in the following publications [18, 11, 16].

1.3.10 Chapter 11: Two Case Studies

In this chapter we discuss *ProCLAIM* in light of our experience in defining and developing the transplant and the environmental scenarios. Through out this chapter we describe the issues raised by the two applications considered with respect to argumentation and how these have informed our theoretical formalisations. Because these two case studies had very distinct development processes, they provide us with a broad perspective on the proposed model's applicability, contributions and limitations. The problems addressed in the transplant scenario have initially inspired the conceptualisation of *ProCLAIM* and has guided us throughout its development. This scenario not only showed to be a fruitful case study for the development of the ASPIC's large scale demonstrator (see §1.3.9) but it also shown to contribute to the transplant domain with publications in transplant conferences [9, 15, 10] elaborated in collaboration with transplant professionals of the Hospital de la Santa Creu

[5] http://www.argumentation.org

i Sant Pau[6]. In the environmental scenario, on the other hand, *ProCLAIM* was used as the initial inspiration for the definition of a new scenario within the context of wastewater management. The development of this scenario was undertaken by researchers from a Laboratory of Chemical and Environmental Engineering[7] and thus by researcher who are not familiar with Argumentation and with limited experience in Computer Science in general. This scenario was very useful for identifying the scope of applicability and for the definitions of procedures to facilitate the construction of the Argument Schemes Repository which constitutes the backbone of our proposed model.

The three main contributions in this chapter are *1)* identify the limitation in current argumentation in addressing the two applications; *2)* the assessment of the *ProCLAIM* model in two case studies; and *2)* the actual developed the two scenarios.

The work in this chapter is address in the following publications [16, 12, 11, 17, 15, 9, 10, 4] addressing the transplant scenario and [3, 6, 2, 1] which address the environmental scenario. Our work in the field has inspired other lines of research in the same field (*e.g.* [40, 93]).

1.3.11 Chapter 12: Conclusions

In this chapter we give our overall conclusions of this Thesis identifying the main contributions and limitations of our work and plans for future work.

1.4 Objectives and Contributions

The main objective of this study is to provide frameworks and associated guidance for the implementation of environments in which agents can efficiently and effectively deliberate over the safety of proposed actions, accounting for the domain consented knowledge, for the evidence associated to previously resolved similar deliberations and to expert opinion.

- To make deliberations efficient and effective:
  - With the use of argument schemes together with the MA's guiding task, the deliberation can be both highly focused on the problem at hand while exhaustive in addressing all the relevant lines of reasoning. This last aspect being of great value in safety-critical domains where there is an extra obligation to explore all the possible lines of reasoning.
  - The use of specialised argument schemes together with the effective visualisation characteristic of argumentation-based interfaces, facilitate domain experts participation in the deliberation. Graphical representation of the exchanged arguments provides an intuitive understanding of the stage and content of the deliberation. Natural language representation allows exploring each of the arguments' content, while programming language representations allow artificial

[6] http://www.santpau.cat/default.asp?Language=en

[7] http://lequia.udg.es/eng/index.htm, from the University of Girona

  agents to automate part or the totality of the deliberation as well as provide assistance to human agents on their participation, for example, in the argument construction.
  - With the use of scenario specific argument schemes, scheme instantiations (*i.e.* argument construction) becomes a transparent process with no overhead for the participant agents. Therefore, the defined deliberation can be regarded as an argumentative process for eliciting knowledge from the participants, as opposed to defining a strategic dialogue in which a better choice of arguments may better serve the agents' individual goal
  - By referencing the available knowledge resources, the MA prevents spurious arguments to disrupt the course of the deliberation.

- Account for domain consented knowledge, evidence and expert opinion. This is addressed in two complementary ways:
  - Experts submit arguments that address the factors they believe to be relevant for the decision making, while the MA submit additional arguments not accounted for by the participants, but deemed relevant from the viewpoint of the guidelines or that were used in previous similar deliberation and so may be relevant in the current deliberation. Thus, in this way the deliberation accounts for all the available knowledge for the decision making.
  - The resulting tree of interacting arguments is then evaluated by the MA assigning a preference relation between the mutually attacking arguments and so propose a solution on the basis of guidelines, evidence and expert opinions.

- The guidance for the construction of specialised argument schemes enables developers who are not familiar with Argumentation implement *ProCLAIM* in novel domains.

Among the many contributions introduced in this Thesis, there are five main contributions that can be highlighted above the rest:

- The principled way *ProCLAIM* provides to address the collaborative decision making regarding whether a safety-critical action can be performed.

- The protocol-based exchanged of arguments based on the structured set of argument schemes and critical questions specialised for arguing over whether a proposed action can safely be performed, which, in turn facilitates the further specialised, scenario-specific, argument schemes and their associated critical questions.

- The model's Case-Based Reasoning component that allows to reuse the knowledge encoded in *ProCLAIM* deliberations.

- A large scale demonstrator developed for the FP6-European Project ASPIC that serves as an advanced proof of concept of *ProCLAIM*.

- The application of *ProCLAIM* in an alternative scenario, developed primarily by environmental engineers not familiar with Argumentation.

Chapter 2

Background for the Argumentation

There has been a recent rise in interest in the use of argumentation techniques to handle reasoning in automated systems [190, 49, 185]. This has been driven by the desire to handle defeasible reasoning in contexts where proof cannot be used, *e.g.,* in domains where information is incomplete, uncertain or implicit. An argument is less tightly specified than a proof, as arguments offer open-ended defeasibility whereby new information can be brought to an issue and the reasoning can proceed non-monotonically[1]. Also, arguments provide us with a concept of subjectivity when topics of debate involve reasoning about choices that may rationally be acceptable to one agent but not to another. Proofs play an essential role in matters where information is certain and complete, but most real-world situations do not have such clear cut information available and it is here where argument plays its important role. In such situations argumentation can be used to provide tentative conclusions for or against a claim in the absence of further information to the contrary of the claim.

To model the process of argumentation in automated reasoning systems, requires methods that enable our reasoning agents to both generate arguments and proposals about what to believe and what to do, and methods to enable reasoning agents to assess the relative worth of the arguments pertinent to a particular debate, *i.e.,* which arguments are the most convincing and why. Here we set out the main mechanisms that we will use for these purposes: argument schemes, which we describe in §2.1, and argumentation frameworks, which we describe in §2.2. Also important for agents' interaction is to account for context and purpose of such interaction, this we address in §2.3 where we introduce Walton and Krabbe's classification of human dialogues [232] where we focuss on deliberation dialogues. In order to implement these dialogues one must define protocols for the agents' interactions termed dialogue games, these we introduce in §2.4. In §2.5 we briefly discuss one very important feature for argumentation, its suitability for interfacing human-computer interaction. This is due to its suitability for modelling commonsense reasoning, for its suitability for visualisation and because computer produced reasoning can easily be presented in natural language, via argument schemes. Finally, in §2.6, we review other works proposing argumentation-based systems intended for safety critical domains and discuss their relation

[1] That is, allowing for the possibility that additional arguments might reduce the list of justified claims that can be derived from the overall given arguments.

with the *ProCLAIM* model. The background theory and review of related works addressing Case-Based Reasoning and Argumentation is covered in §9.

2.1 Argument Schemes and Critical Questions

Argumentation schemes were introduced in the informal logic literature as a method for argument representation. They have appeared in different guises (*e.g.* [180], [220], [171], [129], or [106]) and were generally intended for use in the analysis of natural language argument (*e.g.* [63, 104, 161]). One early example of the use of argument schemes is Toulmin's Argument Schema [220]. This schema allowed for more expressive arguments to be asserted than had been afforded through previous schemes for argument that were based upon logical proofs consisting of the traditional premises and conclusion. It did this through the incorporation of additional elements to describe the different roles that premises can play in an argument (see Figure 2.1). However, what the schema does not provide is a systematic way to attack all elements of an argument, in the same way that the argumentation schemes and critical questions approach does. Advocated by Walton [231][2], also enables a variety of different kinds of argument, each with its own defining characteristics, to be put forward in the course of a debate. Such schemes represent stereotypical patterns of reasoning whereby the scheme contains premises that presumptively licence a conclusion. The presumptions need to stand in the context in which the argument is deployed, so they can be challenged by posing the appropriate critical questions associated with the scheme. In order for the presumptions to stand, satisfactory answers must be given to any such questions that are posed in the given situation. Walton introduced twenty-five schemes with their associated Critical Questions (CQs), amongst which we can site as more relevant to our work, *i.e.* reasoning about safety critical actions, the two schemes for practical reasoning: the necessary condition scheme:

$W1$:
G is a goal for agent a
Doing action A is necessary for agent a to carry out goal G
Therefore agent a ought to do action A.

and the sufficient condition scheme:

$W2$:
G is a goal for agent a
Doing action A is sufficient for agent a to carry out goal G
Therefore agent a ought to do action A.

[2] While Hastings [111] was the first to associate a set of critical questions to an argument scheme as part of its definition, in 1963, It is Walton's more mature work [231] presented in 1996, that had a great influence in AI and Argumentation.

To which Walton associates the following four critical questions:

CQ1 Are there alternative ways of realising goal G?

CQ2 Is it possible to do action A?

CQ3 Does agent a have goals other than G which should be taken into account?

CQ4 Are there other consequences of doing action A which should be taken into account?

Walton's proposed argumentation schemes, their extensions and variations (*e.g.* in [125] Katzav and Reed introduced over one hundred schemes) have been applied in a wide variety of works in AI, though most notably in the legal domain, where schemes that capture arguments from *testimonial evidence*, from *expert opinion* or from *analogy*, among others, are used to model legal cases [230]. In this context, Gordon *et al.*'s Carneades model [101] and System[3] are particulary related to our work in that it assumes the centrality of a repository of argument schemes for the construction of arguments in a dialogue and that it aims to use argument schemes as a means to bridge diverse forms of reasoning (both of these uses of argument schemes and CQs were also advanced by Reed and Walton in [194]). In its more practical facet, Carneades can be regarded as a software library for building argumentation tools intended to assist users in construct a wide variety of arguments (through argument scheme instantiation) in order to improve their ability to protect their interests in dialogues, specially in the legal domain. Carneades's intended generality, with its particular concerned in accurately formalising the complex reasoning present in the legal domain, deviates however from our main concern, which is to focus the deliberation on the very specific problems relevant for deciding, in our case, the safety of a proposed action. *ProCLAIM* does not intend to address all potentially relevant arguments regarding the safety of an action, in the sense that, some lines of reasoning my be only appropriate in a brain storming session with no immediate time constrain (*e.g.* arguing over the validity of different research studies). In short, while Carneades is concern with generality and expressiveness in a very wide sense, we are more focused on affectivity, that is, in a real time deliberation among domain experts.

Though specifically designed to be embedded in a dialogical process, Carneades does not define in itself any dialogue protocol. Instead it is thought to be a reusable component providing services generally needed when an argumentation protocol is specified. Furthermore, Carneades does not make any explicit use of the dialectical nature of schemes and CQs [225]. The schemes and CQ effectively map out the *relevant* space of argumentation, in the sense that for any argument they identify the valid attacking arguments from amongst those that are logically possible. In particular, they provide a natural basis for structuring argumentation based dialogue protocols. The idea is that the associated CQs to a scheme are themselves defined in terms of schemes, which in turn have associated CQs, and thus forming a circuit of schemes and CQs. Hence, for every submitted argument, its associated CQs identify the argumentation moves that can be mad in its reply. We have taken this approach in [7, 17] in an early formalisation of *ProCLAIM*. In this work we have defined a repository of schemes specialised for the transplant scenario, we introduce this scenario

[3] http://carneades.berlios.de/

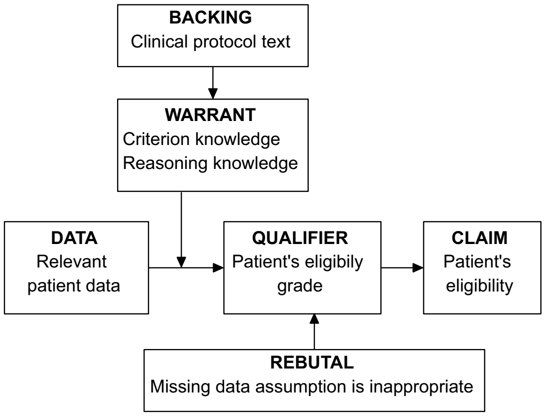


Figure 2.1: Toulmin Scheme with its six components: **Claim** (an assertion or conclusion put forward for general acceptance); **Data** (particular facts about a situation on which a claim is made); **Warrant** (knowledge that justifies the leap from data to a claim); **Backing** (general body of information or experience that validate the warrant); **Qualifier** (phrase that shows the confidence with which the claim is supported to be true); and **Rebuttal** (anomaly or exception for which the claim would not be true). Example extracted from [204] where Toulmin schemes are proposed to present medical explanations.

in §3.1. These specialised schemes were constructed in a rather *ad hoc* fashion; hence, while we made good progress in this scenario [11, 12, 18], it was difficult to implement *ProCLAIM* (mainly, to construct a repository of schemes) in an alternative scenarios, such as the wastewater scenario [39, 6, 2, 1], which we introduce in §3.2. In order to address this issue we took Atkinson's *et. al.* work [32, 34] for reasoning over action proposal as a reference point. In this work Atkinson's *et. al.* refine the two schemes for practical reasoning (sufficient and necessary condition), defined by Walton, proposing a single scheme for action proposal with sixteen associated critical questions:

Atk:
In the circumstances R
we should perform action A
to achieve new circumstances S
which will realise some goal G
which will promote some value V

Where a goal is some particular subset of S that the action is intended to realised in order to promote the desired value, and where the values represent the social interests promoted through achieving the goal [47, 43]. Thus values are qualitative, as opposed to quantitative, measures of the desirability of a goal (*e.g.* a goal may promote values such as friendship, wealth or safety). The sixteen critical questions address three different type of possible disagreements: *What is true* (*e.g. –Questioning the description of the current circumstances–*), *what is best* (*e.g. –Questioning whether the consequences can be realised by some alternative action–*) and *representational inconsistencies* (*e.g. –Questioning whether the desired features can be realised–*). In [34] this scheme, along with its sixteen CQs are used to define a persuasion dialogue game for reasoning about action proposal.

Argument scheme Atk refines Walton's schemes $W1$ and $W2$ in three significant ways: firstly, as pointed out in [34] the necessary condition scheme $W1$ is a special case of the sufficient condition scheme $W2$, in which CQ1 (*–Are there alternative ways of realising goal G?–*) is answered negatively and thus, they can be taken as a single scheme. Secondly, in Walton's schemes the notion of a goal is ambiguous, potentially referring indifferently to any direct result of the action, the consequence of those results and the reasons why those consequences are desired. In order to clarify this point, Walton's goal G is divided in three parts: a specific set of circumstances S that the action will cause, the goal G that these new circumstances will realise and finally, the reason for this goal to be desired is captured by the value V it promotes. Thirdly, the CQs are significantly extended and refined so as to better capture all the possible lines of reasoning for arguing about what to do.

It is this work that we take as our starting point for defining *ProCLAIM*'s argument schemes. However, as we discus in §6, we made extensive modifications to scheme Atk and its CQs in order to accommodate the requirements of the deliberations for which *ProCLAIM* is intended. These modifications address two main issues: *1) ProCLAIM* deliberations are not intended for arguing about *what to do* in its broader sense, but only about the safety of a proposed action and *2) ProCLAIM*'s deliberations are a means to elicit knowledge from the participants, they are not intended for strategic dialogues in which a better choice of arguments may better serve the agents' individual goals. In addition, as discussed in §9, *ProCLAIM*'s schemes and CQ formalisation is also intended to facilitate the Case-Based Reasoning component's task, which is to reuse previous similar deliberations in order to evaluate arguments on an evidential basis.

Note that, while scheme Atk helps structuring arguments, its instantiation is non-trivial for laymen. That is, it is not obvious for participants in the deliberation, not experts in argumentation, which values should be given to R, A, S, G and V in order to effectively capture what they may have to add to the deliberation. In our experience, discussed in §11, both Atkinson *et al.*'s proposed scheme [34] and Walton's schemes [231] render difficult to use in real time deliberation as they involved too much overhead for the participant (transplant professionals). In our experience, while the participants in the task understood the basic principles of the presented schemes, they initially guessed which could be possible instantiation for the proposed schemes and soon after they were disengaged with the task at hand. The experience with the specialised schemes was more in line with our expectations, where participants were only focused on the content of the deliberation and were not particularly interested in the actual structure of the given scheme, which were presented as a template in natural language using the participant's particular jargon. For this reason we initially proposed in the transplant scenario the use of scenario specific schemes [17]. However, while the results obtained with this formalisation were very positive [11], as discussed above, the *ad-hoc* approach made it difficult to apply in novel scenarios, in particular for developers who are not familiar with Argumentation Theory, as it was the case in the environmental scenario (which we discussed in §11.2). Through the realisation of the value of scenario-specifc schemes, jointly with the need for procedures to facilitate the production of these specialised schemes that has led us to develop the current formalisation of *ProCLAIM*'s Argumentation, introduced in §6, and the procedures to further specialise these schemes for the particular application described in §7.

Whilst argument schemes provide us with a means to generate arguments and question them, we also need a mechanism that will enable us to automatically evaluate the arguments and challenges generated in order to determine the ones that are acceptable. For this we make use of Dung's abstract argumentation theory [80] which has proven to be an influential approach to conflict resolution and non-monotonic reasoning over the past decade. This we discuss in the following section.

2.2 Argumentation Framework

As a result of the realisation that classical logic is not a suitable tool to attempt to formalise commonsense reasoning. Many recognised that rules used in commonsense reasoning are not universally valid but rather defeasible. This insight has led to the development of formalisms supporting various forms of non-monotonic reasoning where the set of conclusions does not monotonically increase with the set of premises. Several of these non-monotonic formalisms are based on a notion of *acceptable argument*, where non-monotonicity arises from the possibility that an argument may be defeated by a stronger counterargument. Exponents of this approach are Pollock (1987) [174], Simari and Loui (1992) [208], Vreeswijk (1993) [228], Dung (1995) [80] and Bondarenko, Dung, Kowalski, and Toni (1997) [58].

Particularly relevant is Dung's work [80] on argumentative semantics for logic programs. In which, in an attempt to reconcile many of the no-monotonic formalisms proposed in the literature, he presents an abstract argumentation framework, where arguments are characterised only by the arguments they attack and are attacked by, where *attack* relates to the notion of disagreement, or conflict, between arguments. Dung's work paved the way for other formalisms, most of them based on different versions of extended logic programming, such as [177, 92], among others. In the latest decade, Dung's abstract argumentation theory has become the mainstream approach in argumentation, where one of its great appeals is that arguments are taken as primitive elements. That is, their internal structure is left completely unspecified, and the focus is exclusively on the arguments' conflict-based interaction. This freedom to specify the arguments' internal structure enables different argument-based formalisms make use of Dung's argumentation framework. Once the argumentation framework is defined, the justified status of its arguments is evaluated based on their interactions. This evaluation is in turn based on the notion of an argument being acceptable with respect to a set of arguments if it is not attacked by a member of that set, and all its attackers are attacked by a member of that set.

Definition 2.1 *An* ***Argumentation Framework*** *is defined by a pair $AF =< AR, Attack >$, where AR is a set of arguments, and $attack$ is a binary relation on AR. [80]*

Where, if $A1, A2 \in \mathbf{AR}$ and $(A1, A2) \in Attack$ then, $A1$ and $A2$ are arguments and $A1$ attacks $A2$. So if we take the framework $< \{A1, A2, A3\}, \{(A2, A1), (A3, A2)\} >$, depicted in Figure 2.2a, we can intuitively see that the winning, *justified*, arguments are $A1$ and $A2$. Of course, in more complex frameworks, *i.e.* graphs of interacting arguments, identifying the acceptable arguments is not straightforward. This has motivated the definition of various semantics of arguments acceptability. Thus, for example, in the graph of arguments depicted in Figure 2.2d, under different semantics the arguments deemed acceptably vary.

Let us continue introducing some basic definitions of Dung's Argumentation Framework ([80]):

Definition 2.2 *Let $S \subseteq AR$ be a set of arguments, and $a, b, c \in AR$:*

- *S is **conflict-free** if there are no arguments $a, b \in S$ such that a attacks b.*
- *S attacks an argument a if there exists an argument $b \in S$ such that b attacks a.*
- *If an argument a is attacked by b which itself is attacked by an argument c, we say that c **defends** a from b. Thus, we say that S defends an argument a if for every argument b attacking a there exist an argument $c \in S$ such that c attacks b.*

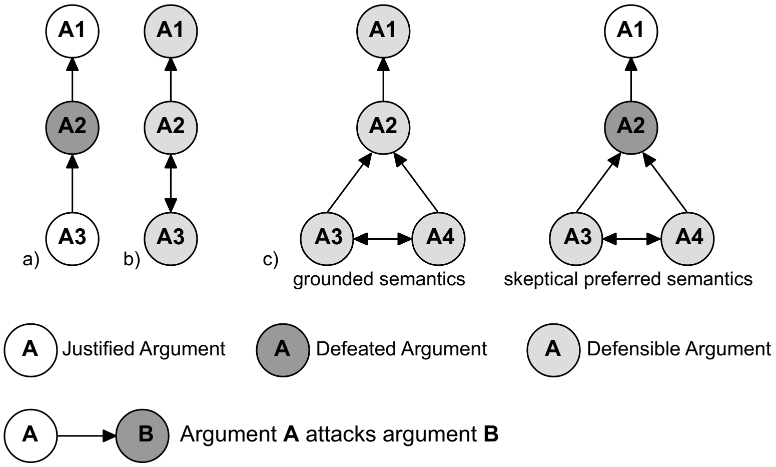


Figure 2.2: a) A simple Dung framework with three arguments; b) The mutual attack between argument $A2$ and $A3$ prevents deciding which are the winning arguments. c) A simple argumentation framework where it is not so intuitive to decide which are the winning arguments. We obtain different results under different semantics: the grounded extension is empty, the preferred extensions are $\{A1, A3\}$ and $\{A1, A4\}$ and the skeptical preferred extension is $\{A1\}$.

Thus in the above introduced framework (Figure 2.2a.), $\{A1, A3\}$ is conflict free set that attacks argument $A2$ and defends $A1$ from $A2$.

Let $S \subseteq AR$ be a set of arguments then: ([80])

Admissible S is an *admissible set* if and only if S is conflict-free and S defends all its elements.

Preferred Extension S is a *preferred extension* if and only if S is maximal for the set inclusion among the admissible sets of A.

Stable Extension S is a *stable extension* if and only if S is conflict-free and S defeats each argument which does not belong to S.

Complete Extension *S* is a *complete extension* if *S* is admissible, and each argument which is defended by *S* is in *S*.

Grounded Extension S is a *grounded extension* if it is the least (with respect to set inclusion) complete extension. Or equally, if S is the least fixed point of the characteristic function F of $< AR, Attack >$ $(F : 2^{<AR,Attack>} \rightarrow 2^{<AR,Attack>}$ with $F(S) = \{A$ such that A is defended by $S\}$

Skeptical Preferred Extension S is the skeptical preferred extension if and only if S is contained by all the preferred extensions. ([58])

Suppose we have the following framework $< \{A1, A2, A3, A4, A5\}, \{(A2, A1),$ $(A3, A2), (A2, A3), (A4, A3), (A5, A2), (A2, A5)\} >$ conforming the tree of arguments depicted in Figure 2.3. The admissible extensions of this framework are: $\{A1\}$, $\{A2\}$, $\{A4\}$, $\{A5\}$ $\{A1, A5\}$, $\{A2, A4\}$ and $\{A1, A5, A4\}$. These could represent, for example, defensible positions in a dialogue. The preferred extension are the maximal sets of the admissible extensions. Thus, they are $\{A1, A4, A5\}$ and $\{A2, A4\}$. Thus, there are two defensible position in the dialogue. The stable extensions coincides with the preferred extensions. While the grounded and the skeptical preferred extensions are both $\{A4\}$.

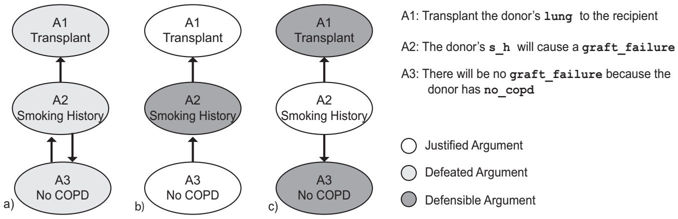


Figure 2.3: a) In this five-argument framework, argument $A1$ is undecided and $A5$ is the only argument of the grounded extension. b) If argument $A5$ is preferred to $A2$, the arguments $A1$, $A5$ and $A4$ would be justified under the grounded semantics. c) If $A2$ is preferred to $A5$, then argument $A1$ would be defeated and $A2$ and $A5$ justified under the grounded semantics.

In general, there may be more than one preferred and stable extensions in an argumentation framework, and only one, possibly empty, grounded and skeptical preferred extension. The different extensions of the preferred and stable extensions can represent options of acceptable positions in an argumentation. Thus a participant may either endorse arguments $\{A1, A4, A5\}$ or $\{A1, A4\}$ and they will both be acceptable positions. In that sense, the preferred and stable extensions are both *credulous*, either position is acceptable. In that same sense the grounded and skeptical preferred extensions are *skeptical*. Of course, we should seek for a skeptical approach for identifying the acceptable arguments when deciding whether or not a proposed action can safely be performed, since we are not interested in an *admissible* solution, but that which is most safe. For this reason we choose to take to grounded semantic. One advantage of the grounded semantics is that computing its extension is a linear problem and that the extension always exists, though it may be empty (for this reason the grounded semantic has been argued to be too skeptical which may be a disadvantage in some contexts [81]). Now, it is important to note the difference between an argument $A4$ in the grounded extension, argument $A3$ attacked by an argument in the grounded extension and argument $A1$ which is neither in the grounded extension nor it is attacked by it. Thus, this suggests three notions of arguments acceptability, those that are wining or *justified* arguments, those which are loosing or *defeated*, arguments and those that are simply undecided, which are sometimes called *defensible*.

Definition 2.3 *Given an argumentation framework $< AR, Attack >$, with G the grounded extension. Then, if $a \in AR$:*

- *a is said to be **justified** if $a \in G$.*
- *a is **defeated** if there exist an argument $b \in \mathbf{G}$ such that $(b, a) \in Attack$.*
- *Otherwise, a is said to be **defensible**.*

Thus, under the grounded semantics we can say that $A4$ is justified, $A1$ is defensible and $A3$ is defeated.

Suppose the above introduced framework, depicted in Figure 2.3, represents the following argumentation:

$A1$**:** Transplant the available kidney to the recipient.

$A2$**:** Because the donor cause of death is a streptococcus viridian endocarditis, the recipient will result having a streptococcus viridans infection , which is a severe infection. And thus, the transplant should not be performed.

$A3$**:** Treating the recipient with penicillin, can prevent the recipient's infection.

$A4$**:** The recipient is allergic to penicillin.

$A5$**:** Treating the recipient with teicoplanin, can prevent the recipient's infection.

Knowing that $A4$ is justified and $A1$ is defensible (undecided), in this context, is of little help. A desirable solution to the argumentation is one in which $A1$ is deemed either justified or defeated. Note that what prevents $A1$ from being either justified or defeated is the symmetrical attack between arguments $A2$ and $A4$. To address this situations numerous works ([24, 48, 159] have extended Dung's argumentation framework so that an attack from an argument $Arg1$ to an argument $Arg2$ can be disregarded if $Arg2$ is for some reason stronger than or preferred to $Arg1$. Then, the acceptable arguments are evaluated based only on the *successful attacks* (that are usually referred to as *defeats*). This allows for example to resolve local disputes between mutually (symmetrically) attacking arguments, so that if $Arg1$ attacks $Arg2$ and $Arg2$ attacks $Arg1$, then a relative preference over these arguments will determine that one asymmetrically defeats the other. In particular, if $A5$ is preferred to $A2$, indicating that teicoplanin can effectively prevent the recipient's infection, argument $A1$ would be deemed justified, and so the transplant would be deemed safe. On the other hand, if $A2$ is preferred to $A5$, the action will be deemed unsafe. Namely, $A1$ would be deemed defeated.

The preference relation between arguments usually stems from a strength associated to the rules and facts with which the arguments are constructed, where the strength usually represent the degree of the agent's belief in a fact or rule being the case [25, 181]. In §8.2 we describe how *ProCLAIM*'s preference assignment is derived from three independent knowledge resources, so that the final decision accounts for domain guidelines and regulations, evidence collected from past deliberations and exerts' opinions.

Although Dung's theory has become the standard approach for argument evaluation, there are some works, though very few, that take alternative approaches to compute the status acceptability of arguments. Most notably is the Carneades model [101]. Underlying the Carneades system we presented above (in §2.1) is a formal model also called Cardenades. This is mathematical model of argument structure and evaluation which applies *proof standards* to determine the acceptability of statements on an issue-by-issue basis. As opposed to Dung's argumentation framework where the only native relation between arguments is the attack relation, the Cardenades model continues the tradition of Kunz and Rittel's Issue Based Information System [138] where arguments can be *pro* or *con* an issue, or a statement. Gordon *et al.* argue in [101] that the Dung's approach, where arguments are deemed justified if they survive all attacks by counterarguments, ignores that arguments are embedded in a procedural context. They argue that different proofs standards should be applied to different situations. For instance, in a brain storming phase of a dialogue a weak proof standard may be appropriate, such as *Scintilla of Evidence* standard where a statement meets this standard if it is supported by at least one defensible *pro* argument. Where a defensible argument here is an argument which all its premises hold. In a different phase of the dialogue a different proof standard may be more appropriate, such as the *Dialectical Validity* standard where a statement meets this standard if it is supported by at least one defensible pro argument and none of its con arguments are defensible.

While it could be argued that the different proof standards can be formalised under Dung's framework by shifting among different semantics and accommodating the arguments' notion of acceptability, the Cardenades model already embeds this procedural mode of argument evaluation in its formalisation, facilitating in this way its implementation. In *ProCLAIM* deliberation there is no need to define different proof standards nor there is a need to define pro arguments, for which we took the standard Dung's theory approach.

Another important contribution of the Cardenades model, strongly related to a dialogical context, is that it explicitly addresses the notion of *burden of proof*. Namely, it models the obligation the participants in the deliberation may have to defend a proposed argument when one of its premisses is challenged, where a challenge may be regarded as a dialogical move requesting evidence in support of the challenged premise. The effect this has in the argument evaluation is that an argument may effectively be defeated by an unreplied challenge depending on which party made the challenge and to which is the challenged premiss. That is, the argument may be deemed defeated if the challenge is not met. For instance, in the law of civil procedure the burden of proof may be allocated to the party who has the better access to the evidence. Hence, in this context, a challenges to a premiss may be deemed ineffective if it is made by the party that has better access to the evidence for or against the challenged premiss.

While *ProCLAIM* allows for arguments to be challenged, it does not account for the notion of burden of proof. That is, while participants in the deliberation can request for evidence in support of a given premiss, no agent is under the obligation, from the argumentation point of view, to provide this evidence. That is, participant agents do share an obligation to provide their knowledge about the problem at hand, but not to defend any particular position.

The approaches to argument modelling described above (broadly speaking: argument schemes and Dung's theory) form the basis for some of the elements in our system that are used to generate and evaluate arguments. However, as it becomes clear when discussing the Carneades model, we also need to account for the context in which the agents' argumentation dialogue takes place and the purpose of this dialogu. One important aspect that shapes *ProCLAIM*'s dialogues is that they are deliberative.

2.3 Deliberation Dialogues

As defined by Walton and Krabbe in their influential classification of human dialogues [232], deliberation dialogues involve participants deciding what action or course of actions should be undertaken in a given situation. Typically participants in such a dialogue believe that they share a responsibility for deciding what to do, which provides a collaborative context to the dialogue. Besides deliberation other five primary dialogue types are characterised in [232], where the classification is based on the dialogue goal, the participants individual goals and the information they have at the beginning of the dialogue. Thus, in **Information-Seeking Dialogues** an agent seeks to answer a question made by its interlocutor, who believes the former knows the answer. In **Inquiry Dialogues** agents collaboratively seek to answer a question which is unknown to all of them beforehand. In **Persuasion Dialogues** an agent, typically characterised as the proponent of the dialogue, seeks to persuade his interlocutor, the opponent, to accept a statement he does not yet endorse. **Negotiation Dialogues** involves dividing some scarce resource (including the participants' time) which defines an adversarial situation where agents need to bargain in order to maximise their share of the resource. And finally in **Eristic Dialogues**, participants seek to vent perceived grievances, and the dialogue may act as a substitute for physical fighting.

While information-seeking and inquiry dialogues relate more to clarifying an issue, negotiation, persuasion and deliberation may be used for multi-agent decision making. However, if intending for providing an environment for a fully cooperative interaction, it is worth taking into account that persuasion and negotiation approaches may hinder this purpose. As discussed in [113]:

- In deliberation dialogues agents have no fixed initial commitment with respect to the basic subject matter of the dialogue. As opposed to persuasion dialogues typically characterised by the figures of a proponent and an opponent. Although in deliberation dialogues agents exchange proposal and express their positions about what is to be done, the goal is to jointly reach the best or most sensible thing to do, rather than to defend a particular position.

- In deliberation dialogues the agents' individual interests may influence their way of viewing the problem and evaluating the exchanged arguments. However, focus is not on reconciling competing interests as it is in negotiation dialogues, where in the purpose of reaching an individual goal a participant can hide information and individual preferences. In a deliberation participants benefit from sharing all relevant infirmation and personal interest as a way to reach the best collective solution.

In collaboratively deciding whether a proposed action is or not safe it is paramount that selfish interests do not guide the course of the dialogue. Hiding information or any strategic consideration should not preclude from exploring all the relevant facts to be considered. It may, of course, be the case that within a deliberation participants find the need to divide some scarce resource, for example decide who does what, then the purpose of the dialogue changes and so agents may play a different game.

Deciding whether an action is safe or not may be rephrased in terms of a persuasion dialogue. A proponent defends the statement *the action is safe* from an opponent. In this context participants may consider that it is only worth exploring that which is disagreed upon. Therefore, if all participants believe that an action is safe (resp. unsafe), they may see no need undertake a dialogue. In general, for any argument put forward during the dialogue that all participants agree upon, it will not be questioned. Figure 2.4 illustrate cases where agents agree upon a statement but each based on different reasons. If these reasons are not unfolded the wrong decision may be taken. In that sense, deliberation dialogues is about exploring all the relevant aspects with respect to the final decision, since its goal is not to persuade one another, but to collaboratively decide what is the best thing to do, as a group. And, in particular, this may result in collaboratively arriving to a proposal deemed optimal by the group that may have been considered suboptimal (not appropriate) by each of the participants as illustrated in Figure 2.4.

In recent years several formal models of these dialogue types have been developed. For instance [115] proposes a model for information-seeking dialogues, in [152] a model for inquiry dialogues is proposed, models for persuasion dialogues are proposed in [176, 36, 27, 234] and models for negotiation dialogues are proposed in [26, 149, 199]. Furthermore, given that in most real-world dialogues these primary types are intertwined, some models have been proposed for complex combinations of primary dialogues, for example, iterated, sequential, parallel and embedded dialogues [153, 188]. To our knowledge, the only formal models proposed for deliberation dialogues are McBurney and Hitchcock's Deliberation Dialogue Framework [150] and the recent adaptation of Prakken's persuasion dialogue game [176] into a deliberation dialogue game [132]. Dialogue formalisation are usually based on

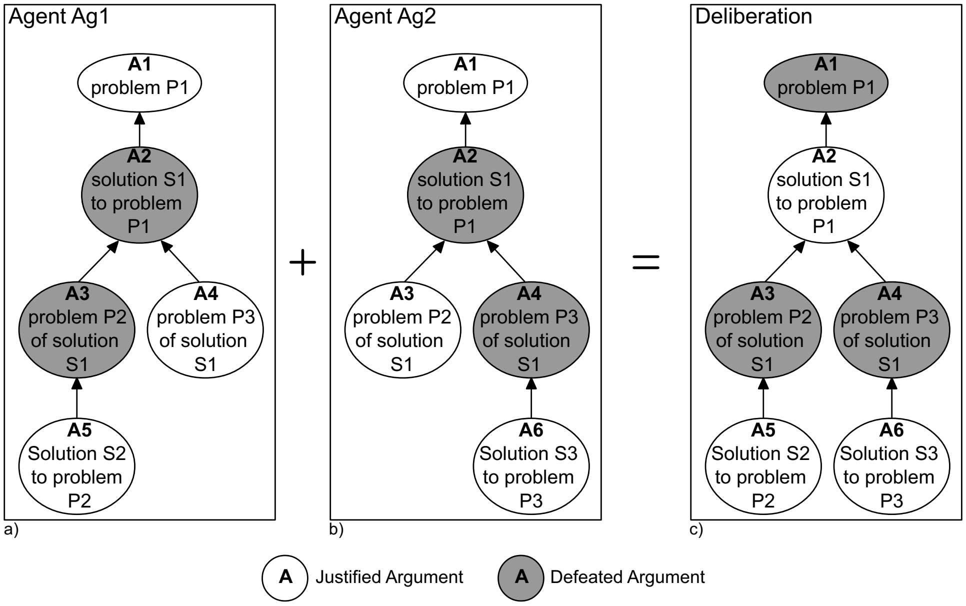


Figure 2.4: Agents $Ag1$ and $Ag2$ both believe that problem $P1$ cannot be solved. So if engaged in a persuasion dialogue, they would have no motives to unfold the reasons for which they believe $P1$ to be irresolvable. However, in a deliberation participant agents should provide these reasons and in so doing they may jointly find a way to solve $P1$.

the notion of a dialogue game which defines the agents' interaction in the dialogue. We thus introduce the basic notions of a dialogue game in the following section.

2.4 Dialogue Game

Dialogue games are interactions between two or more participants who *move* by uttering locutions, according to certain rules. They were first studied by Aristotle [28] and then by argumentation theorists and logicians in the post-war period (*e.g.*, [110, 144]). Over the last decade they have been applied in various areas of Computer Science and Artificial Intelligence, particularly for rule-governed interactions between software agents[4].

A dialogue game may be specified by listing the legal locutions, together with the rules which govern the utterance of these locutions, the opening and termination of dialogues, and the rules for manipulation of any dialogical commitments incurred by the participants during a dialogue [153]. Where the notion of commitment, made popular by Hamblin in [110], is used as a way to ensure some degree of coherence and consistency in the agents' participation in the dialogue.

Dialogue games can be described in terms of their *syntax*, *semantics* and *pragmatics*.

[4] For a recent and detailed review of dialogue games applications, particularly addressing argumentation dialogues, we refer the reader to [155].

Thus, following the standard division proposed in linguistic theory where broadly speaking [142]: syntax is concerned with the surface form and combinatorial properties of utterances, words and their components; semantics focuses on the truth or falsity of utterances; and pragmatics on those aspects of the meaning of utterances other than their truth or falsity (*e.g.* the desire and intension of the speaker).

The *syntax level* takes care of the specification of the possible locutions the participant agents can make and the rules that govern the order in which these locutions can be made. This specification may account for the agents dialogical commitments, which are usually stored in a publicly-readable database, called a commitment store. This is because, the agents' incurred commitments may affect the locutions these agents may make at the different stages of the dialogue.

The locutions are typically defined in two layers [139], where an inner layer contains the content, or topic, of the message, while the outer layer (wrapper) express the illocutionary force of the inner content. Classical example of this are the two agent communication languages KQML [85] and FIPA ACL [86].

The *semantic level* of the dialogue game is usually concerned[5] with facilitating the designers task in developing artificial agents that have a shared understanding of the meaning of the messages they exchange during the dialogue. To this end a number of semantics have been developed [221]. Most frequently used are the axiomatic semantics [158] and the operational semantics [221]. The former defines each location in terms of the pre-conditions that must exist before the locution can be submitted and the post-conditions which apply following its submission. These pre and post conditions provide a set of rules to regulate participation in rule-governed interactions. The operational semantics, on the other hand, sees locutions as computational instructions which operate on the states of some abstract machine. The participant agents together with the dialogue are regarded conceptually as parts of an abstract computer, where the overall state of this computer is altered with the submission of valid locutions or with the participant agents' internal decision processes. Operational semantics may be used to study the more formal aspects of a dialogue, *e.g.* whether there are non reachable states of a dialogue, particularly useful to prove that implementing certain interaction protocols the dialogue can reach a termination state.

The *pragmatics level* is strongly influenced by the speech act theory proposed by Austin [41] and Searle [201], that classified utterances by their intended and actual effects on the world, including the internal mental states of the receiver agent. A classical example of this is FIPA's axiomatic Semantic Language [86] defined in terms of the participant agents' beliefs, desires and intentions. For example an *inform* locutions from an agent $Ag1$ to an agent $Ag2$ of a property p requires *1)* $Ag1$ to believe p to be true; *2)* $Ag1$ to have the intention for $Ag2$ to believe p to be true; *3)* $Ag1$ to believe that $Ag2$ does not have beliefs about p.

In a similar fashion the FIPA ACL defines in total 22 different locutions. McBurney and Parsons extends this language, with their *Fatio* protocol [154], in order to support ar-

[5] The semantics of the dialogue game may also focus on other issues, *e.g.* provide a means by which different agent communications languages and protocols may be compared with one another formally and with rigor [155].

gumentation by defining five additional locutions: assert, question, challenge, justify and retract. This extension is consistent with the FIPA axiomatic Semantic Language, thus, the semantic of each such locution is defined in terms of the participant agent beliefs desires and intentions.

Numerous dialogue games have been proposed for argumentation dialogues, most of which model persuasion [38, 52, 176] and negotiation dialogues [207, 156, 184] and very few actually address deliberation dialogues [150, 132]. The most elaborated dialogue game intended for deliberation is the framework for ideal Deliberation Dialogues proposed by McBurney, Hitchcock and Parsons in [150]. In that framework, deliberation dialogues may proceed through eight successive stages:

Open Stage in which each dialogue participant explicitly enters the dialogue.

Inform Stage in which participants share information about the goals of the dialogue, the constraints under which any decision must be made, their evaluation criteria for decision options, and relevant facts about the world.

Propose Stage in which proposals for actions (or courses of actions are made).

Consider Stage in which preferences between action-proposals and justifications for such preferences are expressed.

Revise Stage in which previously-uttered proposals are revised in the light of utterances made in the Inform or the Consider stages.

Recommend Stage in which one action proposal is formally considered for acceptance or rejection by all the participants, for example by means of a vote.

Confirm Stage in which an accepted proposal is confirmed as the decision of the dialogue participants.

Close Stage in which dialogue participants withdraw from the dialogue.

The participants's goal in these dialogues change according to the stage of the dialogue. It is for this reason that stages are marked explicitly, so as to better enable participants to know what is expected of them at each stage. In this framework, there are some constraints on the order of the stages, and some stages may be repeated, in an iterative fashion, before the dialogue is completed. Nonetheless, the deliberation is quite liberal in the sense that very few restrictions are imposed on the participant agents. In [150] the axiomatic semantics of the deliberation dialogue is presented, with the *pre* and *post* condition for the locution, the meaning of the locution, whether for each locution any response from any other participant is required and the effect the locution has on the commitment store.

As the deliberation progresses, the participants' assertions (which may be action proposals, intended goals, constraints, perspectives by which a potential action may be evaluated, facts and evaluations) and their submitted preferences are added to a commitment store capturing the agent's expressed beliefs and intentions[6]. Each such submitted assertion or preference may be retracted and thus removed from the commitment store. While the dialogue game does account for the conflicts on beliefs and interests among the participant agents, to a great degree their conflict interaction and resolution are left implicit.

ProCLAIM's dialogue game, introduce in §5 bears certain similarity with McBurney's *et al.* dialogue game in that it defines a number of stages and interaction layers through which the deliberation may move. However, we take these stages as conceptual, in the sense that they only help organise the game's definition, participant agents need not know which locution corresponds to each stage and locutions of different stages may be submitted in parallel. A great difference between the two approaches is that while McBurney's *et al.* dialogue game is intended for an open deliberation about what to do, *ProCLAIM*'s dialogue game is specialised for deliberating over the safety of a proposed action where great part of the interaction is shaped (and constraint) by the argument schemes the participant agents can instantiate at each stage of the deliberation. One immediate consequence of this is that the conflict between the agents' positions is explicitly captured in a form of a Dung argumentation graph and thus, the resolution of the deliberation involves identifying the acceptability status of the argument proposing the safety critical argument.

Another deliberation dialogue game was proposed in [132]. This work is an adaptation of Prakken's persuasion dialogue game [176]. The main argued contribution of this work is to explicitly model the conflict interaction between the agents' submitted arguments so as to have an account of the acceptability status of the agents' submitted argument at each stage of the deliberation. Both [132] and [176] assume the argument interaction to form a Dungs' argumentation framework under the grounded semantics. In both these game, each submitted argument must defeat the argument they reply to. Thus, while this enables to directly identify the winning arguments, these games assume that the relative strength of the arguments are known a priory, which is not the case in *ProCLAIM*'s argumentation. In [132], for each action proposal there is an independent dialogue tree where the proposal is evaluated. A choice is given among those proposals that survive the agents' attacks ordered in terms of the agents' stated preferences. In [11] we have used Prakken's persuasion dialogue game to model the agents' interaction in *ProCLAIM*. However, besides the fact that this dialogue game was intended for persuasion rather than deliberation, both [132] and [176], as well as McBurney's *et al.* [150], fail to capture some of *ProCLAIM*'s deliberation nuances. For example, *ProCLAIM* deliberation decouples the resolution of *what is the case* and the deliberation over the actions' safety. Firstly, this gives priority to the main question:–*Is the action safe in current circumstances?*-so that, for example, questioning the current circumstances is licensed only if this challenges the action's safety. Secondly, as we show in §5, this decoupling allows one to address, in a relatively simple fashion, problems such as incomplete or uncertain information, at the time of constructing the arguments, when updating the new available information and when evaluating the agents' submitted arguments.

Similar argument can be given for the main difference between *ProCLAIM* dialogue game and the PARMA (for Persuasive ARgument for Multiple Agents) Action Persuasion

[6] The participant agents' expressed beliefs may not be their actual, internal, beliefs as in the case of FIPA ACL [86].

Protocol [38]. PARMA is the dialogue game proposed by Atkinson *et al.* in order to implement their theory of persuasion over actions that we introduced in 2.1. Nonetheless, to a great degree this dialogue game is related to that of *ProCLAIM* because its argumentation moves are defined in terms of the argument scheme Atk and its associated CQs. And as we have discussed in 2.1 it is on this scheme and CQs that *ProCLAIM*'s circuit of schemes and CQs are based on (see §6). In a similar fashion as in [150], the participants' beliefs and interests, potentially in conflict, are added into a commitment store and no explicit model for the argument interaction and resolution is given. However, as it can be seen in their later work [34], a Dung's argumentation framework can easily be constructed from the dialogue commitment store.

Furthermore, for the intended scenario a collaborative decision making is required in which the different stakeholders' view points (possibly in conflict with each other) need to account for. Traditional approaches based on numerical paradigms may indeed help in the process of argument construction, validation and evaluation, as all these processes require domain knowledge and are agnostic to where the knowledge comes from. What *ProCLAIM* propose is the integration of any kind of knowledge by means of a deliberation dialogue so that different perspectives can be accounted for and in particular so that domain expert can effectively participate in the decision making.

Having described the foundation on which *ProCLAIM*is based on for the argument construction, interchange and evaluation, we know briefly comment on an additional aspect of argumentation that has a great value for the intended use of *ProCLAIM*. That is, argumentation as a tool to facilitate human-computer interaction.

2.5 Argumentation for Human-Computer Interaction

The use of Argumentation for Human-Computer interaction has intensively been explored over the lasts 20 years. This came with the acknowledgement that the knowledge derived from a computer is easier to understand and thus more persuasive for human users if presented in a structured, argumentative fashion [44, 241, 240, 164], as opposed to, for instance, if presented as a set of rules.

The progress made in argumentation theory in capturing commonsense reasoning together with the diagrammatic nature of argumentation schemes, such Toulmin's [220] depicted in Figure 2.1 or Kunz and Rittel's Issue Based Information System [138] depicted in Figure 2.5 has motivated a number of argumentation-based visualisation tools [213, 98, 191, 222, 83, 61, 206] particularly interested in addressing ill-defined problems in collaborative settings. Argument visualisation allows presenting complex reasoning in a clear and unambiguous way facilitating communicative interactions and the development of shared understandings among participants in the decision making process, which ultimately leads to more coherent, focused and productive discussions [114, 130] even in time pressure situation [145]. While no tool, to our knowledge, has based its graphical representation on Dung's argument graphs, their use in the academic literature to illustrate the interaction among arguments is pervasive. Through the use of directed graphs, as those used throughout this Thesis, the argument interaction can easily and intuitively be grasped, and by colouring the nodes of the graph one can immediately identify which are the winning, loosing and undecided arguments (under the chosen semantics). This is particularly the case when the argument graphs have a relatively simple structure as those in *ProCLAIM*, which have a tree structure.

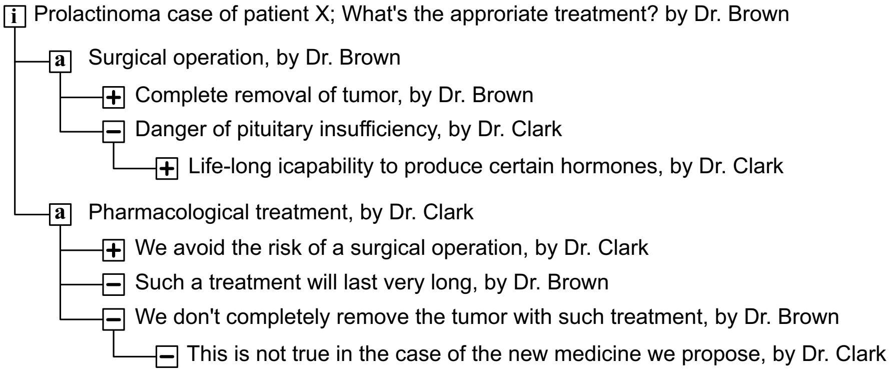

i Prolactinoma case of patient X; What's the approriate treatment? by Dr. Brown
a Surgical operation, by Dr. Brown
+ Complete removal of tumor, by Dr. Brown
− Danger of pituitary insufficiency, by Dr. Clark
+ Life-long icapability to produce certain hormones, by Dr. Clark
a Pharmacological treatment, by Dr. Clark
+ We avoid the risk of a surgical operation, by Dr. Clark
− Such a treatment will last very long, by Dr. Brown
− We don't completely remove the tumor with such treatment, by Dr. Brown
− This is not true in the case of the new medicine we propose, by Dr. Clark

Figure 2.5: Classical visualisation of an argumentation systems based on the Issue Based Information System. In the picture **i** is the *issue* or *dominant question*, **a** are *alternatives* to resolve the issue, and the plus and minus signs are *positions* in *favour* or *against* an alternative or of another position. Example extracted from the Hermes System [120], an argument-based collaborative decision support system.

In addition to the visual appeal of argumentation, another meeting point for the interaction between humans and computers resides in the fact that arguments can be built from argument schemes represented in natural language. This leads to the idea of using argument schemes as a means for a human and artificial agents interaction using natural language. In [17] we followed this idea and proposed a repository of argument schemes, tailored for reasoning over the viability of an organ transplant, so that through the use of these schemes heterogeneous (*e.g.* human an artificial) agents can argue over a transplant safety. In a later work [11] we presented a prototype that supported this argumentation process. Although this idea has been envisioned elsewhere [194] and to some extent it is a present idea in other approaches, such as in the Carneades System[7] or Atkinson's *et al.* work [33], we know of no other implementation that supports this kind of argument dialogue among human and artificial agents. In §7 we discuss this use of argument schemes, in §10.1 discuss the implemented prototype we presented in [11]. Before we continue however, it is worth noting that the links between argumentation and natural language representation goes beyond this use of argument schemes, see [189].

All this to say that the reasons for choosing argumentation to address the intended problem not only includes its suitability for modelling commonsense reasoning and communication between reasoning agents but also, because it delivers ready made artifacts for human-computer interaction. As we just pointed out, argumentation has been shown to be a good means for presenting information to laymen in a clear and persuasive fashion. With

[7] http://carneades.berlios.de/

the use of argument schemes this information can be presented in natural language without the need for tools for natural language generation. Moreover, this presentation can be further improved with the use of visual support so as to provide users with a global view of the arguments' interaction along with an intuitive understanding of which are the winning, loosing and undecided arguments.

2.6 Argumentation and Safety Critical Domains

Along the 90s numerous works in Artificial Intelligence proposed alternatives to address decision making under uncertainty (see [169] for a review). These were mainly driven by the practical difficulties of implementation of classical decision theory based on probabilistic inference [227, 187], specially in practical situations where it is difficult or simply not possible to quantify uncertainty. As an alternative to these *numerical* approaches, many works proposed the use of argumentation as a *qualitative* method to address decision making under uncertainty [90, 136, 135, 141, 218], where many of these works were particularly concerned with risk assessment. One early result of this research line is the StAR System [137, 118], a software application intended for identifying the risk of carcinogenicity associated with chemical compounds. Given the lack of environmental and epidemiological impact statistics, the StAR System uses general scientific knowledge to construct arguments for or against the carcinogenicity of the chemical in question. On the basis of the presented arguments the gravity of the risk may be estimated. Another important contribution in this research line es Fox *et al.*'s *PROforma* [89, 87], a language and technology for designing, implementing, testing and delivering software agents that can operate in dynamic and uncertain environments. With a similar approach of the StAR System, the *PROforma* application RAGs [76] was designed to assist a family doctor in evaluating the genetic risk of breast cancer for a given patient. Following this trend Glasspool and Fox proposed the REACT application [96, 97], intended to assist medical doctors in the creation and enactment of care plans, where arguments for and against possible clinical interventions are presented so as to facilitate the doctors' decision making.

An important aspect shared by StART, RAGs and REACT is the emphasis in communicating the rationale for the proposed solutions. These systems where no longer conceived as *expert systems* but as *decision support systems*. As stated in [88, 95], when computational applications are intended for a field which is not fully understood (*e.g.* medicine), it should account for the fact that any knowledge base, even if based on state-of-the-art in the filed, may contain no demonstrable errors or inconsistencies. Hence, though this knowledge base may operate as intended, its advice may be sub-optimal or even unsafe. This suggests restricting these computational applications to decision support rather than decision-making. The interface then becomes central to displaying the reasoning by which the program has arrived at its recommendations in an understandable fashion, for which, as discussed in §2.5, argumentation is particularly suitable. Providing end user with the reasons for a recommendation not only makes the advice more persuasive when appropriate, it may help experienced users not follow unsafe or sub-optimal advise when they disagree with the given reasons or when they believe these fail to capture important aspects of the decision making.

Indeed, a structured and clear presentation of the arguments for a proposed decision (or solution to a problem) are particularly useful in safety-critical domains, for they not only provide the reasons behind the proposal, but they also helps verifying that all relevant factors are taken into consideration. A good example of this are safety cases, where a safety case consist of a structured collection of arguments and evidence used to provide an assessment of the potential risks in a project (*e.g.* a software, technology or a facility, for instance a nuclear power plant) and of the measures to be taken to minimise these risks so that the final product is acceptably safe to operate in a particular context over its lifetime. Since the proposal of McDermid [157] to use of Toulmin's argument scheme [220] (see fig. 2.1) to structure the arguments conforming a safety case, numerous works have elaborated on this idea proposing extensions and variation to this schema that better suits the formulation of their safety cases [53, 20, 127, 128, 83, 109]. These schemes are used both to help elaborating the safety cases in a structured fashion and to later facilitate the reviewing process by supporting navigation across the argument structure (the claim, the evidence in its support, the warrant...) using tools such as hypertext.

The above mentioned proposals, while addressing various complexities of decision making in safety critical domains, they do so by adopting the perspective of an impartial, omniscient observer. That is, they do not address the fact that different knowledge sources (*e.g.* expertise, stakeholders,...) may be in disagreement. McBurney and Parson address this issue by proposing their *Risk Agoras* [151], an inquiry dialogue game intended to formally model and represent debates in the risk domain. Though the scope of application may be wider, their work focusses in modelling discussions over potential health and environmental risks of new chemicals and substances, as well as the appropriate regulation required for these substances. That is, their concern is to facilitate public policy debates about the risks of new substances. Their work is inspirited by Habermas' proposed framework [108] intended for consenting members of a community to engage in a civil discourse. In this framework, and so in its computational version proposed by McBurney and Parson[8], participant agents may not only advance arguments for or against a proposal, but they may also question the participants' given definitions or the modes of reasoning used in their arguments formulation. That is, participants may discuss over which rules of inference or argument schemes to use. In other words, these *Risk Agoras* advocate for an ideal public debate, promoting both transparency in public policy decision making and democratic participation in debates about environmental and health risk.

A more recent work in multi-agent argumentation concerning risk scenarios is Fox *et al.*'s guardian agent [160]. In this work a Guardian Agent is proposed to supervise the different medical treatments prescribed for a patient by different Specialised Medical Domain Agents to jointly arrive at the most appropriate, safe, treatment. The focus of this work is to explore the use of a *Logic of Argumentation* [135] for agents' decision making. However, this work does not address the agents' dialogical interaction.

Another interesting work addressing agents' interaction is Black and Hunter's inquiry

[8] We know of no implementation of this system.

dialogue system [57]. Broadly speaking, they propose an enquiry dialogue game as a procedure to find the *ideal* outcome from the interaction between two agents. This ideal outcome is the solution that would arise from joining the two agents' belief bases, which may themselves be inconsistent. Therefore, the outcome of such dialogues is predetermined. They argue that this kind of dialogue would be suitable for safety critical scenarios where the purpose of the argumentation would be arriving to this 'ideal' solution. One very important added value of this approach is that they can prove soundness and completeness properties of inquiry dialogues. The downside is that the dialogue game only accepts two agents, it assume they have the same knowledge representation and that their belief base do not change during a dialogue.

All the above works make use of important features of argumentation: dealing with uncertainty, providing a clear understanding of complex reasoning (via informal argumentation, in the example of safety cases), modelling agents' reasoning and dealing with disagreement among rational agents. However none of the above mentioned works addresses the particularities of *ProCLAIM*'s deliberations which main purpose is to provide a setting for an effective and efficient for *heterogeneous* agents to deliberate over the safety of a proposed action. As we will see in §6 *ProCLAIM*'s interaction protocol, defined on the basis of schemes and CQs, imposes numerous restrictions on the argumentation which are motivated by the context of application. These restrictions are intended to promote efficiency and affectivity by focusing the deliberation only on the relevant matters to be discussed. However, *ProCLAIM* does not make any assumption nor imposition on the agents' internal reasoning (as [57] or [160] do). As discussed in §2.1, *ProCLAIM* uses argumentation as a meeting point, via argument schemes, for the different knowledge resources interaction (*e.g.* human and artificial agents or the Case-Based Reasoning component featured by *ProCLAIM*).

On the other side of the spectrum, approaches such as Risk Agoras [151] or McBurney *et al*'s Deliberation Dialogue Framework [150] assume ideal conditions for the argumentation, where for example, agents have no time constraint to engage in a deep discussion. Furthermore, because the scope of the discussion is so wide, participant agents are given little support in the argument construction. These assumptions are not realistic for *ProCLAIM*'s deliberations. As pointed out in [131] and later confirmed by [145], the otherwise quite intuitive fact, that in collaborative decision making communication decreases as participant are more stressed, either with workload or because their faced with an emergency situations, *e.g.* a rapid deterioration of a patient [145].

ProCLAIM deliberation assume a context where most issues are beyond dispute (*e.g.* in the medical scenario questioning the morality or general effectiveness of organ transplantation is beyond dispute) in order to focus de deliberation only on the relevant matters. Furthermore, the idea is that in normal circumstances the human agents will only have to validate the automatically generated proposal (presented as a tree or interacting arguments), and only when they disagree with the proposed decision that they are directed in the submission of the arguments that better embody their understanding of the problem.

There are some recent works that may be useful for a future extension of the *ProCLAIM* model. In [237] an ontology-based argumentation framework is proposed within the medical domain. This work can be useful at the time of constructing the scenario specific argument schemes in medical domains. While *ProCLAIM* does provide support in this process, it assumes a domain specific ontology. In a more recent work [103] a language for encoding clinical trials and an argumentation-based framework for comparing the efficiency of clinical treatments is proposed. Currently, in *ProCLAIM* different treatments are not compared so as to choose the best treatment, rather, any complementary treatment to the main proposal is considered either effective or not effective in preventing an undesirable side effect. That is, *ProCLAIM* is not intended for choosing the best treatment but rather, whether or not the proposed treatment is safe. For this reason it would be interesting to explore this option in future work.

Having introduced the context in which *ProCLAIM* is defined, the grounds on which it is built as well as a review of similar works, we proceed in the following chapter to introduce to two scenarios used to motivate the *ProCLAIM* model, where the first and central use case is the transplant scenario introduced in §3.1 and the second, complementary use case, is the environmental scenario discussed in §3.2. Later in §4 we start our presentation of the *ProCLAIM* model. We would like to add at this point, that our goal with the proposed model is to facilitate argumentation based deliberation of the type that occurs amongst human users, as we aim to support deliberations among domain experts that may have conflicting opinions and judgements. Indeed as seen throughout this chapter, one of the acknowledged strengths of argumentation as a paradigm for is that it not only captures formal logic-based models of reasoning in the presence of uncertainty, but also supports human reasoning and debate, and thus provides a bridge between formal models and human modes of reasoning, so that the former can normatively guide the latter. In this view, quantitative theoretical models that involve the use of probabilities and numerical risk values, do not reflect the way discussion and debate takes place in human practice, and so do not lend themselves to supporting human deliberation. Furthermore, as shown in [76], there is evidence that argumentation based qualitative approaches yield comparable outcomes to numerical risk based approaches in the medical risk assessment domain.

Chapter 3

Two Safety Critical Scenarios

In this chapter two safety critical scenarios are presented, one in the field of human organ transplantation and the other addressing wastewater management in river basins. In both scenarios the implementation of an efficient deliberation among experts, of the type proposed by *ProCLAIM*, can be of great value. In the former scenario, *ProCLAIM* has the potential to help reduce the increasing disparity between demand and supply of human organs. While in the environmental scenario the implementation of such model can help reduce industrial wastewater discharge's environmental impact in river basins. The two scenarios serve to motivate *ProCLAIM*'s value and illustrate its use.

The transplant scenario is the main case study of this Thesis, a number of prototypes have been developed in this scenario addressing different aspects of the problem, and also positive feedback have been received from domain experts in assessing the proposal acceptance in the transplant community. The environmental scenario is taken as a complementary case study which serves to test the scope of the model's applicability. A full research on the environmental problem using *ProCLAIM* is in fact taken by the LEQUIA research group[1] that, as non-experts in argumentation, their experience has been (and still is) very helpful in developing (and refining) a methodology for the instantiation of *ProCLAIM* in new scenarios.

In the following two sections the safety critical scenarios are introduced and in §3.3 a discussion is given regarding the role these scenarios played in the development of *ProCLAIM*. After introducing the *ProCLAIM* we return in §11 to these two case study to discuss our experience in implementing them.

3.1 Human Organ Transplantation

In the last 30 to 20 years human organ transplantation has consolidated as a common-practice therapy for many life-threatening diseases. However, while the increasing success of transplants has led to increase in demand, the lack of a concomitant increase in donor organ availability has led to a growing disparity between supply and demand. This scarcity

[1] http://www.lequia.udg.es/

of donors has in turn led to the creation of national and international coalitions of transplant organisations intended to make the best use of the organs that are made available. This has resulted in requirements for managing and processing vast and complex data, and accommodation of a complex set of regulations in a distributed context.

To facilitate the transplant professionals' work in this increasingly complex and demanding tasks a number of distributed (Multi-Agent) systems where proposed for the efficient management of the data to be processed. Each of such approaches focused in different AI paradigms (coordination [21], planning [224], norms [223]). Currently transplant organisations are modernising their software systems to support this distributed environment (for example, the Sintra system in Argentina or in Argentina[2] or the SUIL system in Spain[3]). Hence, sooner rather than latter, transplant professionals will be able to interact efficiently via a distributed software system.

A critical issue not addressed by any of the proposed distributed systems (formal or implemented) but of increasing concern in the transplant community is the important number of available organs for transplantation that are discarded as being deemed non-viable (not suitable) for that purpose. The focus of this transplant research is to extend the donor and organ acceptability criteria so that more organs are accepted as suitable for transplantation. In other words, the propose is to prevent discarding organs that although not *ideal* their discard (unsuitability) cannot be justified. Transplant organisations do revise these acceptability criteria and periodically updates and publish them. However, these criteria rapidly evolve because of the researchers' effort in extending them to reduce organ discards. Hence, the more advanced transplant teams deviate from the basic consented criteria. Put in other words in [238]:

> ...strict adherence to those 'standard donor criteria' resulted in a progredient[4] undersupply of available organs with the result of significantly extended waiting times and increased mortality on the waiting list. [...] Recent evidence confirms that certain donor criteria can be liberalised to increase the available donor pool by accepting 'Marginal Donors' who would, under conventional guidelines, be declined as potential donors.

One of the obstacles for reducing the organ discards is that current organ selection processes do not account for the fact that transplant professionals may disagree. Currently, the decision to offer or not to offer an available organ for transplantation is based exclusively on the assessment of experts located at the donor site, on the basis of whether they believe it to be viable (suitable) for transplantation. Thus, it may be the case that the transplant professionals located at the donor site will not offer an organ because they believe it not to be viable, while transplant professionals responsible of a potential recipient may have deemed the organ viable for their patient and, if given the chance, they may have successfully defend the organ's viability. For that reason we intend to facilitate support for a joint deliberation between donor and recipient agents (respectively representing professionals responsible for

[2] http://sintra.incucai.gov.ar/

[3] http://www.ont.es

[4] progressive

the donor and recipients) so that an organ that would ordinarily be discarded (not offered), because it was deemed non-viable by a donor agent, may now be transplanted if a recipient agent can successfully argue that it is suitable for the particular recipient it represents. When the outcome of a deliberation suggest the transplant to be safe, the organ may then be transplanted, otherwise agents have justified the organ's non-viability and thus the organ should be discarded.

In §3.1.3 we propose an alternative organ selection process where all organs that are made available for transplantation must be offered. It is through deliberation that their suitability for transplantation is decided. This alternative process is formalised as an extension of the agent-based organisation CARREL [224], intended to facilitate the offer and allocation of human organs for transplantation. Thus, in the following subsection we briefly introduce the CARREL system.

3.1.1 CARREL: An Agent based Institution

In line with other distributed systems, CARREL is proposed [224] to ease and partly automate the increasingly complex tasks assigned to the transplant professionals. In short, CARREL is proposed for an efficient management of the data to be processed in carrying out recipient selection, organ and tissue allocation, ensuring adherence to legislation, following approved protocols and preparing delivery plans. In order to perform these tasks CARREL is required to manage and process vast and complex data, as well as to adhere to complex, in some cases conflicting, sets of national and international regulations and protocols governing exchange of organs and tissues. This has motivated development of CARREL as an electronic institution [224]. As such, CARREL encodes sets of legislation and protocols based on two physical institutions representing examples of best practice: the OCATT (Organització CATalana de Trasplantaments)[5] and ONT (Organización Nacional de Transplantes)[6] organ transplantation organisations for Catalonia and Spain respectively. Spain has improved its organ transplant process by creating a model with two organisational levels:

- At the intra-hospital level, a hospitals transplant coordinator coordinates everyone involved in donor procurement, organ allocation, and transplantation.

- At the inter-hospital level, an intermediary organisation (OCATT for just Catalonia and ONT for all of Spain) monitors the communication and coordination of all participating health-care transplant organisations.

Figure 3.1 illustrates CARREL managing the inter-hospital level and shows the entities with which it interacts. OCATT and ONT denote the organ transplantation organisations that own the agent platform and act as observers (they have special access to the agent platform). Each transplant coordination unit (UCT) represents a hospital associated with

[5] http://www10.gencat.net/catsalut/ocatt/en/htm/index.htm

[6] http://www.ont.es

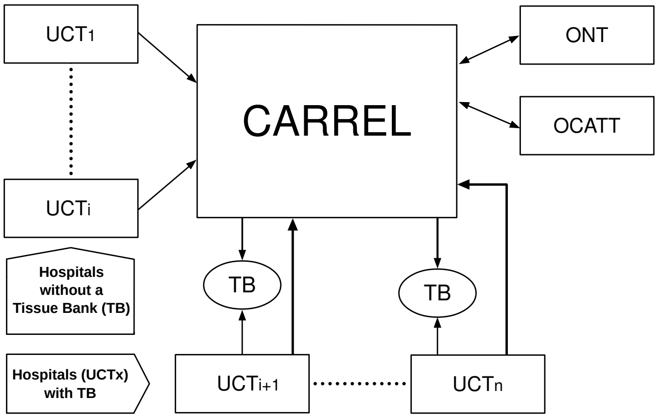


Figure 3.1: The CARREL agent-based institution. The inter-hospital level.

CARREL. The UCTs manage the intra-hospital level; each UCT aims to successfully coordinate organ and tissue procurement, extraction, and implantation, and in turn each is modelled as an agency [74]. CARREL must:

- ensure that all the agents entered into the system follow behavioral norms,
- remain informed of all the recipients registered on the waiting lists,
- check that all hospitals fulfill the requirements needed to interact with CARREL,
- coordinate the organ or tissue delivery from one facility to another, and
- register all incidents relating to a particular organ or tissue.

A hospital must join the CARREL system to use its services. In doing so, the hospital undertakes an obligation to respect the norms ruling CARREL interactions. For example,

- CARREL must process all organ offers and tissue requests,
- hospitals must accept the outcomes of the assignation process, and
- hospitals must update CARREL with any relevant event related to organs and tissues received through CARREL.

CARREL is formalised as a dialogical system in which all interactions are compositions of message exchanges, or illocutions, structured through agent group meetings called scenes or rooms. Each agent can have one or more roles, which define the rooms the agent can enter and the protocols it should follow. So, extending CARREL involves defining new roles or illocutions, where the former might require defining new rooms. For example, the task of offering an organ for transplantation and searching for a recipient is fulfilled by having UCT agents take on the hospital *finder agent* role. Similarly when CARREL contacts a UCT to inform of an available organ suitable for a recipient on that UCTs waiting list, the UCT agent takes on the *hospital contact* agent role. Another role defined by CARREL relevant for the transplant scenario is that of the *hospital information agent*, sent by hospitals to keep the CARREL system updated about any event related to a piece or the state of the waiting lists. For a more detailed description of the roles and scenes we refer the reader to [224] and [14].

In the following subsection we present the current human organ selection as it is defined in Spain and so as it is modelled in CARREL. Once the problem to resolve is highlighted, we present in §3.1.3 an alternative organ selection process that we believe has the potential to increase human organ availability for transplantation.

3.1.2 Human Organ Selection

In Spain, the human-organ selection process begins when a transplant coordinator, represented by a UCTs donor agent[7] (DA), detects a potential donor. After analysing the relevant donors characteristics, the DA informs OCATT (assuming the DA is in Catalonia) of the organs it considers viable for transplantation. Any organ deemed non-viable is simply discarded (in other words, surgeons will not extract it from the donor). If a recipient is found the allocation starts and OCATT offers the organ to a Catalan recipient agent, RA. This agent might or might not accept the organ. It is worth mentioning that, at this stage, the offered organ has not yet been extracted. If the RA refuses the organ, OCATT will offer the organ to other RAs until either one accepts it or all have refused it. The RA that accepts the organ can discard it during or after extraction, in which case it is likely that the organ will not be transplanted. If no RA on Catalonias waiting list accepts the organ, OCATT will offer it to the ONT, and a similar process takes place, this time embracing all of Spain. If all RAs refuse the organ, ONT will offer it to transplant organisation in Europe. If every organisation fails to allocate the organ, the organ will be discarded.

In 2005, OCATT reported that Catalonia, a leader in organ transplantations, discarded approximately 20 percent of livers and kidneys, 60 percent of hearts, 85 percent of lungs, and 95 percent of pancreases [166].It has been acknowledged [143] that these discard rates can be reduced if one accounts for two factors that are currently not taken into account in the current organ selection process: *1)* doctors often disagree as to whether an organ is viable; *2)* organs are rarely viable or non-viable *per se*, but rather assessment of viability should depend on both the donor and potential recipient characteristics, as well as for courses of action to be undertaken during transplantation.

3.1.3 Alternative Human Organ Selection

Human-organ selection illustrates the ubiquity of disagreement and conflict of opinion in the medical domain. What might be sufficient reason for discarding an organ for some qualified professionals might not be for others. Consider a donor with a smoking history of more than 20 to 30 packs of cigarettes per year and no history of chronic obstructive pulmonary disease

[7] Here agent may be artificial or human. For readability reasons the names of the agents are changed from those defined in the UCTs [74].

(*COPD*). Some would cite the donor's smoking history as sufficient reason for deeming the lung non-viable, but others would disagree given theres no history of COPD [143]. Similarly, some would discard the kidney of a donor who died from *streptococcus viridans endocarditis*. Others would reason that by administrating penicillin to the recipient, they could safely transplant the kidney [143]. So, although a DA_i (DA of UCT$_i$) might argue that an organ is non-viable, a RA_j (RA of UCT$_j$) might provide a stronger argument for considering the organ as viable. On the other hand, DA_i might argue that an organ is viable, and this argument might be stronger than an RA_js argument for non-viability, thus making the RA_j reconsider. By enabling supervised and validated deliberation over a human organs viability, the number of discarded organs can *safely* be reduced.

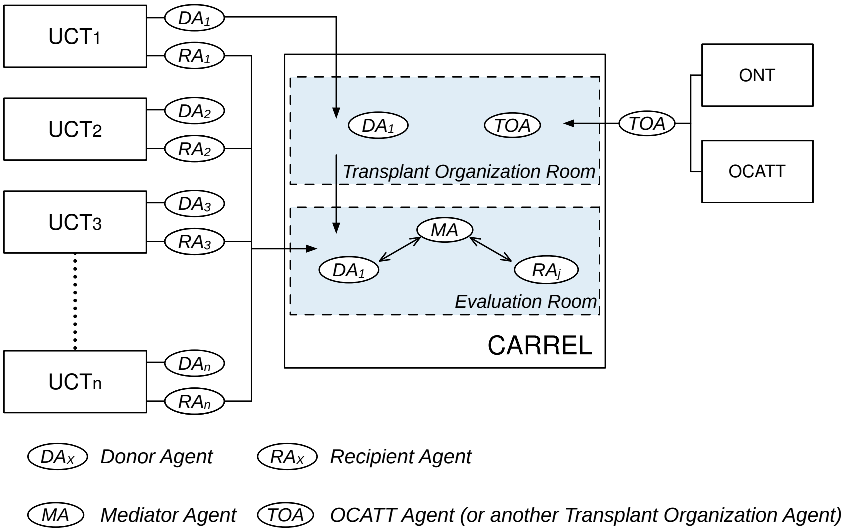


Figure 3.2: CARREL$^+$

For CARREL to implement *ProCLAIM* and thus support the new selection process some changes are required. The main of which is the definition of the MA role for managing the DAs and RAs deliberation dialogue over the safety of the available organs transplant. The deliberation takes place in two new rooms: the *donor evaluation room* and *recipient evaluation room* as depicted in figure 3.3. For simplicity, we refer to these as the *evaluation room*. Other extensions are required to accommodate the new selection process, for example, introducing the transplant organisation agent (TOA) and *transplant organisation room*, which distribute organ offers to the appropriate recipient agents (see figure 3.2). The DA and RA extend the roles of a UCTs *hospital finder agent* and *hospital contact agent*, respectively, to facilitate the submission of arguments relevant to assessing an offered organ's viability in the evaluation room. To report on the outcome of the transplant operation the RA takes the role of the *hospital information agent*. Other changes are required that are not described here as they bring no insight to the implementation of *ProCLAIM*, we refer the reader to [14] for a detailed description of CARREL$^+$.

In the new selection process, having identified a potential donor, DA_i will enter the

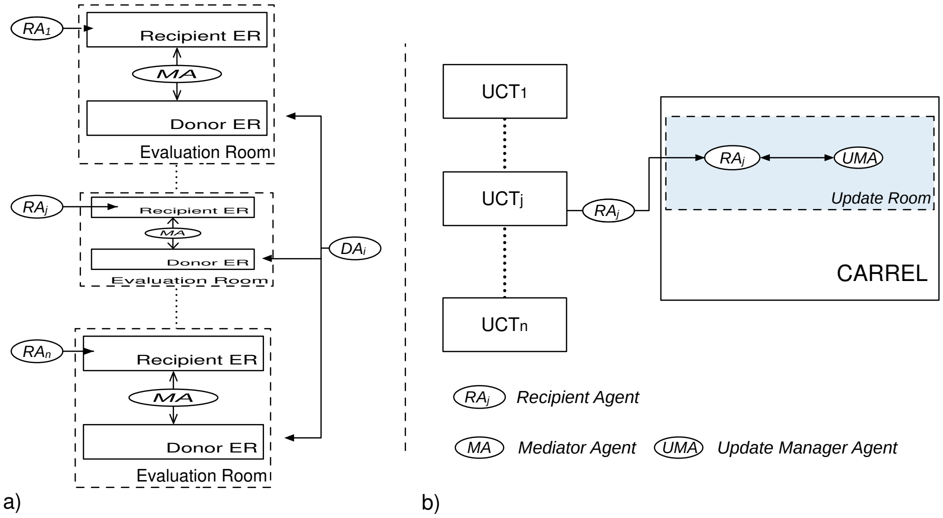


Figure 3.3: CARREL$^+$'s Evaluation Room and Update Room

transplant organisation room and communicate basic organ data (for example, organ type) and donor's data (such as the donors clinical history) to the TOA representing OCATT or ONT. The TOA may delegate the offer to another transplant organisation or pass the offer to the evaluation room, in which case the TOA contacts each RA_j identified as a potential recipient on the basis of basic organ and donor data. The MA in the evaluation room then mediates the exchange of RA_j and DA_i arguments for and against the transplant safety, possibly submitting additional arguments (see figure 3.3). Finally, for each patient represented by the contacted RA_j the MA evaluates the submitted arguments to decide[8] whether it is safe or not to transplant the offered organ, that is, whether the organs are deemed viable or non-viable for transplantation for a particular potential recipient.[9]. Figure 3.4 illustrates the proposed human organ selection process. In particular we can see how organs deemed non-viable by a DA_i may nonetheless be transplanted by a RA_j.

If the transplant is deemed safe (the organ is labelled as viable) for a RA_j and assigned to its potential recipient, the surgeons of the UCT$_j$ will proceed to extract the organ from the donor, at this stage new evidence may indicate the organ is in fact not suitable. Otherwise, the surgeons may proceed to transplant it to their patient, which may turn to be a successful operation, or not. Feedback on the outcome of the operation must be reported to CARREL, even in the case where the organ was deemed unsuitable just after the extraction. This feedback will later allow reusing previous deliberation to solve new cases on evidential basis.

The idea is that in normal circumstances little intervention will be require from trans-

[8] It should be clear that final decisions are always given or at least supervised by human agents.

[9] This selection process should not be confused with the assignation process, in which an available organ is assigned to the most appropriate recipient. An extension to this selection process so that it covers organ assignation is proposed in [8].

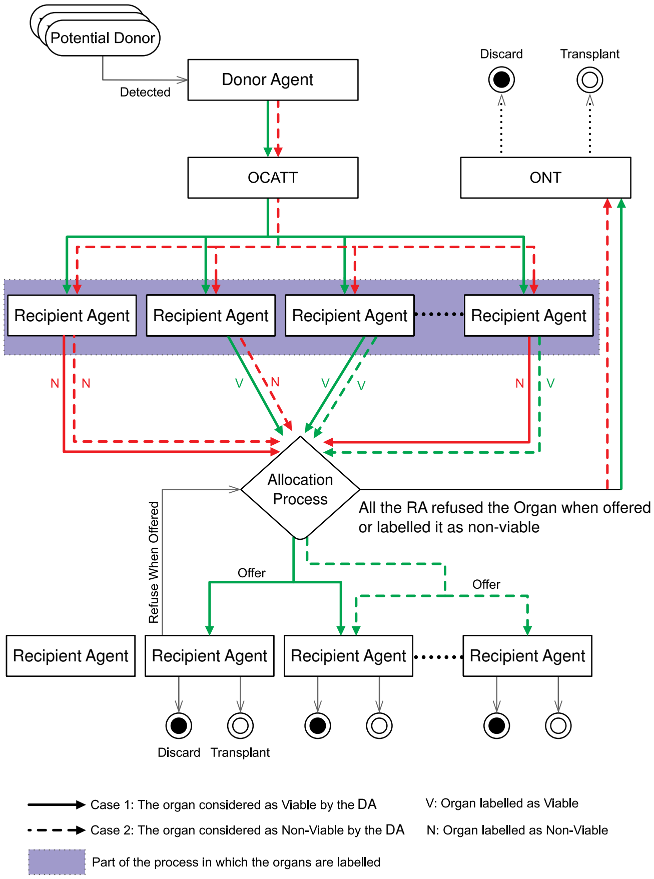

Potential Donor
Detected
Donor Agent
Discard
Transplant
OCATT
ONT
Recipient Agent
Recipient Agent
Recipient Agent
Recipient Agent
N
N
V
N
V
V
N
V
Allocation Process
All the RA refused the Organ when offered or labelled it as non-viable
Refuse When Offered
Offer
Offer
Recipient Agent
Recipient Agent
Recipient Agent
Recipient Agent
Discard
Transplant
Case 1: The organ considered as Viable by the DA
Case 2: The organ considered as Non-Viable by the DA
Part of the process in which the organs are labelled
V: Organ labelled as Viable
N: Organ labelled as Non-Viable

Figure 3.4: Flow of the proposed human organ selection process.

plant professionals in a deliberation. Most of the argument exchange can be automated, and in so far decisions are compliant with guidelines (domain consented knowledge) the professionals intervention may consist in validating the decisions which are given in a form a a tree of arguments and so providing a justification (explanation) for the proposed solution. However, when any of the professionals disagree with the given justification further arguments can be submitted to express this disagreement, which of course can motivate the submission of additional arguments by the interlocutor, thus proceeding with the deliberation which outcome may deviate from standards. Nonetheless, accounting for guidelines, evidence and expert opinion.

The transplant scenario, which consist in formalising the alternative organ selection process, has been developed under supervision of transplant professionals of the Sant Pau Hospital and has been presented in a number of conferences and transplantation events [9, 10, 15] receiving positive feedback from professionals in the field. Throughout this document we will be using this scenario to describe the details of the *ProCLAIM* model. In the following subsection we introduce our second scenario used to test *ProCLAIM*'s generality.

3.2 Wastewater Management in River Basins

Fast demographic growth, together with an increase of industrial activity, historically placed near the riversides to use water as a resource, has created severe ecological imbalances in fluvial ecosystems. These imbalances have been also influenced by the idiosyncrasy of each river basin (altitude, latitude, geology, rainfall regime, hydrological net, *etc*).

According to the European Union Water Framework Directive (WFD) [71] a river basin comprises the area of land from which all surface run-off flows through a sequence of streams, rivers and, possibly, lakes into the sea at a single river mouth, estuary or delta. Thus, river basins are multi-dimensional units and their management surpasses administrative boundaries; their scale of problems increase [211, 54, 173] since the state of rivers in a given point depends on what happens on its surroundings and upper stretches of the river. Hence, the WFD introduces the holistic approach to reveal the major pressures, the impact on the receiving waters and the water resources management at river basin level.

The case study on which this work is based is located in Catalonia, where the majority of river basins have sewer infrastructures and Wastewater Treatment Plants (WWTPs). It is important to highlight that almost all sewer systems are unitary which means pluvial, domestic and even several industrial wastewater streams are collected together, adding another challenge to wastewater management [217]. Peculiarities of Mediterranean rainfall regimes (short duration but intensive rains) [179, 165] together with a high diversification of industries and little integrated wastewater management makes it difficult to reach into a good ecological status of rivers defined by WFD [71]. All these elements (sanitation infrastructures and industries) form a complex system in which several agents come together with different goals and interests difficult to manage as a whole without special methodologies [107].

This has recently motivated exploring multi-agent approach to formalise an integrated management of wastewater in a river basin [39]. This research is undertaken by the Laboratory of Chemical and Environmental Engineering at University of Girona. In this research the *ProCLAIM* model is used to support a collaborative decision making intended to reduce the environmental impact of industrial discharges in emergency situation. The problem is particularly challenging for the variety of actors involved in the process (industries, sewer systems, wastewater treatment plants, storing tanks...) with diverse interests and knowledge about problem. Also challenging because of the variety of external factors relevant for the decision making (*e.g.* meteorological context, mechanical failure in an infrastructure, concurrent wastewater produced by other sources, industrial and non-industrial, etc...).

In normal situation the multi-agent system is intended to facilitate information exchange for planning *normal* (expected) industrial discharges, and so addressing scheduling problems while ensuring guidelines and legislation enactment. However, in emergency situations, for example, when an industry has an abnormal discharge (regarding content of the spill, timing or amount of wastewater), rescheduling may not be enough. It may not be possible to keep the punctual abnormal industrial discharge within legal thresholds. Moreover, although guidelines may provide some indications on what to do in order to reduce the discharge's environmental impact, the special circumstances may motivate alternative decisions justified as safer.

3.2.1 Towards a Multi-Agent System for Integrated Management of Urban Wastewater Systems

A simplified version of the wastewater system under study is depicted in figure 3.5 identifying the main actors involved in an industrial spill. In this figure we can see that the normal course of a spill, highlighted with a thicker line, begins with industry that produces a discharge and spills it into the sewer system, the sewer system conducts the discharge to the closest WWTP that, after properly treating the wastewater, discharges it into the final receiving media, in this case the river. The figure also illustrates different alternative courses for the discharges. For example, temporally storing a wastewater in a tank may help regulating the timing and amount of wastewater to be discharged. Also a WWTP that cannot treat a wastewater may bypass the discharge to another nearby WWTP for it to deal with discharge.

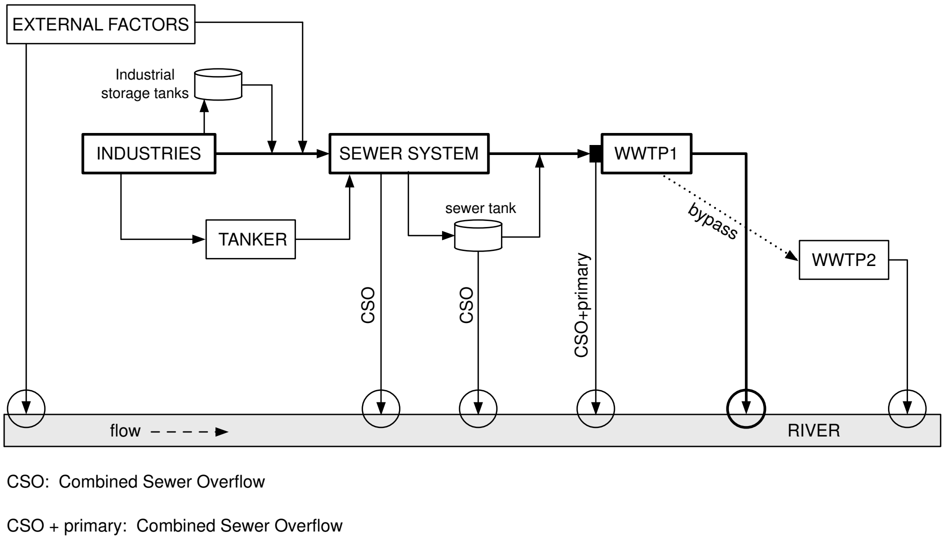


Figure 3.5: The Wastewater System.

The multi-agent system proposed in [39], defines a number of agents which main purpose is to jointly cooperate in order to optimise infrastructure operations while minimising the environmental impact of industrial discharges in emergency situation. Among the defined agents we find:

- **Industry Agent (InA)**: represents individual industries and/or groups of industries that need to manage their produced wastewater as a result of their production process.

InA discharge their produced wastewater into the sewer system, where it is collected together with other inflows and transported to the WWTP.

- **Industrial Tank Agent (ITA)**: represents the infrastructure available to store industrial wastewater, with the function to contain and laminate wastewater towards the sewer net.

- **Sewer Agent (SA)**: represents the sewer infrastructure that is mainly responsible of collecting and distributing wastewater (domestic and industrial), together with the collected rainfall, to the WWTP.

- **Wastewater Treatment Agent (WTA)**: represents the manager of WWTP. Its main function is to keep track of wastewater flow arriving at WWTP as well as to supervise and control the treatment process. It gives the convenient alarms when necessary and the orders to change the operational set points. This responsibility is shared between the managers of WWTP (WTA_M) and the operators (WTA_O).

- **River Consortium Agent (RCA)**: represents the maximum authority in the catchment, whose main objective is to preserve the river quality. Its main functions are to manage and coordinate a group of WWTPs in the river catchment as well as to monitor river quality and to prevent possible hazardous contamination by supervising InA and WTA.

- **Household Agent (HA)**:represents a simple information carrying agent that supplies the domestic wastewater discharge data (domestic wastewater production).

- **Meteorologist Agent ($MetA$)**: represents weather conditions and holds data from rainfalls events when occurring (intensity and duration of the event).

When an emergency situation is presented, these agents must collaborate in order to prevent undertaking actions that may cause severe undesirable side effects. This decision making process is formalised using the *ProCLAIM* model.

3.2.2 Reasoning About Safety of an Industrial Discharge

In industrialised areas where industrial discharges are connected to the sewer system and finally treated by the WWTP (together with domestic wastewater and rainfall), industrial discharges represent an important load contribution to the Urban Wastewater System (UWS). Several types of industrial discharges with different characteristics (*e.g.* content of organic matter, nutrients and/or presence of pollutants) can affect the growth of micro-organisms into the WWTP and so the WWTP treatment operation and the final result. For that reason, there exists an important body of research intended to improve and increase the knowledge on WWTP operational problems related to influent discharges (*e.g.* [69, 70, 117, 235, 112]). Typically, the representation of this knowledge, based on on-line and off-line data as well as the experts' heuristics, is organised and formalised by means of decision trees and/or knowledge-based flow diagrams (*e.g.* [203, 197, 72]). That is, the knowledge is organised hierarchically by means of top-down descriptions of the interactions between the different parameters and factors used to solve a problem. This representation allows an easy interpretation of the available knowledge, mostly, in terms of cause-effect relations for a concrete problem.

These approaches typically develop their knowledge exploring common benchmark problems with the result of an increasing understanding of such stereotypical scenarios. However, because of the high diversification of industries (*e.g.* long and short-term variations), it is difficult to define *typical industrial operating conditions* and abstract from external factors such as weather conditions and/or other urban wastewater discharges. As a result, WWTP managers are left with little decision support when confronting situation that deviate from these benchmark problems. In particular, it is not easy to alter, on the fly, decision trees in order to adapt them to alternative situations (*e.g.* to express a cause-effect relation among diverse factors commonly treated independently).

Guidelines and regulations do exists to protect WWTP from hazardous industrial discharges that can cause operational problems to the WWTP. Such regulations are currently based on the application of discharge standards to point sources defining the permitted quality of wastewater discharged. The description of these standards is made by means of numerical limits (*i.e.* thresholds) for a set of polluting parameters indicating a concentration and/or load. Such numerical limits are defined with independence from the particular situation in which the industrial spill is intended, thus ignoring the WWTP particular state and characteristics as well as any external factor that may affect either the spill or the WWTP. However, the fact is that, special circumstances may sometimes motivate either to reject discharges that are under legal limits (*e.g.* the WWTP is overloaded) in order to prevent potential complications; or to accept discharges that are above legal thresholds since, for example, weather condition (*e.g.* rain may dilute the concentration of a toxic) permits WWTP to safely deal with the industrial spill and, in so doing, the use of the infrastructure is optimised.

This situation suggests the need for a more flexible decision support mechanism in order to successfully adapt WWTP operation to influent variability and avoid and/or mitigate operational problems into WWTP. This flexible decision support is provided by the *ProCLAIM* model. Where, in the same spirit as in the transplant scenario, great part of the decision making can be automated by the artificial agents that explore the alternative options for dealing with the industrial discharge. If a course of action is accepted by all affected agents as being safe, (*i.e.* it is validated by the experts and compliant with guidelines), the wastewater can safely be discharged. However, if no safe course of action is found, experts must then take a more active role in the deliberation, possibly proposing solutions that deviate from guidelines and that may disagree with other experts. Note that the industrial wastewater must be treated eventually, even all alternatives will cause some undesirable side effect. The question will then be, *how* which is the course of action less harmful for the WWTP and, more importantly, for the fluvial ecosystem.

To illustrate a possible situation where the participation of a number of agents of the system is required for a collaborative decision making, suppose an industry arrives to an emergency situation and proposes discharging an abnormal wastewater into WWTP1 (see figure 3.5). We can think of this action proposal formalised as an argument $A1$, as depicted

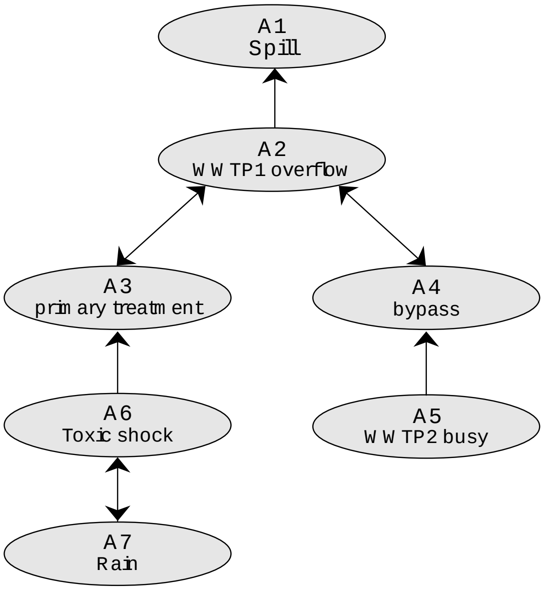


Figure 3.6: Tree of interacting arguments which result from deciding the safety of an industrial spill.

in figure 3.6. Suppose the sewer system can cope with the discharge but not the WWTP1. So the $WTA1$ (the Wastewater Treatment Agent of WWTP1) may argue against $A1$ indicating, with argument $A2$ that the discharge may cause a critical situation to the plant (*e.g.* an overflow, bulking or foaming). The $WTA1$ himself may propose to either bypass the discharge to WWTP2 (argument $A3$) or to only make a primary (partial) treatment to the wastewater ($A4$). Now, the $WAT2$ (the Wastewater Treatment Agent of WWTP2) may then attack argument $A3$ with argument $A5$ indicating that it is expecting another wastewater discharge, incompatible with the spill WTA_M1 proposed to bypass to WWTP2. On the other hand the RCA may argue against $A4$, with argument $A6$ deeming that the primary treatment is not enough for mitigating the impact of the discharge in the river. Suppose as well that the $MetA$ informs of the proximity of a hard rainfall, the WTA_M1 may then use this information to argue ($A7$) that given that hard rainfall is expected, the harmful substances in the industrial discharge will be sufficiently diluted so as to minimise the discharge's impact on the river.

Thus *ProCLAIM* is used to provide a setting where these agents can effectively and efficiently deliberate over the spill's safety. Furthermore, these arguments, arranged as a tree of arguments (as depicted in Fig. 3.6), are evaluated by *ProCLAIM*'s defined knowledge resources so as to suggest a possible solution. Thus, for example, if arguments $A3$ and $A4$ are taken as preferred to $A2$, namely, the operational problems on the WWTP1 can be prevented by either bypassing the spill or by performing only a primary treatment to the spill, then, argument $A1$ will be justified if and only if $A7$ is deemed preferred to $A6$. That is, the spill will be deemed safe if it is believed that the rainfall, together with the WWTP1's primary treatment, will effectively reduces the environmental impact of the industrial spill.

3.3 Discussion

In this chapter two safety critical scenarios were presented in which, supported by a distributed (multi-agent) system, a deliberation among experts is of great value. In the transplant scenario it may help combating the scarcity of human organs and in the environmental scenario it may help mitigating the environmental impact of industrial discharges. For their critical nature, the requirements for the addressed scenarios are very strong: deliberations must be: highly efficient, focused, non-disruptive, unambiguous, provide a clear representation of the problem and must promote safe decisions, in other words, final decisions must account for all the relevant factors for the decision making, and the proposal must be evaluated on the basis of reliable knowledge sources.

Within the transplant scenario a number of prototypes were developed by the author in order to address the many aspects of *ProCLAIM*, the most developed prototype was successfully presented as the large scale demonstrator of de FP6 EU-Project ASPIC[10](Argumentation Service Platform with Integrated Components) in which the functionality of argument-based components (inference and dialogue) were tested. This prototype presented in [11] will be described in detail in §10.1. As with respect to acceptance of the proposal form the medical community, the gathered feedback is very promising. A number of our papers were presented in transplant conferences [9, 10, 15] and we obtained positive response from the interaction with transplant professionals, particularly with transplant professionals of the Sant Pau Hospital and members of the Argentinean transplant organisation (INCUCAI [11]).

While the transplant scenario resulted in a very fruitful test case for the use of *ProCLAIM*, the environmental scenario is yet in a premature research phase. Nonetheless, it has thrown light on a number of issues worth addressing in *ProCLAIM* that were undetected when applied to the transplant scenario, we discuss these aspects in §11. Furthermore, as mentioned above, the environmental scenario is being developed by members of the Laboratory of Chemical and Environmental Engineering of the University of Girona, as chemical engineers they are not experts in argumentation, hence their feedback in applying *ProCLAIM* is very useful for a identifying possible difficulties in implementing *ProCLAIM* in a new scenario.

To conclude, it is worth explaining another important difference between the two case studies. The transplant case addressed a relatively well understood problem in that: *1)* the proposal extends an already defined multi-agent system, *2)* papers in the transplant domain provides different arguments for deeming organs as viable or not, hence while the deliberation approach is not proposed, the type of arguments relevant for such deliberation can be, in part, derived from those used in the academical papers (and later contrasted with expert opinions). In the environmental scenario however, *1)* no mature, multi-agent approach exists for an integrated management of urban wastewater system, so *ProCLAIM* has been adapted to the many stages of the multi-agent system formulation. And *2)* nowhere could

[10] http://www.argumentation.org/carrel.htm

[11] http://incucai.gov.ar/

we find well organised arguments addressing the problem the environmental scenario aims to solve. The main reason for this, is that the environmental scenario is addressing a completely novel situation. As in practice, there is no holistic, mature, view nor implementation of the management of a wastewater system[12]. Each actor is taken as independent and so, their task is to perform individually within legal limits (thresholds). This has a clear problem when one of the actor can not comply arriving to an emergency situation, the problem can propagate with no control.

Therefore, while the transplant scenario was used to define the many features of *ProCLAIM*, in the environmental scenario the *ProCLAIM* model is used to better understand an ill-defined problem. In so doing important feedback is gained for better defining *ProCLAIM* and understating its scope of applicability. These aspects are discussed in §11.

We now proceed to introduce the *ProCLAIM* model.

[12] In the transplant scenario this holistic view is given by the national and international transplant organisations.

Chapter 4

ProCLAIM's Framework

The *ProCLAIM* model is intended to assist developers in extending multi-agent systems[1] so that they support deliberation dialogues among agents for deciding whether a proposed action is safe. Therefore, with *ProCLAIM* we intend to provide an abstract model for deliberation dialogues that can be instantiated in a family of problems; and procedures to facilitate its application in new scenarios. In this chapter we present an overview of *ProCLAIM*'s framework. Subsequent chapters will focus on different aspects of this framework: §5 introduces the model's dialogue game that governs the agents' overall interaction; §7 focuses on the agents' argumentation process on the more representational level, while in §7 we put the agents' argumentation into practice; and in §8 we discuss the arguments' evaluation and *ProCLAIM*'s solution proposal.

ProCLAIM can be regarded as defining a centralised medium through which heterogeneous agents can effectively deliberate. This medium is tailored for the deliberation purpose and is intended to focus the deliberation on the relevant matters to be discussed keeping track of the participants' submitted arguments and evaluate them to propose a solution to the addressed problem. This centralised medium is embodied by a Mediator Agent (MA) which role ensure the success of the deliberation process, enabled by the MA's access to a number of knowledge resources, that all together conform the *ProCLAIM*'s architecture.

A deliberation in *ProCLAIM* starts with the submission of an argument proposing an action, then participants of the deliberation, taken as experts in the domain, submit further arguments that attack or defend the appropriateness of the proposal on the basis of whether they believe the action is safe or not. What defines to great extend *ProCLAIM* is the role of the MA in managing the deliberation. The MA can be regarded as a *proactive blackboard* where agents submit their arguments, the MA organises them as a tree of interacting arguments, accepting those which are relevant and rejecting those which are not. Furthermore, the MA guides the participants indicating them which arguments (argument schemes) they can use to attack or defend other arguments in the tree of arguments. Moreover, the MA may also submit additional arguments and when all participants have submitted their arguments the MA proposes a solution to the deliberation.

[1] By multi-agent system we understand any distributed system where entities interact. Thus, the distributed system can be implemented in alternative ways, *e.g.* using web services.

In the following section the MA's role is defined as a way to introduce to *ProCLAIM*'s components which define its architecture. Then in §4.2 the model's architecture is instantiated in the transplant and environmental scenario as a means to illustrate the use of the model's components.

4.1 *ProCLAIM*'s Mediator Agent

The setting *ProCLAIM* provides for participant agents to efficiently deliberate is best described by defining the mediator agent's tasks:

- **Direct participants** on what argument-based moves (argument schemes or critical questions) they can submit at each stage of the deliberation. Thus, for each argument a participant wants to reply to, she is given a set of schemes that she can instantiate and submit as a valid attack, in so far as the instantiation is appropriate. A participant may also challenge some of the submitted arguments and, in turn, a participant may answer to these challenges with the instantiation of the appropriate argument scheme.

- **Validate the incoming arguments** in order to exclude arguments that may jeopardise or disrupt the course of the deliberation. While agents are given the schemes to instantiate, there is no guarantee that the instantiation will be relevant for the discussion. Thus, one of MA's tasks is to discern between relevant and non-relevant instantiations, so as to keep the deliberation highly focused on only the important matters. Each validated argument and challenge is added to a tree of interacting arguments whose root is the argument proposing the initial action.

- **Submit additional arguments** that introduce new factors not taken into account by the participants but that either guidelines and/or evidence associated with previous similar cases indicate as relevant. Thus, for example, if α is taken as a fact, and guidelines indicate that α is a contraindication for performing the proposed action, but, for some reason no participant highlights this, the MA will submit an argument against the action proposal indicating that there is a contraindication α for its performance. This argument will be added to the tree of interacting arguments.

- **Evaluate the tree of interacting arguments** so as to propose a solution to whether the proposed action is safe or not. A solution is proposed by means of assigning a preference between conflicting (mutually attacking) arguments and then evaluating the justified arguments as defined by Dung's theory.

In order to perform the above introduced tasks, the MA references four knowledge resources, as shown diagrammatically in Figure 4.1 and also described below:

Argument Scheme Repository (ASR): In order to direct the participant agents in the submission and exchange of arguments, the MA references a repository of argument schemes and their associated critical questions. The schemes and critical questions are instantiated by agents to construct arguments, and effectively encode the full 'space of argumentation', *i.e.*, all possible lines of reasoning that should be pursued w.r.t a given issue. The repository is structured in such a way that it defines the protocol-based exchange of arguments. Thus, given an argument (that instantiates a scheme in ASR) the repository returns the schemes that agents can instantiate in its reply (as well as the critical questions used to challenge it).

Domain Consented Knowledge (DCK): This component enables the MA to check whether the arguments comply with the established knowledge, by checking what the valid instantiations of the schemes in ASR are (the ASR can thus be regarded as an abstraction of the DCK). This is of particular importance in safety critical domains where *1)* one is under extra obligation to ensure that spurious instantiations of argument schemes should not bear on the outcome of any deliberation; and *2)* guidelines usually exist in such domains and so should be taken into account when evaluating the submitted arguments. The MA also references the DCK in order to check whether any known factor is not being addressed by the participants (experts) in spite of being deemed relevant from the view point of the guidelines. In such a case, the MA uses the DCK in order to submit additional arguments, which account for these neglected, but relevant, factors. In this last sense, the MA can be regarded as a participant expert in guidelines.

Case-Based Reasoning Component (CBRc): This component enables the MA to assign a preference relation between mutually attacking arguments (*i.e.* resolve conflicts amongst pairs of arguments) on the basis of their associated evidence gathered from past deliberations. The CBRc may also provide additional arguments that were deemed relevant in previous similar situations and are applicable in the current target problem. Again, in this last sense, the MA plays the role of an expert or specialist in collecting evidence from previous deliberations.

Argument Endorsement Manager (AEM): Depending on who endorses an argument, the strengths of arguments may be readjusted by the MA. Thus, this component manages the knowledge related to, for example, the agents' roles and/or reputations.

A deliberation dialogue begins with one of the agents[2] submitting an argument proposing an action, the MA will then guide the proponent agents in the submission of further arguments that will attack or defend the justification given for the proposed action. Each submitted argument (or challenge) instantiates a scheme (or critical question) of the ASR. Hence the MA references the ASR in order to indicate which are the schemes or critical questions they can instantiate in replay to each of the submitted arguments or challenges. These schemes and critical questions are specific to the application at hand (*e.g.* the transplant or environmental scenario). Thus, for example, in the transplant scenario, if the Recipient Agent (RA) were to argue against the safety of an organ transplant the MA will provide him with a series of schemes and critical questions encoded in the ASR, among which may be the scheme:

[2] This agent may well be the MA if the action is a default one.

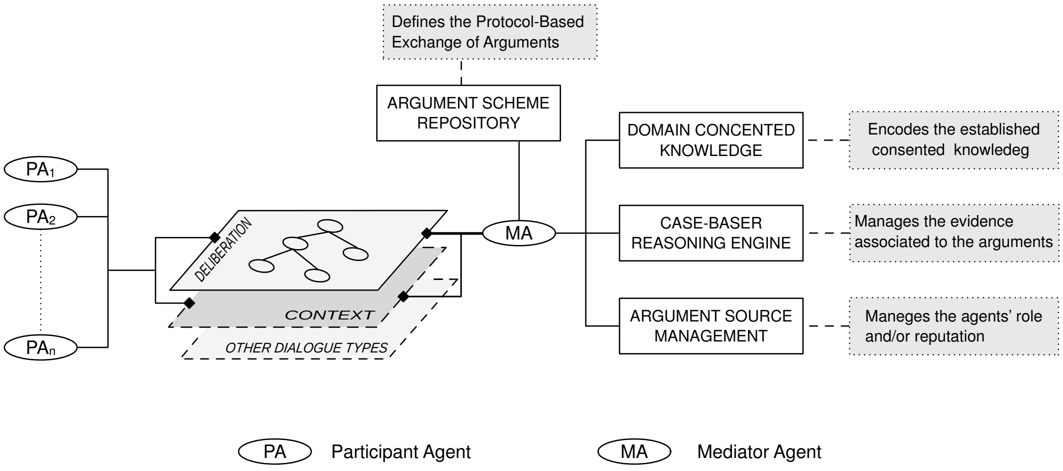

Defines the Protocol-Based Exchange of Arguments
ARGUMENT SCHEME REPOSITORY
DOMAIN CONCENTED KNOWLEDGE
Encodes the established consented knowledeg
CASE-BASER REASONING ENGINE
Manages the evidence associated to the arguments
ARGUMENT SOURCE MANAGEMENT
Maneges the agents' role and/or reputation
PA1
PA2
PAn
MA
DELIBERATION
CONTEXT
OTHER DIALOGUE TYPES
PA Participant Agent
MA Mediator Agent

Figure 4.1: *ProCLAIM*'s architecture.

–The donor's ***C*** *will cause a graft failure.–*

And so, by replacing the variable C with the appropriate values (*e.g.* `smokinh_history`, if the transplanted organ is a lung, which should be known from the context) the RA may submit the desired argument. Although this scheme is formally wrong (*e.g.* the donor's condition C does not*cause*, but it is the reason why the transplant will cause a graft failure) it successfully convey the right reasoning in a succinct and natural way. Furthermore, it does not identify the donor, the recipient and the organ in question which are necesary for the argument but can clearly be known by the participant human agent by the context, this is further discussed in §7.

Each submitted argument, if legal (instantiates an appropriate scheme), is evaluated by the MA in order to determine whether the instantiation of the scheme is a valid one. This is done by the MA referencing the DCK, CBRc and AEM. If an argument is compliant with guidelines, *i.e.* validated by the DCK, the argument is accepted and added to the **tree of interacting argument**, which we denote as $\mathbb{T}$. If the incoming argument is not validated by the DCK it may still be accepted if either the CBRc indicates the argument has previously been accepted or the AEM indicates that the submitter is sufficiently reliable so as to exceptionably accept this argument. In either case the argument is added to the tree of arguments $\mathbb{T}$ and the MA broadcasts this new argument to all participants together with the schemes they can instantiate in reply. If the argument is not accepted by the knowledge resources, the MA informs the submitter of the reasons for it being rejected. Other approaches have considered the relevance of an argument in terms of its impact, or potential impact, on the dialectical status of acceptability of the already posed arguments [176, 170, 148]. Broadly speaking, if an argument to submit will have little or no effect on the acceptability status of the existing arguments, it may be deemed irrelevant and so may not be submitted[3]. This

[3] Also interesting is the work presented in [67], where a pruning of which arguments to account for is made in order to compute dialectical trees efficiently.

approach to the relevance of an argument could be used in *ProCLAIM* for example to stop the deliberation once it is clear that there are enough arguments to deem the proposed action unsafe.

The agents' interaction is governed by a dialogue game which we describe in detail in §5. This dialogue game is defined in three interaction levels: *1)* On the deepest level there is the content of the message, *e.g.*, the submitted arguments. *2)* Each of these messages is wrapped in the appropriate deliberation locution defined by the dialogue game (*e.g.* an argument is wrapped in an *argue* locution); and finally *3)* each of these deliberation locutions is wrapped in either an *inform* or *request* locution. This is because all agents' interaction is mediated by the MA through `inform-request` messages. For instance, arguments are submitted by PAs through `request` locutions and with an `inform` locution the MA informs of their acceptance or rejection. At the intermediate level the dialogue game is organised in six dialogue stages: *Open Stage* in which the dialogue is opened and agents can enter into the dialogue; *Contexts Stage* in which agents submit all the facts and actions they believe to be potentially relevant for the decision making; *Argumentation Stage* where agents submit their arguments and challenges for or against the action's safety; *Endorsement Stage* in which the agents inform of the arguments in $\mathbb{T}$ they endorse; and finally at the *Resolution Stage* the deliberation is resolved. An *Inform Stage* is defined for the MA to provide, upon request, information to the PAs on the deliberation stage (*e.g.* which arguments or set of facts were submitted so far in the deliberation).

Figure 4.1 depicts different dialogue stages as different layer. The first and most important in *ProCLAIM* is the argumentation stage in which the tree of interacting arguments is constructed. Below is the context layer in which PAs submit the available knowledge they believe to be potentially relevant. In so doing, they construct the context in which the decision is undertaken.

Once all participants have submitted their arguments they may move to the *Resolution Stage*, where $\mathbb{T}$ is evaluated by the MA. Firstly, the MA may submit additional arguments to $\mathbb{T}$ deemed relevant from the viewpoint of guidelines and past recorded cases. To do so, the MA references the DCK and the CBRc. And thus, we could consider the MA as playing the role of two additional PAs: an expert or specialist in domain consented knowledge, and another specialist in reusing evidence collected from past deliberations. Secondly, recall that $\mathbb{T}$ may contain arguments that mutually attack each other preventing a definitive solution. Hence, in order to evaluate $\mathbb{T}$ MA has to assign a preference relation between the mutually attacking arguments, and so change the symmetric attacks into asymmetric ones (recall that these surviving successful attacks are often called defeats in the literature). Once this is done, MA applies Dung's evaluation of the justified arguments (under the grounded semantics) to propose a solution (see Figure 4.2). In order to assign this preference between mutually attacking arguments the MA again references the DCK, the CBRc and the AEM. From each resource the MA derives a preference assignment. These may all be in agreement (*e.g.* $A3$ preferred to $A2$) or not, *i.e.* some prefer one argument while another knowledge resource prefers the other argument. The MA's task, is to provide a solution that accounts for each of the knowledge resources' preference assignment. So a solution in which not all resources agree could be of the type: -While guidelines indicate that $S2$ is not a solution to problem $P2$, trustworthy experts argue that $S2$ is a solution to $P2$ and

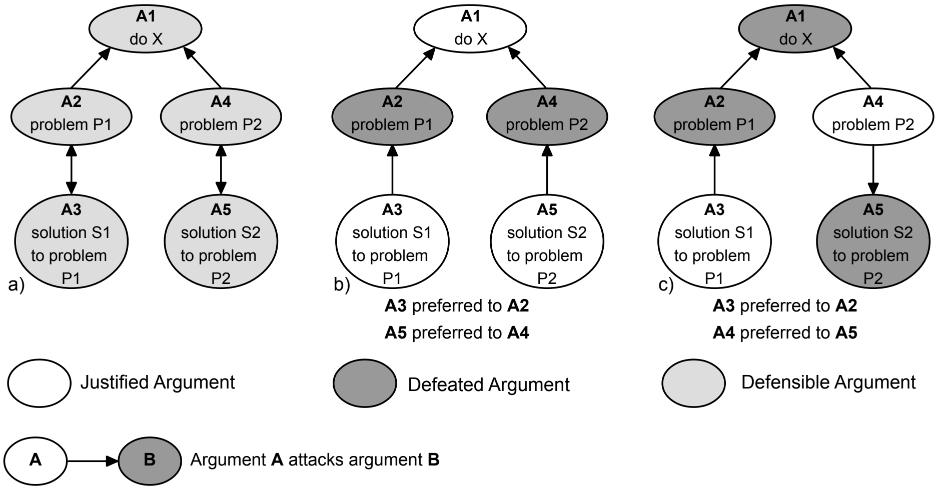


Figure 4.2: Resolving a tree of interacting arguments in order to decide whether or not to perform X. In figure *a)* no solution can be proposed since it is still undecided as to whether the respective solutions address the respective problems, as indicate by the symmetric attacks between A2 and A3, and A4 and A5. In figures *b)* and *c)* the solutions are, respectively, to perform action X and not to, depending on the arguments' preference assignment.

this position is weakly backed up by evidence. On the basis of this information, the person responsible for the final decision will decide whether or not to perform action X.

Eventually a decision has to be taken on whether or not to perform the safety critical action. If the action is deemed safe the action would be performed, unless new information is made available that suggest otherwise. If the action is ultimately undertaken, it may indeed turn out to be safe, or else it may cause undesirable side effects. In *ProCLAIM*, PAs involved in the execution of the safety critical actions must update $\mathbb{T}$ to capture the actual outcome of the performed actions. If an action is satisfactorily performed $\mathbb{T}$ may require no further update. However, if the performed action brought about undesirable side effects this information should be fed back into $\mathbb{T}$ by the appropriate PAs. In this way, a $\mathbb{T}$ associated with a performed action encodes the reasons why the action is or is not safe. Once a tree of arguments is updated, and thus encoding the evidence for which the action is safe or not, it is retained in the Case-Base. As discuused in §9 the CBRc will reuse these updated $\mathbb{T}$s to resolve future similar cases on an evidential basis.

It should be noted that due to the critical nature of the intended scenarios, *ProCLAIM* assumes a rather regulated environment. In particular, *ProCLAIM* does not address any of the normative aspects that would naturally be associated with a safety critical environment. It also assumes that issues such as information privacy, or foreign attacks from malicious agents are also resolved. A good example of the context in which *ProCLAIM* can be used is the transplant scenario within the CARREL system [224] we introduced in the previous chapter. Before we dive into the details of *ProCLAIM* formalisation we outline in the following section the instantiation of *ProCLAIM* in the transplant (§4.2.1) the environmental scenario (§4.2.2).

4.2 Applying the *ProCLAIM* model

Form the viewpoint of the framework presented above, applying *ProCLAIM* in a new scenario involves:

- Defining which are the dominant questions, what is to be argued about.
- Defining which are the participant agents;
- Implementing the DCK, that is, a knowledge base that encodes the domain consented knowledge: guidelines, regulations and legislation governing the domain;
- Identifying the reasoning patterns typical of the domain which will help construct the ASR;
- Instantiating the CBRc, which in great part is done when constructing the ASR, this is we will discuss in the detail in§9.
- Instantiating the AEM. This involves identifying which characteristics of the participant agents motivates increasing or decreasing the confidence in their endorsed arguments. This confidence may vary depending on the domain area the arguments address.

The following two subsections outline the instantiation of *ProCLAIM* in our two case studies.

4.2.1 Applying *ProCLAIM* in the transplant scenario

The dominant question in the transplant scenario is whether an available organ for transplantation can safely be transplanted to a given potential recipient. To address this question an argument proposing the transplant (argument $A1$ in fig. 4.4) is submitted to initiate the deliberation, as illustrated in figure 4.4. As discussed in §3.1.3 the agents involved in the deliberation over the transplant safety are a donor agent (DA), representing the transplant unit responsible for the potential donor, and a recipient agent (RA), representing the transplant unit responsible for the potential recipient, see Figure 4.3. The final decision in this deliberation is primarily evaluated by the DCK. In this scenario the DCK encodes the donor and organ acceptability criteria together with the consented complementary procedures proposed for a successful transplant. For example, if a condition of the potential donor is believed to be a contraindication for being a donor (*e.g. Hepatitis C*, argument $A2$ of Figure 4.4) then the DCK would validate an argument against the transplant safety on the basis of such condition. Currently, while some professionals [143] believe that if the potential recipient also has *Hepatitis C* the organ can safely be transplanted, other professionals

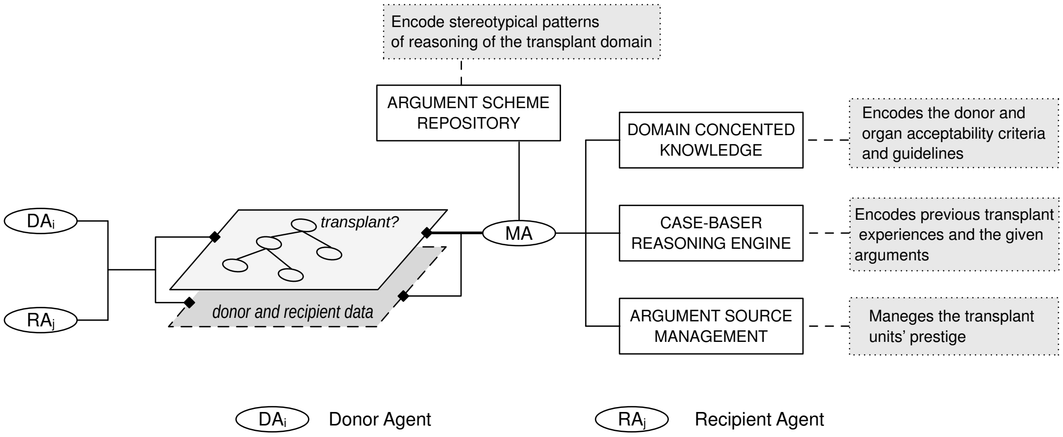

Encode stereotypical patterns
of reasoning of the transplant domain
ARGUMENT SCHEME REPOSITORY
DOMAIN CONCENTED KNOWLEDGE
Encodes the donor and organ acceptability criteria and guidelines
DA$_i$
transplant?
MA
CASE-BASER REASONING ENGINE
Encodes previous transplant experiences and the given arguments
RA$_j$
donor and recipient data
ARGUMENT SOURCE MANAGEMENT
Maneges the transplant units' prestige
DA$_i$ Donor Agent
RA$_j$ Recipient Agent

Figure 4.3: *ProCLAIM*'s architecture in the transplant scenario.

disagree [238]. Thus, while the DCK will validate argument $A3$, as it is relevant argument (it makes sense), it is not so clear whether it should deem $A3$ as preferred to $A2$.

The ASR encodes the schemes and critical question that help eliciting from the DA and RA, step by step, the relevant factors for the decision making. These factors involve in great part the relevant donor and recipient properties as well as the complementary courses of actions that can be performed to ensure a safe transplant operation. Also these factors may involve logistical information required to ensure that the organ can arrive to destination within a safe time span.

All deliberations in the transplant scenario begin with the instantiation of the following scheme:

Given an available organ O *of a donor* D *and potential recipient* R*,* O *should be transplanted to* R*.*

Note that as soon as an organ is made available for a potential recipient, variables O, D and R can be instantiated, and so the deliberation can begin. Recall, that the most basic matching requirements (the organ being of the type needed by the patient) are addressed in the Transplant Organisation Room (see §3.1.3). To this scheme are associated CQs that question the transplant safety. For example, whether the donor has any contraindication. A scheme that embodies this CQ is:

–The donor's $C1$ will cause $C2$ to the recipient, which is a $severe\ infection$–

And so, if both C1 and C2 are instantiated as hepatitis_c we would obtain argument $A2$ of figure 4.4. In reply to this argument CQs will guide PAs to explore possible reasons for reconsidering the action safety, *e.g.* whether there is a course of action that can prevent the viral infection, or whether for this recipient Hepatitis C is not an undesirable side effect of the transplant. This would indeed be the case if the recipient would already have Hepatitis C as argued with $A3$. In §7 we describe the construction of the ASR and the MA's role in guiding the PAs throughout the deliberation.

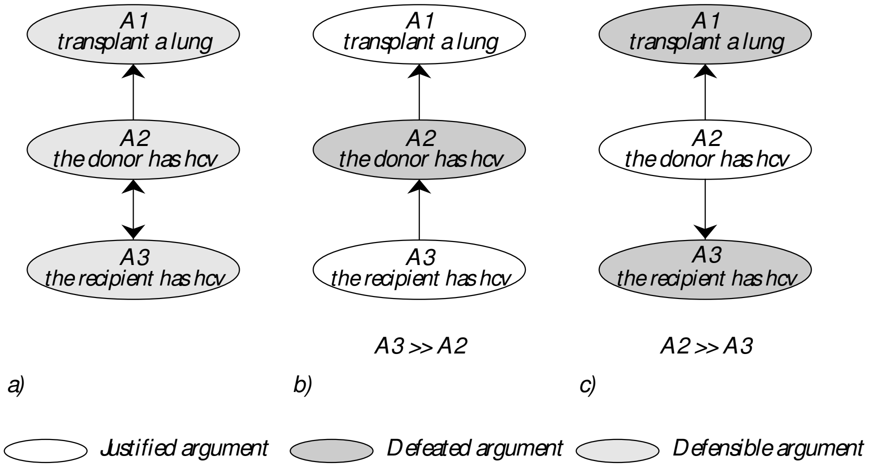


Figure 4.4: Tree of interacting arguments for deciding the safety of a transplant of an organ of a donor with Hepatitis C.

The remaining components to be instantiated are the CBRc and the AEM. The former will help deciding the safety of a transplant operation by reusing similar previous deliberation, while the latter will enable assigning a preference between arguments on the basis of the agents that endorse them. The CBRc, described in §9, is mostly instantiated when building the ASR. This is because the Case-Base is organised in terms of the ASR's structure, which allows a broad comparison between cases, where cases that used the same schemes in the argumentation (they have used the same scenario specific reasoning patterns) are broadly similar.

In §9 we show how the CBRc requires the definition of an ontology that organises the terms and concepts used in the scenario application. This will allow a more fine grained comparison between cases, where cases that used the same schemes and were instantiated with similar terms of the defined ontology can then be deemed similar. The instantiation of the AEM, on the other hand, involves valuating the different transplant units' prestige, based on their rate of success and also favouring those transplant unit that promote the use of marginal organs. Thus the AEM will positively bias the arguments endorsed by prestigious transplant units as their assessment is deemed more reliable.

4.2.2 Applying *ProCLAIM* in the environmental scenario

The dominant question in the environmental scenario is –whether an industrial spill is environmentally safe?– where the answer is not intrinsic to the substances and amount of the wastewater discharge, but to whether the wastewater system as a whole can assimilate the spill without deteriorating the ecosystem. The deliberation thus require the participation of all the agents that represent the infrastructure or media that may affect or be affected by the industrial spill. Therefore, depending on each particular circumstances different agents may be required in the deliberation. In §3.2.1 we have introduced potential participants, such as the Industry Agent (InA), the Wastewater Treatment Agent (WTA) or the River Consortium Agent (RCA). Some deliberations may require the participation of several instances of the same agent type. For example, two or more Industry Agents when concurrent spills need to be addressed, or as illustrated in §3.2.2 two WTA when it is proposed to bypass a spill from one WWTP to another. Hence, one special requirement in this scenario is that agents should be able to enter and leave the deliberation as they are needed. As we will see in §5 *ProCLAIM* allows for this flexibility. Note as well that, as illustrated in Figure 4.5, some agents may only take part in the decision making providing potentially useful information (*e.g.* weather forecast), and so they do not actively participate in the argumentation process itself (see figure4.5).

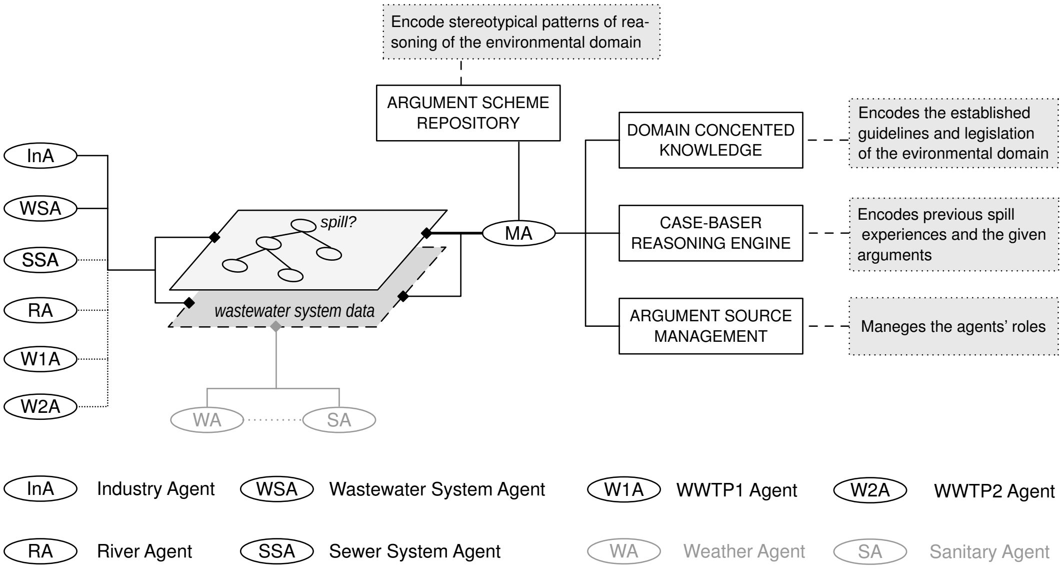

Encode stereotypical patterns of reasoning of the environmental domain
ARGUMENT SCHEME REPOSITORY
DOMAIN CONCENTED KNOWLEDGE
Encodes the established guidelines and legislation of the evironmental domain
CASE-BASER REASONING ENGINE
Encodes previous spill experiences and the given arguments
ARGUMENT SOURCE MANAGEMENT
Maneges the agents' roles
MA
spill?
wastewater system data
InA
WSA
SSA
RA
W1A
W2A
WA
SA
InA Industry Agent
WSA Wastewater System Agent
W1A WWTP1 Agent
W2A WWTP2 Agent
RA River Agent
SSA Sewer System Agent
WA Weather Agent
SA Sanitary Agent

Figure 4.5: *ProCLAIM*'s architecture in the environmental scenario.

The DCK will primarily encode the guidelines and regulations governing the wastewater system, where the discharge standards to point sources are defined indicating the permitted quality of wastewater discharged. This is defined through the provision of thresholds that set the permitted polluting parameters in terms of concentration and load. The safety of an industrial spill depends on a wide variety of factors, from the spills' composition and quantity, to the characteristics and circumstances of the different actors of the wastewater system, including the courses of actions these actors can perform. Hence, in so far as the DCK will not account for all these factors, the participants' opinion and the CBRc's assessment would play a more predominant role.

The schemes and CQs of the ASR should guide the PAs in addressing this wide range of potentially relevant factors. However, provisional results on this scenario suggest a lack of preexisting reasoning patterns suitable for an effective deliberation. We believe, this is principally due to the novelty of the proposed scenario. To our knowledge little work as being done addressing decision-making in wastewater system from an holistic viewpoint. For that reason, current state of research on this scenario [39][4] is focused on gaining a better understanding of the wide range of factors involved in the decision making, their interrelations and the reasons why they are relevant, that is, how these factors affect the spill's ecological impact. All this information is required to adequately construct the ASR.

In our case study, when an industry has some wastewater to discharge, if connected to a WWTP, the standard course of action to follow is to spill the discharge *via* the sewer system to the WWTP which, after treatment, discharges it to the final receiving media, the river. Thus in the environmental scenario, a scheme with which to begin the argumentation is of the form:

Industry `Ind` will discharge the wastewater `W` via the sewer system `SS` to be treated by the WWTP `WWtp` and discharged to the river `R`.

In this context, once an industry connected to a sewer system has some wastewater to discharge, this scheme can be instantiated. Subsequent schemes and CQs will guide agents to consider the safety of the spill based on the properties of the wastewater and the particularities of the wastewater systems, either circumstantial (*e.g.* weather conditions) or intrinsical (*e.g.* each WWTP particular characteristics).

Regarding the instantiation of the CBRc, as discussed above, it is mostly instantiated when building the ASR, of course there are other aspects to be addressed, such as the different evidential support associated to each resolved case as discussed in §9.

Regarding the AEM, the approach taken so far is to classify the confidence in the agents' arguments on the basis of the roles they enact. Thus, for example, arguments addressing the WWTPs operations will be deemed more reliable if endorsed by a WTA than if only endorsed by an InA.

4.3 Discussion

In this chapter we give a high level description of the *ProCLAIM* model focusing on its architecture and introducing its different components. We highlighted how, following an argumentative process shaped by the schemes in the ASR, *ProCLAIM* is intended to elicit the relevant knowledge from the participant agents, as well as from the DCK and the CBRc, organise this knowledge in a tree of interacting arguments, denoted as $\mathbb{T}$. In addition to these arguments, $\mathbb{T}$ may contain other information such as the preferences between arguments given by the different knowledge resources. This extended tree of arguments conform a solution proposal for the problem at hand. If at the end of the deliberation the main proposed action is deemed safe and eventually performed, $\mathbb{T}$ is adequately updated to encode any safety-related complication that may occur during, or after, performing this action. For instance, if the action caused some severe undesirable side effects, these side effects and the reasons for them being caused, should be added to $\mathbb{T}$. This updated tree of arguments will

[4] Undertaken by members of the Laboratory of Chemical and Environmental Engineering at University of Girona, as discussed in §3.2.

then represent this particular case (*e.g.* organ transplant or industrial spill). These updated $\mathbb{T}$s which contain deliberation solutions backed by evidence are retained by the CBRc and used to resolve future similar cases on an evidential basis.

The use of a mediator agent is certainly not new. Many works addressing cooperative environments (*e.g.* production and logistics [91, 205]) feature an agent, or set of agents, dedicated to coordinate the tasks performed by the different working components or agents, in occasions these agents are referred to as mediator agents. In argumentation, the role of the mediator agent is usually associated with negotiation dialogues [214, 64, 209, 167], where the main objective of the mediator agent is to help reconcile the competing agents' positions, for which the mediator agent usually relies on mental models of the participants. In the chapter dedicated to the CBRc we describe other approaches that also make use of Case-Based Reasoning for that purpose (*e.g.* [214]). Notably relevant to our work is the recently proposed SANA [167] framework (Supporting Artifacts for Negotiation with Argumentation), presented as a conceptual framework for negotiation dialogues. This framework proposes a number of so called artifacts, such as a social Dialogue Artifact, that acts as a mediator which regulates the agents dialogue interaction and a social Argumentation Artifact that can be regarded as a sophisticated commitment store, where the agents' submitted arguments, as well as other arguments that may be publicly available, are organised and their *social* acceptability status can be evaluated following different algorithms. Similar to our approach, the SANA framework defines, as part of the social Dialogue Artifact, an Argumentation Store (AS) that stores a collection of *socially acceptable* arguments. The main difference being, that while a central part of *ProCLAIM* is the definition of the structure and relation of the schemes in ASR tailored for the specific purpose of deliberating over safety critical actions, the SANA's AS is presented as a placeholder for any argument scheme, that is, developers are given little guidance on which argument schemes the AS should encode. In a broather view, the SANA approach is similar to that proposed in the FP6-European project ASPIC[5], where a set of generic components where developed: (Inference Engine, Dialogue Manager, Learning Component, Decision-Making component) that can be plugged into an agent in order to add argumentation capabilities. In [11] we presented an implementation of CARREL instantiating the *ProCLAIM* model making use of the Inference Engine as well as the Dialogue Manager; we discuss this implementation in §10.1.

Having introduced the *ProCLAIM*'s architecture, in the next chapters we describe the different components of the model in detail. In the following chapter we introduce the model's deliberation dialogue game, which defines the overall interaction between the different components.

[5] http://www.argumentation.org

Chapter 5

ProCLAIM's Deliberation Dialogue

In this chapter we describe the interaction protocol that governs *ProCLAIM*'s dialogue. In each exchanged locution we distinguish three interaction levels:

1. On the deepest level there is the content of the message, *.e.g.* the submitted arguments.
2. Each of these messages is wrapped in the appropriate deliberation locution defined by the dialogue game (*.e.g.* an argument is wrapped in an *argue* locution).
3. In turn, each of these deliberation locutions is wrapped in either an *inform* or *request* locution. This is because the Participant Agents (PAs) always interact with the MA, never with other PAs. They submit a request to, for example, enter the dialogue, submit a new argument or add new information. The MA then decides whether to accept or reject their request. Thus, the MA acts as a proxy for the PAs (see Figure 5.1a.)

In the following section we describe the inform-request interaction which we call the *proxy* dialogue game. And in §5.2, we introduce *ProCLAIM*'s deliberation dialogue game where we define the locutions that can be submitted at each stage and layer of the deliberation. We then introduce the dialogue game's axiomatic semantics defining the *pre* and *post* conditions for each dialogue move.

5.1 *Proxy* Dialogue Game

Agents participate in the deliberation via the MA, which decides whether an incoming message should be accepted or rejected. Messages are obviously rejected if syntactically ill-formed, but also if the content is not appropriate. For example, a submitted argument may be rejected if the MA deems it non relevant for the deliberation. For that reason, each participant message is wrapped in a `request` locution to which the MA replies with an `inform` locution, either to inform of its rejection (and *why* it is rejected) or to act upon the request. For example, if the request is to enter the dialogue, MA will inform of the participant's acceptance, along with the extra information required for the appropriate participation. The MA may also send an `inform` locution without prior requests, *e.g.* to inform of a *time-out* constraint which forces the deliberation to conclude.

The messages' structure is as follows:

`request(pa_id, ma, conv_id,msg_id, target_id, R)`: where `pa_id` is the sender's id (a PA), `ma` is the receiver agent (the MA), `conv_id` is the conversation id, `msg_id` is the message identifier, `target_id` is the message to which this locution is directed (when the message is not directed to any specific message, `target_id` should be set to `-1`). `R` is a variable denoting the content being communicated in the request locution. The possible values of `R` are discussed in the following subsection.

`inform(ma,PA,conv_id,msg_id,target_id, I)`: Here, the locution may be addressed to a single receiver, in which case `PA` is `pa_id`, or it may be broadcast to all the participants, in which case `PA` is `all`, or to a subgroup of PAs. `I` is a variable denoting the content being communicated in the inform locution, which may be in reply to a request of a PA's request.

While the conversation is open, the PAs can submit their request at any time and the MA must reply to their request with the appropriate `inform` message. In the following subsection we define the messages' content, *i.e.*, the `R` and the `I`.

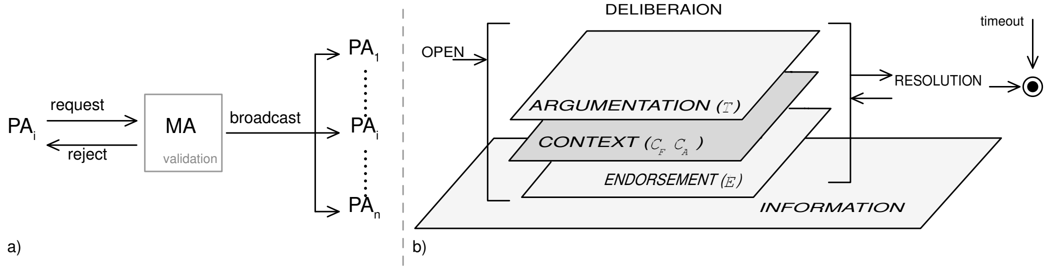


Figure 5.1: *a)* Illustrating the proxy dialogue game; *b)* Depiction of *ProCLAIM*'s deliberation dialogue game with its stages and interaction layers.

5.2 The Deliberation Dialogue Game

In this subsection we introduce *ProCLAIM*'s deliberation dialogue game. That is, we introduce the legal locutions, together with the rules which govern their use as well as the commencement and termination of the deliberation dialogue. As illustrated in 5.1a the deliberation dialogue game can be subdivided in three stages: *Open*, *Deliberation* and *Resolutions*. The Deliberation Stage can in turn be subdivided in three layers: *Argumentation*, *Context*, *Endorsement* layers. The moves in these three layers may happen in parallel and the moves at one layer may have an effect on another layer. As depicted in 5.1b we define yet another interaction layer, called *Information layer* in which PAs can request the MA for updates on ongoing deliberation. It is worth noting that these distinct layers and stages are conceptual and are used as metaphors to better organise the dialogue game, *i.e.* the PAs need not know of these distraction in order to effectively participate.

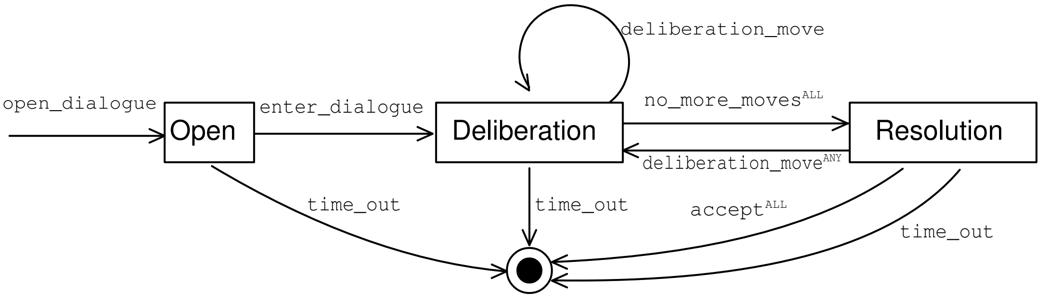


Figure 5.2: Deliberation dialogue, stage transition. We use `deliberation_move` to indicate any dialogue move defined in the *Deliberation Stage* (*e.g.* `assert`, `argue`, `endorse`...). Note that to move from the *Deliberation Stage* into the *Resolution Stage* all PA must have submitted the `no_more_moves` move. Similarly, to conclude the deliberation all PA must have submitted the `accept` move, unless the `time_out` has been triggered. If any PA submits a deliberation move, while being a the *Resolution Stage*, the dialogue moves back to the *Deliberation Stage*.

5.2.1 Open Stage:

The first stage is *Open* in which the proposal is made, the participants are introduced and basic available information is provided to all participants:

`open_dialogue(proposal):` `proposal` is an argument proposing the main action (*e.g.* transplant an available organ). `proposal` also contains the preconditions for the action's performance (*e.g.* an available organ and a potential recipient). As we see in §6, `proposal` is an instantiation of an argument scheme.

If the proposal is made by a PA (and not by the MA), this message is wrapped in a `request` locution that the MA would have to validate. If the request is validated, the MA will submit the `open_dialogue` locution and contact the potential participants in order to enter the dialogue.

`enter_dialogue(proposal,role, basic_info):` Each agent willing to participate in a deliberation over `proposal` will enter her `role` in the deliberation (*e.g.* *donor* or *recipient* agent) and the information (`basic_info`) she deems potentially relevant for the decision making (*e.g.* the patient's clinical record). This message is wrapped in a `request` locution.

If the `enter_dialogue` is accepted, the introduced facts, via `basic_info`, will be stored in a set of facts, which we denote $\mathbb{C}_F$.Similarly, we define a second set denoted as $\mathbb{C}_A$ which contains the agents' proposed actions. Initially $\mathbb{C}_F$ contains only the facts introduced in the `enter_dialogue` locution and the preconditions introduced in the argument proposing the main action. When the deliberation starts $\mathbb{C}_A$ contains only the initially proposed action (*e.g.* transplant the available organ to the potential recipient).

For simplicity, let us denote $\mathbb{C}_F$ and $\mathbb{C}_A$ together as $\mathbb{C}_{F\wedge A}$. During the deliberation $\mathbb{C}_{F\wedge A}$ may be updated, this happens at the *Context Layer* (§5.2.2).

The proposal `proposal` made in `open_dialogue(proposal)` is an argument for the main action. Thus, this is the first argument added to the tree of arguments $\mathbb{T}$. Further submitted arguments at the *Argumentation Layer* (§5.2.3) will update $\mathbb{T}$.

A PA may request to enter the dialogue at the beginning of the deliberation, or later, when both $\mathbb{C}_{F\wedge A}$ and $\mathbb{T}$ may have more information than the minimal information available at the beginning. Thus if at any stage an agent's request to participate is accepted, the MA will reply by broadcasting the following message.

`entered_dialogue(proposal, role, basic_info, pas,` $\mathbb{C}_{F\wedge A}$**`,`**$\mathbb{T}$**`, legal_replies`**): The MA informs all the participants that an agent enacting the role `role` just entered in the deliberation and has introduced the information `basic_info`. The MA also informs, of the PAs already in the deliberation `pas`, as well as of the updated values of $\mathbb{C}_{F\wedge A}$ and $\mathbb{T}$. Within this message the MA attaches the legal replies (`legal_replies`) to the arguments in $\mathbb{T}$. This set of legal replies (argument schemes and critical questions) will guide the PA on which argument moves they can submit as a reply to those in $\mathbb{T}$. (This is further discussed in §7.2).

Thus, for example, if an agent with id `ag_id` enacting the role `role_id` wishes to enter the deliberation over the proposal `proposal` she will send a request locution:

```
request(ag_id,ma,conv_id,0,-1,enter_dialogue(proposal,role_id,
basic_info))
```

If the MA accepts the request it will broadcast to all but `ag_id` that an agent playing the role `role_id` has entered the dialogue and reply to agent `ag_id` that her request was accepted:

`inform(ma, all-{ag_id},conv_id,1,-1, enter_dialogue(proposal, role_id, basic_info, pas,` $\mathbb{C}_{F\wedge A}$`,` $\mathbb{T}$`, legal_replies))`

`inform(ma, ag_id, conv_id,1,0,entered_dialogue(proposal, role_id, basic_info, pas,` $\mathbb{C}_{F\wedge A}$`,` $\mathbb{T}$`, legal_replies))`

If the request is rejected the MA informs the PA with id `ag_id` why her request was rejected.

```
inform(ma, ag_id, conv_id, 1, 0, rejected(reason)).
```

Once a PA enters the dialogue it moves into the deliberation stage and it can interact in its three layers:

5.2.2 Context Layer:

Once an agent enters the dialogue it can inform of facts it deems potentially relevant for the decision making, as well as propose complementary courses of actions that may prevent undesirable side effects that may be caused by the main action.

`assert(fact):` a PA asserts that the fact `fact` is the case. If accepted, `fact` is added to $\mathbb{C}_F$.

`propose(action):` a PA proposes to perform the action `action`. If accepted, `action` is added to $\mathbb{C}_A$.

`retract(fact):` a PA retracts an assertion that a fact `fact` is the case. If accepted, `fact` is removed from $\mathbb{C}_F$.

`retract(action):` a PA retracts the proposal to perform the action `action`. If accepted, `action` is removed from $\mathbb{C}_A$.

Each of the above messages, when sent by a PA, is wrapped in a request locution. If they are accepted, they will be broadcasted to all participants by the MA.

Participants may assert and retract facts as well as propose and retract actions, at any time, as long as the deliberation is open. The only restriction is that facts and actions asserted or proposed cannot be inconsistent[1]. Hence, given a consequence relation $\vdash$ and a background theory Γ, then $\mathbb{C}_F$ and $\mathbb{C}_A$ must be such that $\mathbb{C}_F \nvdash_\Gamma \perp$ and $\mathbb{C}_A \nvdash_\Gamma \perp$. For instance, $\mathbb{C}_F$ cannot contain both: *a)* the donor does not have cancer and *b)* the donor has a malignant tumour.[2]. In other words, the state of affairs defined in $\mathbb{C}_{F \wedge A}$, though may be uncertain and may evolve throughout the deliberation, cannot be inconsistent.

At the current state of development of *ProCLAIM* does not support a conflict resolution among PAs that disagree over the described contexts of facts. From our explored scenarios (transplant and environmental) we have learned to be odd for one PA to dispute another PA's state of affairs description. This is because, each PA provides information on that she has a privileged access to. Hence, it is odd for a DA to dispute the information about a potential recipient given by a RA; similarly for an agent representing an industry to dispute information regarding the status of the wastewater treatment plant. For this reason, and in order to keep the deliberation focused, conflicts regarding whether or not a fact `x` is the case is either resolved outside *ProCLAIM* (*e.g.* by facilitating a persuasion dialogue or via a phone call) or should take `x` as uncertain. Nonetheless, as we will see in §6.2.3, PAs can still challenge an argument requesting evidence in support of some fact and my highlight

[1] Where by inconsistent actions we mean actions that cannot be performed simultaneously (*e.g.* heat and cool, stay and go, etc...).

[2] Note however that $\mathbb{C}_F$ may contain *a)* clinical records indicate the donor does not have cancer and *b)* the donor has cancer

the weakness of that evidence, which may motivate the retraction of the disagreed upon fact (*e.g.* the retraction of x, which leaves room to the submission of $\neg x$). In future work we intend to further develop this intuition, which may lead extending *ProCLAIM* to support such conflict resolution.

In line with the above discussion, we currently assume that the information given by the PAs when entering the dialogue (*i.e.* the set of basic information `basic_info`) are mutually consistent. This assumption should indeed be addressed in future work, either to allow these information to be inconsistent or to better motivate the reasons that license this assumption. For the time being, we will assume in the transplant scenario that a DA_i's submitted `basic_info`, containing data of the potential donor, is consistent with the data given by a RA_j of the potential recipient. Same assumption applies for the environmental scenario.

We should also note that we deem outside *ProCLAIM*'s scope to define policies for deciding which facts or action can be asserted (resp. proposed) or retracted by each participant. Namely, we believe these decisions are application dependent. Possibly, good practice would be to allow participants asserting (resp. proposing) only those facts which they have some knowledge about (resp. action that they can perform), thus preventing, for example, a RA to add information about the donor. In the same way it would be reasonable to allow participants to retract only those facts (resp. actions) that they have asserted (resp. proposed). All such decisions should be made by the MA at the proxy layer.

Note that the facts and actions introduced at this stage of the deliberation (*i.e.* $\mathbb{C}_{F \wedge A}$) do not themselves indicate whether or not the main proposed action is safe. $\mathbb{C}_{F \wedge A}$ is the context in which the main action is intended to be performed. Participants should thus decide whether the proposed action is safe given this context, where $\mathbb{C}_{F \wedge A}$ may change during the course of the deliberation. This decision making occurs at the *Argumentation Layer*. Although clearly, if the main proposed action or the preconditions for such an action are retracted, the deliberation concludes.

5.2.3 Argumentation Layer:

At the argumentation layer there are only two locutions: `argue` and `challenge`. A PA uses these locutions to *request* submitting an argument or a challenge. A challenge made on an argument questions the validity of the argument. From the perspective of an argumentation framework challenges can be represented as regular arguments that attack the argument under challenge. If the PA's request for submitting an argument or a challenge is accepted, the MA broadcasts this move to all participants using its version of the `argue` and `challenge` locutions. When this request is rejected, the MA's reply occurs at the proxy layer. Let us mark the locutions made by PAs with an **R** for request, and the MA's broadcasting message with an **I** for inform:

R: `argue(argument, target)`: an argument `argument` is submitted by a PA in reply (as an attack) to the *argument* or *challenge* in $\mathbb{T}$, whose id is `target`. If the argument is accepted it will be broadcasted to all participants.

I: `argue(id, argument, target, legal_replies)`: an argument `argument` submitted in reply (as an attack) to the *argument* or *challenge* whose id is `target` has been accepted by the MA who broadcasts it to all participants indicating that the argument's id is `id`. Within the same message, the MA attaches the legal replies (`legal_replies`) to `argument`. This set of legal replies (argument schemes and critical questions) will guide the PA on which argument moves they can submit at each stage of the deliberation (this is further discussed in §7.2). `argument` is also added to $\mathbb{T}$, attacking the argument or challenge with id `target`.

R: `challenge(challenge, target)`: a challenge `challenge` is made by a PA on an argument in $\mathbb{T}$ with id `target`. In reply to a challenge participants can submit an argument that meets the challenge (see §6.2.3).

I: `challenge(id,challenge, target, legal_replies)`: a challenge `challenge` made on an argument with id `target` has been accepted by the MA who broadcasts it to all participants, indicating that the challenge's id is `id`. Within the same message, the MA attaches the legal replies (`legal_replies`) to `challenge`. The challenge is added to $\mathbb{T}$ as an argument attacking the argument with id `target`.

All participants, including the MA, can submit arguments and challenges at any time as long as the deliberation is open and the target argument or challenge is in $\mathbb{T}$. However, the MA can reject a submitted argument or challenge because it is not a relevant move. That is, the MA's validation task introduced in §4 is performed at the proxy layer.

The fact that a participant submits an argument does not imply she endorses it. A participant may attack her own submitted arguments with other moves. This is because it is a collaborative setting, as opposed to a competitive one. Participants introduce the knowledge they have of the problem in the form of arguments. Thus, for example, the same agent can highlight a contraindication for performing the main action (attacking the initial argument) but then propose a complementary action that will mitigate its undesirable side effects and thus reinstate the main action proposal. In the same spirit, once a challenge or argument is added to $\mathbb{T}$ participants cannot retract it, *i.e.* delete it from $\mathbb{T}$. As discussed in §4.1, if an argument is added to $\mathbb{T}$ it is because the MA deemed the argument to be relevant for the deliberation. An argument may of course be defeated, but it should remain in the tree of arguments.

5.2.4 Endorsement Layer:

As arguments are added to the tree of arguments, participants can decide which arguments they endorse. This endorsement will affect MA's argument evaluation. For example, arguments endorsed by participants with a good reputation will be deemed stronger. Nonetheless, this argument may still be weak because, for instance, there is strong empirical evidence against it. The locutions at the *Endorsement Layer* are:

`endorse(pa_id,arg_id)`: The participant `pa_id` endorses argument or challenge with id `arg_id`.

`retract_endorsement(pa_id,arg_id)`: The participant `pa_id` retracts her endorsement of argument or challenge with id `arg_id`.

These moves can be submitted at any time while the dialogue is open and on any argument or challenge on $\mathbb{T}$. If an agent endorses two conflicting arguments, the later endorsement prevails and the earlier is automatically retracted.

When an endorsement (resp. its retraction) of an argument or challenge in $\mathbb{T}$ is made by a PA (via a request locution), the MA adds (resp. subtracts) this endorsement (represented as the predicate `endorse(agent_id,arg_id)`) from the **endorsement set**, which we denote as $\mathbb{E}$.

5.2.5 Resolution Stage:

Once participants have constructed the context of facts and actions $\mathbb{C}_{F \wedge A}$, the tree of arguments $\mathbb{T}$, and have informed of their endorsements, the MA proceeds to evaluate $\mathbb{T}$. The deliberation moves into the *Resolution Stage* either because all the participants have informed that they have no further moves to submit that may change either $\mathbb{C}_{F \wedge A}$, $\mathbb{T}$, or $\mathbb{E}$; or because a *timeout* was triggered. In either case, the MA proposes a solution for the deliberation, based on the evaluation of $\mathbb{T}$. If a *timeout* has been triggered, PAs will not have the chance to revise the proposed solution. In §8 we discuss the nature of such evaluation and how a recommended solution is not merely a *safe*/*unsafe* answer.

`no_more_moves()`: The participant informs that she has no further moves to submit (moves that may change either $\mathbb{C}_{F \wedge A}$, $\mathbb{T}$, or $\mathbb{E}$), for consistency she does so via a request move. Once all participants submitted this move, the MA proceeds to evaluate $\mathbb{T}$. This move, however, does not prevent participants from submitting further moves, overriding her own move of `no_more_moves`. This is important to allow because new relevant information may be available at any time and should be included in the deliberation. If the move `no_more_moves` is overridden, the deliberation moves again the deliberation stage.

`leave_dialogue(reason)`: The participant request that to leave the deliberation and may provide a reason `reason` for that. If this move is accepted by the MA all PAs will be informed that the participant has left the deliberation. Of course, if all participants leave the deliberation the deliberation concludes and the MA will propose a solution (via the `close_deliberation` locution) on the basis of the available knowledge $\mathbb{C}_{F \wedge A}$, $\mathbb{T}$, and $\mathbb{E}$.

`time_out(reason)`: The MA informs that a timeout has been triggered. In general terms this means that too much time has been spent in the deliberation and so a new resolution policy should be applied. For instance, picking-up the telephone. How to proceed with a timeout is application dependent. Provisionally we formalise it as a trigger for the MA to evaluate $\mathbb{T}$ with the available knowledge ($\mathbb{C}_{F \wedge A}$, $\mathbb{T}$, and $\mathbb{E}$) and propose a solution while disabling any further moves from the participants. The MA may provide a reason `reason` for the timeout.

`solution(solution,sol_id)`: Once all participants have submitted the `no_more_moves` (and did not override it with any other move) the MA proposes a solution `solution` whose id is `sol_id`. The proposed solution may motivate participants to submit further moves or to simply accept the solution. If a participant submits a move in the *context*, *argumentation* or *endorsement* stage, she should again submit the `no_more_moves` locution for the MA to propose the new solution. However, if the timeout is triggered, the deliberation will conclude with the given solution providing no chance for the participants to submit further moves[3].

`accept(sol_id)`: Once a solution with id `sol_id` is given, if all agents accept it, the deliberation concludes.

`close_deliberation(solution)`: The deliberation is closed with the proposed solution `solution`. This locution is submitted either after all participants have submitted the `accept(sol_id)` move or the timeout has been triggered and the MA has proposed a solution.

We are working under the assumption that the CBRc (case based reasoning component) is *time consuming* and requires the full $\mathbb{T}$ for argument evaluation. However, if we manage to develop a CBRc whose performance can be adjusted to real-time deliberation, a proposal for resolution of the $\mathbb{T}$ will always be visible for the participants and the cycle `solution(solution,id_sol)`, `accept(id_sol)` will not be necessary. It would be enough to submit the `no_more_moves` locution.

5.2.6 Inform Layer:

Throughout the deliberation dialogue, participants can request from the MA an update of the argument tree, in which facts have been introduced, or request for the legal replies to a given argument or challenge in $\mathbb{T}$. Thus, if for whatever reason a participant misses a piece of information she can recover it upon request.

R: `get_arg_tree()`: A PA requests the MA for the updated $\mathbb{T}$.

I: `arg_tree(`$\mathbb{T}$`)`: The MA informs a PA of the updated $\mathbb{T}$.

R: `get_context()`: A PA requests the MA for the updated $\mathbb{C}_{F \wedge A}$.

I: `context(`$\mathbb{C}_F$`,` $\mathbb{C}_A$`)`: The MA informs a PA of the updated $\mathbb{C}_{F \wedge A}$.

R: `get_endorsement()`: A PA requests the MA for the updated $\mathbb{E}$.

I: `endorsement(`$\mathbb{E}$`)`: The MA informs a PA of the updated $\mathbb{E}$.

R: `get_legal_replies(arg_id)`: A PA requests the MA for the legal replies to an argument or challenge in $\mathbb{T}$ with id `arg_id`.

[3] In that case, the decision making process may indeed continue, but following a different policy.

I: `legal_replies(arg_id, legal_replies)`: The MA informs a PA of the legal replies to an argument or challenge in $\mathbb{T}$ with id `arg_id`.

Having introduced all the locutions in the dialogue game, we now list their *pre* and *post* conditions.

5.3 Dialogue Game's Axiomatic Semantics:

Let us now introduce the axiomatic semantics of *ProCLAIM*'s dialogue game. Let us first however, recall that we defined the dialogue game in three levels, where the deepest level addresses the content of the messages (*e.g.* the exchanged arguments), the locutions that wrap the messages content and finally, on the shallowest level we defined the Proxy level in which the PAs interact via `request` - `inform` cycles through the MA. The axiomatic semantics will only cover the second level. At the Proxy level there is not much to say. There are no preconditions to submit an `inform` or a `request` locution, other than that defined on at the second level of interaction. The only post condition is associated to the PAs `request` locutions to which the MA has to respond accordingly. In the following description we will obviate the `request` - `inform` interaction and assume all locutions at the second level are `inform`. The deepest level, that of the content is related more with the argument validation process described in §7.3. Most moves share the same set of preconditions, for simplicity let us refer to these preconditions as *standard preconditions*. These are: *1)* Dialogue is open, *2)* the `time_out` locution has not been triggered and *3)* a solution to the deliberation has not been accepted by all PAs.

`open_dialogue(proposal)`

Pre Conditions:

The dialogue is not open

Post Conditions:

The dialogue is open, `proposal` is set as the root of $\mathbb{T}$ the preconditions of `proposal` are added to $\mathbb{C}_F$ and the proposed action is added to $\mathbb{C}_A$.

`entered_dialogue(proposal, role, basic_info, pas,` $\mathbb{C}_{F \wedge A}$`,`$\mathbb{T}$`, legal_replies)`

Pre Conditions:

standard preconditions and PA is not already in the dialogue.

Post Conditions:

The PA enters the dialogue and `basic_info` is added to $\mathbb{C}_F$.

`assert(fact)`

Pre Conditions:

standard preconditions.

Post Conditions:

If `fact` is consistent with $\mathbb{C}_F$ then `fact` is added to $\mathbb{C}_F$. If the deliberation

`propose(action)`
Pre Conditions: *standard preconditions*.
Post Conditions: If `action` is consistent with $\mathbb{C}_A$ then `action` is added to $\mathbb{C}_A$.

`retract(facts)`
Pre Conditions: *standard preconditions*.
Post Conditions: `fact` is removed from $\mathbb{C}_F$.

`retract(action)`
Pre Conditions: *standard preconditions*.
Post Conditions: `action` is removed from $\mathbb{C}_A$.

`argue(id, argument, target, legal_replies)`
Pre Conditions: *standard preconditions*.
Post Conditions: `argument` is appropriately added to $\mathbb{T}$ as an attacker to argument with id `target`, provided the argument is accepted by the MA.

`challenge(id,challenge, target, legal_replies)`
Pre Conditions: *standard preconditions*.
Post Conditions: `challenge` is appropriately added to $\mathbb{T}$ as an attacker to argument with id `target`.

`endorse(pa_id,arg_id)`
Pre Conditions: *standard preconditions*.
Post Conditions: If `arg_id` is in $\mathbb{T}$ then `endorse(agent_id,arg_id)` is added to $\mathbb{E}$. If it is the case that `endorse(agent_id,arg_id2)` $\in \mathbb{E}$ with `arg_id2` an attacker of argument `arg_id` then `endorse(agent_id,arg_id2)` is removed from $\mathbb{E}$.

`retract_endorsement(pa_id,arg_id)`
Pre Conditions: *standard preconditions*.
Post Conditions: If `endorse(agent_id,arg_id)` is in $\mathbb{E}$ it is removed.

`no_more_moves()`
Pre Conditions: *standard preconditions*.
Post Conditions: When all APs have uttered this message, the MA will proceed to evaluate $\mathbb{T}$.

`time_out(reason)`
Pre Conditions: *standard preconditions*.
Post Conditions: PAs cannot perform any other move.

`leave_dialogue(reason)`
Pre Conditions: The dialogue is open.
Post Conditions: The PA is no longer in the dialogue.

`solution(solution,sol_id)`
Pre Conditions: The dialogue is open and either the `time_out` locution has been uttered or all PAs have uttered the `no_more_moves()` locution and have not uttered any other move afterwards.
Post Conditions: A solution is proposed.

`accept(sol_id)`
Pre Conditions: The dialogue is open, the `time_out` locution has not yet been uttered and a solution with id `sol_id` has been proposed
Post Conditions: When all PAs accept a a solution, the dialogue can be closed.

`close_deliberation(solution)`
Pre Conditions: The dialogue is open and either the `time_out` locution has been uttered and a solution proposed, or a solution proposed and all PAs have accepted it
Post Conditions: The dialogue is closed.

5.4 Discussion:

The above described dialogue game is rather liberal. PAs can submit almost any locutions at any time during the deliberation. Even at the *resolution stage* PAs can submit an `argue` locution of the *Deliberation Stage*, provided the dialogue is still open and the timeout has not been triggered. At the proxy level, the MA does have the obligation to reply to the PA's requests. Of course, the deliberation dialogue can only be opened once, PAs can only request to enter the dialogue if they are not already in it and they cannot participate once the deliberation is either closed or they have left it (via the `leave_dialogue` locutions). Furthermore, PAs can submit any fact (resp. complementary action) at any time of the deliberation, as long as this fact (resp. action) is not already asserted (resp. proposed) or it is inconsistent with $\mathbb{C}_F$ (resp. $\mathbb{C}_A$). Similarly, PAs can retract any facts and actions in $\mathbb{C}_{F\wedge A}$. Finally, as in the *context layer*, PAs can submit the `argue` or `challenge` locution at any time of the deliberation, in what we termed the *standard preconditions*. The target of their argument or challenge must be an element of $\mathbb{T}$ and, in particular, they can attack their own arguments and they do not have any obligation to defend their arguments from other arguments or challenges. What is at stake is not who is right or wrong, but whether or not the main action can safely be performed.

Another important feature of this dialogue game, particularly important for the purposes of *ProCLAIM*'s deliberations, is the explicit separation of the *Context* and *Argumentation* layers. This facilitates the decoupling of the resolution of *what is the case* and the deliberation over the actions' safety. While both questions are important for deciding the safety of the proposed action, prioritising the later helps focus the deliberation on the problem at hand. For example, by giving priority to the main question:–*Is the action safe in current circumstances?-* we can impose that questioning the current circumstances (*i.e.* the facts in $\mathbb{C}_F$) is licensed only if this challenges the action's safety (this becomes clearer once we define the argument's structure in §6.1). Another important consequence of this decoupling is that it allows one to address, in a relatively simple fashion, problems such as incomplete or uncertain information, at the time of constructing the arguments, when updating the new available information and when evaluating the arguments. This will be discussed in more detail in the following chapter.

As discussed in §2.4, the description of the dialogue game in terms of stages, bares resemblance with McBurney *et al.*'s deliberation dialogue game [150], which defines eight stages (*Open*, *Inform*, *Propose*, *Consider*, *Revise*, *Recommend*, *Confirm* and *Close Stages*) through which the goals of participants change accordingly. As noted in this chapter, we make a more lose use of the defined stages and layers, in the sense that they only help organise the game's definition, participant agents need not know which locution corresponds to each stage and locutions of different stages may be submitted in parallel.

However, the main difference between the two approaches is that while McBurney's *et al.* dialogue game is intended for an open deliberation about what to do, *ProCLAIM*'s dialogue game is specialised for *ProCLAIM*'s deliberations over the safety of a proposed action, where great part of the interaction is shaped (and constraint) by the argument schemes the participant agents can instantiate at each stage of the deliberation. Indeed, it is at the *Argumentation Layer* that the deliberation is kept highly focused on the subject matter, through definition of the arguments and challenges the PAs can submit throughout the deliberation. That is, the set of legal replies (argument schemes and CQs) made available to the participants. In the following chapter we describe in detail the *Argumentation Layer*.

Chapter 6

ProCLAIM's Argumentation Layer

One of the pillars of *ProCLAIM* is the definition of the deliberation dialogue's *Argumentation Layer*, namely, what types of arguments participants can exchange and following what rules. As a way to keep deliberations focused as well as reducing the participants' overhead in terms of argument construction, *ProCLAIM* is quite specific in what can be argued about and how. To this end, the model defines a *protocol-based exchange of arguments* that can be regarded as an argumentative process for eliciting knowledge from the participants, as opposed to defining a strategic dialogue in which a better choice of arguments may better serve the agents' individual goals. This argumentation-protocol is defined in terms of a structured set (a *circuit*) of schemes and their associated CQs (to a scheme are associated a set of CQs which are themselves defined in terms of schemes that have associated CQs, and so on...). *ProCLAIM* defines an application-independent protocol-based exchange of arguments specialised for arguing over safety critical actions. Then, for each target application (*e.g.* transplant or environmental scenario) this application-independent protocol has to be further specialised in order to construct the scenario-specific ASR[1]. This is discussed in §7.1.

In this chapter we present the application-independent circuit of AS and CQs. We start by introducing in the following section the internal structure of *ProCLAIM*'s arguments and in §6.2, we present the protocol-based exchange of arguments.

6.1 The Structure of an Argument

Action proposals are typically motivated by the goals agents wish to realise. Many formal accounts ([202, 94, 231, 225]) of action proposal assume, though sometimes implicitly, the following three dimensions:

R: Domain of facts in circumstances where the action is proposed.

A**:** Domain of actions.

G: Domain of goals.

[1] Argument Scheme Repository

Based on these domains the following argument can be constructed '*an action A is proposed in circumstances R (a set of facts) because it is expected to realise a desirable goal G*'. The problem with such an argument structure is that the notion of a goal is ambiguous, potentially referring indifferently to any direct result of the action, the consequence of those results and the reasons why those consequences are desired [34]. To account for these distinctions, Atkinson *et al.* considered two additional domains:

S: Domain of facts arrived after performing the action.

V: Domain of Values where the values represent the social interests promoted through achieving the goal.

in order to propose the following argument scheme for action proposals:

AtkSch:
In the circumstances R
we should perform action A
to achieve new circumstances S
which will realise some goal G[2]
which will promote some value V

This argument scheme is presented along with sixteen associated CQs which can be classified into three categories: *What is true* (*e.g. –Questioning the description of the current circumstances–*), *what is best* (*e.g. –Questioning whether the consequences can be realised by some alternative action–*) and *representational inconsistencies* (*e.g. –Questioning whether the desired features can be realised–*). In [34] $AtkSch$ along with its sixteen CQs are used to define a persuasion dialogue game for reasoning about action proposal.

Atkinson's persuasion dialogue is primarily addressed at resolving a choice amongst competing action proposals, choosing which action is the best, *i.e.* which action will bring about the best situation, where *best* is relative to an agent and consideration is given to subjective value-based judgements, as well as more objective ones. In arguing about action proposals, participants may undermine an action proposal by questioning whether the action will bring about any undesirable effects[3]. This is just one possibility in the persuasion dialogue; one can also argue as to which goals are desirable or not. In short, participants can argue about whatever is reasonable when deciding *what to do* in general terms. This generality is indeed a desirable feature of Atkinson's persuasion dialogue and for that reason this work is taken as a starting point for the definition of *ProCLAIM*'s *Argumentation Layer*. However, precisely because of this openness, it is inoperable for our intended applications. In §11 we discuss two experiences in which we used both Atkinson *et al.*'s proposed scheme

[2] Where a goal is some particular subset of S that the action is intended to realised in order to promote the desired value.

[3] We cannot assume that because the effect is undesirable it must be a *side effect* of the action. It may actually be a state of affairs that, from the perspective of one participant, is a desirable outcome of the action, but not for all participants.

and Walton's schemes [231] in order to *1)* guide end users in the argument construction in real time deliberations (particularly in the transplant scenario, discussed in §11.1) and *2)* guide developers in the construction of scenario specific schemes that will build a new application's ASR (in the environmental scenario §11.2). While in both cases the general principles were understood, the actual instantiation (or partial instantiation in the later case) involved too much overhead and inconsistent results. We learned that deciding on the fly which are the *goals* and *effects* of a given action, deciding what among all the current facts indicate as *current circumstances* and furthermore, deciding what makes the action safe or unsafe based on these abstract terms was not an easy task. For this reason we initially proposed in the transplant scenario the use of scenario specific schemes [17], while the results obtained with this formalisation were very positive [11], its *ad-hoc* approach made it difficult to apply in novel scenarios, in particular for developers who are not familiar with Argumentation Theory. It is the realisation of the value of scenario-specifc schemes, jointly with the need for procedures to facilitate the production of these specialised schemes that has led us to develop the current formalisation of *ProCLAIM*'s *Argumentation Layer* that we now introduce.

In *ProCLAIM*, the desirable and undesirable goals are assumed to be shared by all participants. Furthermore, the main proposed action itself (*e.g.* transplant an organ or spill the industrial wastewater) is, in default circumstances, taken to be the right thing to do, requiring no further motivation in its proposal. Moreover, decisions in *ProCLAIM* are taken with respect to a single social value *safety* (or patient's quality of life, in the transplant scenario). Therefore, the value dimension can be ignored[4]. A particular consequence of this defined context is that PA's individual goals and values, while may affect which arguments they submit and endorse, in theme selves do not constitute a reason for or against a proposed action. What becomes a matter of debate then, is whether the current circumstances are such that the proposed action can safely be performed. Namely, whether or not the context of facts $\mathbb{C}_F$, constructed at the *Context Layer*, is such that the main action will bring about severe undesirable side effects. The deliberation can thus be regarded as an argumentative process for eliciting from the participants (experts) what are the *relevant* facts ($\mathtt{f}_0$,..,$\mathtt{f}_n \in \mathbb{C}_F$) for assessing the action's safety, accounting for the complementary courses of actions (those actions added to $\mathbb{C}_A$). A formal definition of the relevance of a set of facts is given later in this section (Definition 6.1).

To illustrate the relevance of facts in the medical scenario, let us suppose a donor of a lung is infected with the Hepatitis C virus (hcv). Now, it can be argued that the transplant is unsafe (argument $A2$ in fig. 6.1) because the recipient of the transplanted lung will result in having hcv, which is a severe infection. Thus, the donor being infected with hcv is a relevant fact, given that, because of this fact the transplant will cause an undesirable side effect. Suppose now that the potential recipient also has hcv. And so, it cannot be claimed that, for this recipient, having hcv is an undesirable side effect of the lung transplant (argument $A3$ in fig. 6.1). Therefore, the potential recipient's hcv is a relevant fact. It is because that

[4] It may be interesting to bring into the deliberation the *cost* value. Some proposed actions although deemed safe, cannot be taken because the system cannot afford the expenses incurred by the actions. We leave such an extension for future work.

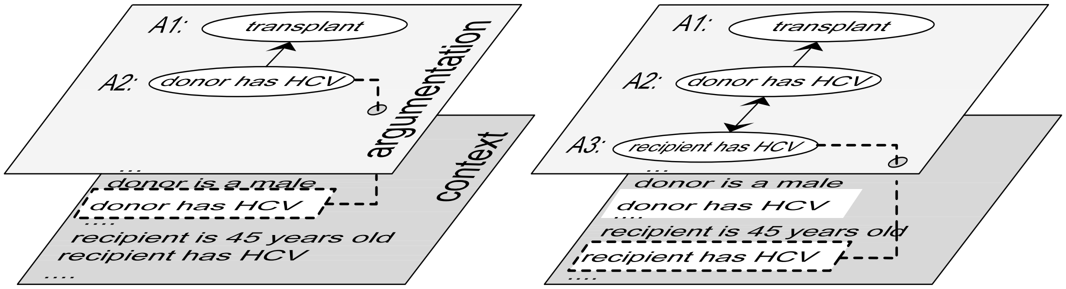


Figure 6.1: As arguments are submitted facts are highlighted as relevant. Note that for example, it has not been deemed relevant that the recipient is 45 years old or the donor is a male. Moreover, if the donor would not have had HCV (Hepatitis C virus), the recipient's HCV may have not been highlighted either.

fact holds that the action does not cause an undesirable side effect. Note however, that if the donor would not have had `hcv`, whether the recipient has `hcv` or not, is irrelevant. That is, relevance is *context dependent*. An attack on argument $A3$ will assume a *context* in which the donor and recipient both have `hcv`. Let us suppose that there are other contraindications for the transplantation that, at least *a priori*, are independent of the donor and recipient's `hcv`. For example, that the available lung is too big for the recipient's thoracic cavity. Such an argument will directly attack argument $A1$, where the context, or to be more precise, the *local context*, is that an available organ is proposed for transplantation into a given patient. To capture this notion, we explicitly associate to each argument a *local context* of facts and of actions.

We denote the **local context of actions** of an argument as $\mathcal{A}$, and the **local context of facts** as $\mathcal{C}$. Upon submission of the first argument, $\mathcal{A}$ and $\mathcal{C}$ are empty. These are updated so that for any subsequent submitted argument $\mathcal{A}$ contains the proposed action itself (*e.g.* the transplant proposal) and $\mathcal{C}$ the minimum set of facts where the proposed action can be performed (*e.g.* an available organ and a potential recipient). In general, each submitted argument updates its $\mathcal{C}$ and $\mathcal{A}$ to account for the particularities of each case (*e.g.* the donor's and recipient's particularities). In the previous example we saw how argument $A2$ extended $\mathcal{C}$ to include the donor's `hcv`. Argument $A3$ then extended $\mathcal{C}$ by adding the recipient's `hcv`. Note that while these facts were already in the (global) context $\mathbb{C}_F$, it's through their use in the argumentation that they are highlighted as relevant. Thus, for a set of facts (resp. actions) to be added to $\mathcal{C}$ (resp. to $\mathcal{A}$) it must be *relevant*. Meaning that, within their *local* context these facts or complementary actions make the main action safe or unsafe.

To continue with the identification of the elements and relations of *ProCLAIM*'s arguments, let us recall that a *ProCLAIM* argument expresses a relation among the four domains: current state ($\mathbf{R}$), actions ($\mathbf{A}$), arrived states ($\mathbf{S}$) and goals ($\mathbf{G}$). We can further constrain $\mathbf{S}$ and $\mathbf{G}$ so that $\mathbf{S}$ contains only *side effects* of the actions in $\mathbf{A}$, and $\mathbf{G}$ contains only *undesirable goals* which such side effects may realise.

Let us formalise these domains in terms of finite sets of grounded predicates which will be written in `teletype`, *e.g.* `av_org(d,o)` $\in \mathbf{R}$ meaning that an organ `o` of a donor `d` is available.

ProCLAIM arguments express relations between elements of the above domains. Specifically, the following elements:

$\mathcal{C}$**:** The local context of facts assumed to be the case, where $\mathcal{C} \subseteq \mathbf{R}$.

$\mathcal{A}$**:** The local contexts of proposed actions, where $\mathcal{A} \subseteq \mathbf{A}$.

R**:** A set of facts, where $R \subseteq \mathbf{R}$. For more than one set of facts we write $R1$, $R2$,... We denote by R_p the set of facts introduced as preconditions for the performance of a proposed action.

A**:** A set of actions, where $A \subseteq \mathbf{A}$. We write A_m to denote the *main set of actions* and A_c the complementary courses of actions argued to prevent the achievement of an undesirable side effect. For more than one set of complementary actions we write A_c1, A_c2,...

S**:** The set of side effects caused by the proposed action, where $S \subseteq \mathbf{S}$. For more than one set of side effects we write $S1$, $S2$,...

g**:** The undesirable goal realised by S, where $g \in \mathbf{G}$. For more than one goal we write $g1$, $g2$,...

Different argument schemes defined by *ProCLAIM* correspond to different relations amongst these elements, where these relations are expressed in terms of special second order predicates, and a defeasible consequence relation $\mid\!\sim$ from which conclusions follow defeasibly or non-monotonically from the set of premises. We thus assume:

- A defeasible consequence relation $\mid\!\sim$;
- A background theory Γ;
- The special predicate **`side_effect`** on subsets of $\mathbf{S}$. Where `side_effect(`S`)`, with $S \subseteq \mathbf{S}$, denotes that S are side effects given a background contexts of facts $\mathcal{C}$ and actions $\mathcal{A}$.
- The special predicate **`und_goal`** on $\mathbf{G}$. Where `und_goal(g)`, with `g` $\in \mathbf{G}$, denotes that g is an undesirable goal realised given a background contexts of facts $\mathcal{C}$ and actions $\mathcal{A}$;
- The special predicate **`intended`**[5] on subsets of $\mathbf{A}$. Where `intended(`A_c`)`, with $A_c \subseteq \mathbf{A}$, denotes that the set of actions A_c is intended.

[5] In the deliberation presented in this work we do not distinguish between *intending* and only *proposing* to perform an action. This is discussed in §6.2.5.

Given a set of facts or actions, we assume its conjunction to be the case, respectively proposed. And, if A and B are two sets of either facts or actions, to say that all the elements in A and of B hold, are respectively intended, we write $A \wedge B$.

Thus, for example, we can write: $R \wedge \mathcal{C} \wedge \texttt{intended}(\mathcal{A}) \wedge \Gamma {\mid\!\sim}\ \texttt{side\_effect}(S)$. Meaning that if R and $\mathcal{C}$ are the case, the proposed actions $\mathcal{A}$ will result in the set of side effects S. The rationale as to why $\mathcal{A}$ will cause S is in the background theory Γ. Each agent and knowledge resource defines its own version of Γ, which may contain different rules and reasoning techniques. For example, a basic artificial agent may contain a fixed table with precodified 4-tuples relating the four dimensions $\mathbf{R} \times \mathbf{A} \times \mathbf{S} \times \mathbf{G}$. A slightly more sophisticated artificial agent will define an internal structure to each of the four dimensions with a number of transition rules. A human agent, on the other hand, will use her own reasoning (her own version of Γ and ${\mid\!\sim}$) to reason about the exchanged arguments. However, all these *heterogeneous* agents will have to agree on the syntax and semantics of the exchanged *ProCLAIM* arguments. This is further discussed in §7 and in §10.

Typically the background theory is written as a subscript on the consequence relations: ${\mid\!\sim}_{\Gamma}$. To emphasise that $\mathcal{C}$ and $\mathcal{A}$ are assumed to be the case, *i.e.* that they are contextual information, they are also written as subscripts on the consequence relations: ${\mid\!\sim}_{\mathcal{C}\wedge\mathcal{A}\wedge\Gamma}$. We also take the liberty of omitting the `intended` predicate wrapping the actions.

With these elements and relations the relevance of a set of facts and actions (w.r.t. realising an undesirable goal) can be defined as follows:

Definition 6.1 *Within the context of facts $\mathcal{C}$ and of proposed actions $\mathcal{A}$ a set of facts $R \subseteq \mathbf{R}$ is said to be* ***relevant*** *if one of the following two situations holds:*

- *In circumstances $\mathcal{C}$, if R holds the actions $\mathcal{A}$ will cause an undesirable side effect. Otherwise, if R does not hold, the undesirable side effect is no longer expected to be caused by $\mathcal{A}$ (in circumstances $\mathcal{C}$):*

  – $R {\mid\!\sim}_{\mathcal{C}\wedge\mathcal{A}\wedge\Gamma}\ \texttt{side\_effect}(S)$*;*
  – $\texttt{side\_effect}(S) {\mid\!\sim}_{R\wedge\mathcal{C}\wedge\Gamma}\ \texttt{und\_goal}(g)$ *and*
  – ${\mid\!\not\sim}_{\mathcal{C}\wedge\mathcal{A}\wedge\Gamma}\ \texttt{side\_effect}(S)$.

  Or

- *In circumstances $\mathcal{C}$ actions $\mathcal{A}$ will cause an undesirable side effect. But if R holds, then either the side effect is not expected to be caused by $\mathcal{A}$, or the side effect cannot be deemed as undesirable (*i.e. *the degree to which the side effect realises the undesirable goal is too weak):*

  – ${\mid\!\sim}_{\mathcal{C}\wedge\mathcal{A}\wedge\Gamma}\ \texttt{side\_effect}(S)$ *and*
  – $\texttt{side\_effect}(S) {\mid\!\sim}_{\mathcal{C}\wedge\Gamma}\ \texttt{und\_goal}(g)$

  but either:

- $R {\not\mid\!\sim}_{\mathcal{C} \wedge \mathcal{A} \wedge \Gamma}$ $\texttt{side\_effect}(S)$ *or;*
- $R \wedge$ $\texttt{side\_effect}(S) {\not\mid\!\sim}_{\mathcal{C} \wedge \mathcal{A} \wedge \Gamma}$ $\texttt{und\_goal}(g)$

Note that when it is said that an undesirable side effect is not expected it is strictly in the local context defined by $\mathcal{C}$ and $\mathcal{A}$. This undesirable side effect may well occur for other reasons, *e.g.*, due to other facts not in $\mathcal{C}$ but in $\mathbb{C}_F$.

The definition of a relevant complementary course of actions is as follows:

Definition 6.2 *Within the context of facts $\mathcal{C}$ and of proposed actions $\mathcal{A}$ a set of actions $A_c \subseteq \mathbf{A}$ is said to be **relevant** if the preconditions R_p for its performance hold ($R_p \subseteq \mathbb{C}_F$) and A_c either prevents an undesirable side effect or it causes one.*

That is:

- ${\mid\!\sim}_{\mathcal{C} \wedge \mathcal{A} \wedge \Gamma}$ $\texttt{side\_effect}(S)$ *and*
- $\texttt{side\_effect}(S) {\mid\!\sim}_{\mathcal{C} \wedge \Gamma}$ $\texttt{und\_goal}(g)$ *and*
- $R_p \wedge intended(A_c) {\not\mid\!\sim}_{\mathcal{C} \wedge \mathcal{A} \wedge \Gamma}$ $\texttt{side\_effect}(S)$

Or;

- $R_p \wedge \texttt{intended}(A_c) {\mid\!\sim}_{\mathcal{C} \wedge \mathcal{A} \wedge \Gamma}$ $\texttt{side\_effect}(S)$ *and*
- $\texttt{side\_effect}(S) {\mid\!\sim}_{\mathcal{C} \wedge \Gamma}$ $\texttt{und\_goal}(g)$ *and*
- ${\not\mid\!\sim}_{\mathcal{C} \wedge \mathcal{A} \wedge \Gamma}$ $\texttt{side\_effect}(S)$.

In what follows we will use the above introduced concepts to define the arguments schemes and their associated critical questions to be used in the *Argumentation Layer*. Using these schemes and critical questions participants will submit their arguments, highlighting with each argument the relevant facts and complementary courses of actions. These relevant factors (facts or actions) are the ones that can be added to the arguments' local contexts. Once PAs have submitted all their arguments, and so all the relevant facts and actions have been introduced, the tree of arguments $\mathbb{T}$ is evaluated to resolve whether the main action can safely be performed.

6.2 Protocol-Based Exchange of Arguments

In this section we introduce the argument schemes and their associated critical questions tailored for deliberating over safety critical actions. Each of these argument schemes encodes a particular relation among elements in $\mathbf{R}$, $\mathbf{A}$, $\mathbf{S}$ and $\mathbf{G}$. Arguments instantiating these schemes represent instances of these relations, while their associated CQs question them. Thus, with the exchange of arguments, participants build a subspace of $\mathbf{R} \times \mathbf{A} \times \mathbf{S} \times \mathbf{G}$ tailored to the particular problem at hand. Hence, the deliberation process can be regarded as a mechanism for exploring the relevant facts in $\mathbf{R}$, accounting for the complementary courses of actions in $\mathbf{A}$, guided by the (undesirable) side effects which are highlighted in $\mathbf{S}$ and $\mathbf{G}$. The relevant elements in $\mathbf{R}$ and $\mathbf{A}$ are those that have an impact in $\mathbf{S}$ and $\mathbf{G}$.

The schemes and critical questions will be introduced in a **modular** fashion. We start by introducing a set of assumptions that will help in constructing a basic circuit of six schemes and their associated CQs:

- **Assum_1:** $\mathbf{R}$, $\mathbf{A}$, $\mathbf{S}$ and $\mathbf{G}$ have no internal structure (*e.g.* no taxonomy). These are Assum_1a, Assum_1b, Assum_1c and Assum_1d respectively.

- **Assum_2:** All introduced facts R are in $\mathbb{C}_F$. Arguments must use facts that are in the context of facts $\mathbb{C}_F$.

- **Assum_3:** *a)* All the proposed actions A are in $\mathbb{C}_A$, *b)* they can be performed ($R_p \subseteq \mathbb{C}_F$), *c)* and they do not conflict with other proposed actions (*i.e.* no two or more action are such that if jointly performed they cause an undesirable side effect).

- **Assum_4:** Each $g \in \mathbf{G}$ is such that if the main action will realise g the action is deemed unsafe.

As we relax some of these assumptions we will be extending this circuit of AS and CQs. In §6.2.2 we enrich $\mathbf{R}$ with a taxonomy by introducing a specificity relation. In §6.2.3 we add a defeasible entailment to $\mathbf{R}$ to allow the inherent uncertainty of the facts in $\mathbb{C}_F$ to be accounted for. In §6.2.4 we permit the use of facts not in $\mathbb{C}_F$, in order to account for incomplete information. Finally, in §6.2.5 we discuss other extensions that we are formalising.

6.2.1 Basic Argument Schemes

In this subsection we present the basic protocol-based exchange of arguments consisting of six argument schemes and their associated critical questions by which players participate in the deliberation, introducing new relevant facts and complementary courses of actions.

Each scheme is presented as a four part composite: A set of *preconditions*, the scheme's *body*, its associated *critical questions* and the scheme's *context updating rule* by which the arguments' local contexts ($\mathcal{C}$ and $\mathcal{A}$) are updated. The body of the scheme is itself presented in three complementary representations: a *narrative* version written in natural language; a *formal* version; and the deliberation's dialogue locutions, *i.e.* the content of the `argue` and `challenge` locutions introduced in §5.2.3. Let us start by introducing the first argument scheme, AS1, that sets the deliberation's topic. In fact, this scheme is instantiated at the deliberation's *Open Stage* (§5.2.1) as the *proposal*. This first argument is the root of $\mathbb{T}$.

Let us just introduce some notation, argument scheme $AS1$ proposes the main action under the assumption that Am will cause no undesirable side effect: $\sim \mathtt{undSideEffect}(A_m)$, where $\sim$ denotes the weak negation. Subsequent arguments will attack this assumption by highlighting an undesirable goal or defend this assumption arguing against the realisation of the highlighted an undesirable goal.

AS1	
Preconditions: $R_p \subseteq \mathbb{C}_F$, $A_m \subseteq \mathbb{C}_A, \mathcal{C}$ = {} and $\mathcal{A}$ = {}	
Body:	In circumstances R_p The proposed course of action A_m can safely be performed.
	$R_p \wedge \sim \texttt{undSideEffect}(A_m) \mathrel{|}\!\!\sim_{\Gamma} \texttt{propose}(\mathrm{Am})$
	$\texttt{argue(}\mathcal{C}\texttt{,}\mathcal{A}\texttt{,propose(}R_p\texttt{,}A_m\texttt{));}$
Critical Questions: **CQ1**: Are current circumstances such that an undesirable side effect will be achieved?	
Context Updating Rule: $\mathcal{C}$:= R_p; $\mathcal{A}$:= A_m.	

To illustrate the use of this scheme, let us introduce an example from the transplant scenario. To start with, let us suppose a `lung` of a donor `d` is available (`av_org(d,lung)`) for a potential recipient `r` (`p_recip(r,lung)`). And so the intention is to transplant the lung to this recipient (`transp(r,lung)`). Hence, `av_org(d,lung)`, `p_recip(r,lung)` $\in \mathbb{C}_F$ and `transp(r,lung)`$\in \mathbb{C}_A$. Therefore the initial argument, say A, can be submitted instantiating $AS1$ as follows:

A: `argue({},{},propose({av_org(d,lung),p_recip(r,lung)},`
`{transp(r,lung)}));`[6]

Typically, critical questions associated with a scheme enable agents to attack the validity of the various elements of the scheme and the connections between them. Also, there may be alternative possible actions and side effects of the proposed action [34]. In the particular case of arguments instantiating $AS1$ what can be questioned is whether there is a fact, or set of facts R, in the current circumstances ($R \subseteq \mathbb{C}_F$) that makes the proposed action unsafe. Hence, what can be questioned is the assumption that there are no contraindications for performing the proposed action. That is, critical question $CQ1$, which we denote as $AS1\_CQ1$, can be used.

An answer *no* to this question, implicitly encoded in the assumption of the initial argument, would imply little progress in the deliberation. An answer *yes* to this question

[6] It is worth noting that an artificial agent may represent internally this argument in many forms, for instance in a more standard support-claim argument structure like $<$ {`av_org(d,lung)` $\wedge$ `p_recip(r,lung)`$\wedge$ $\sim$ `undSideEffect(transp(r,lung))` $\Rightarrow$ `propose(transp(r,lung))`, `av_org(d,lung)`, `p_recip(r,lung)`}, `propose(transp(r,lung))` $>$

constitutes an attack on the argument. Thus, for the deliberation to effectively progress, $AS1\_CQ1$ can only be addressed by introducing a contraindication, *i.e.* a set of facts R that will result in the action causing an undesirable side effect. This use of $AS1\_CQ1$ is effected by an argument instantiatingthe scheme $AS2$, and that attacks the argument instantiating $AS1$.

Finally, to illustrate the scheme's context updating rule, note that any reply to argument A will assume as contextual information that there is an available `lung` of a donor `d`, a potential recipient `r` for that organ and that the transplant is intended. That is, $\mathcal{C} = \{\texttt{av\_org(d,lung),p\_recip(r,lung)}\}$ and $\mathcal{A} = \{\texttt{transp(r,lung)}\}$. Needles to say, if the assertion of any of these facts or actions is retracted at the Context Stage, the deliberation concludes.

AS2

Preconditions: $R \subseteq \mathbb{C}_F$, $S \subseteq \mathbf{S}$, $S \neq \emptyset$, $g \in \mathbf{G}$, and
$\mathcal{C}$ and $\mathcal{A}$ the context of facts and actions of the target argument.

Body:

In circumstances $\mathcal{C}$
Because R holds, actions $\mathcal{A}$ will cause a side effect S
which will realise some undesirable goal g.

- ○ $R \mid\!\sim_{\mathcal{C}\wedge\mathcal{A}\wedge\Gamma} \texttt{side\_effect}(S)$; and
- ○ $\texttt{side\_effect}(S) \mid\!\sim_{\mathcal{C}\wedge R\wedge\Gamma} \texttt{und\_goal(g)}$;

$\texttt{argue(}\mathcal{C}\texttt{,}\mathcal{A}\texttt{,contra(R,S,g))}$;

Critical Questions:

CQ1: Are current circumstances such that the stated side effect will not occur?

CQ2: Are current circumstances such that the side effect will not realise the stated goal?

CQ3: Is there a complementary course of action that prevents the achievement of the stated effect?

Context Updating Rule: $\mathcal{C} := \mathcal{C} \cup R$; $\mathcal{A} := \mathcal{A}$.

That is, an argument instantiating **AS2** identifies contraindications R for performing the proposed actions $\mathcal{A}$, in circumstances $\mathcal{C}$.

Continuing with the above example, let us suppose that the donor of the offered lung has *smoking history* (`d_p(d,s_h)`: donor `d` has property `s_h`). Let us suppose, as well, that the donor agent, DA, that offers the lung for transplantation, believes `s_h` to be a contraindication because the lung may be rejected by the recipient, thus realising the undesirable goal `grft_fail(r)`. Hence, DA believes $AS1\_CQ1$ to be the case, and so may want to attack argument A. This can be done by submitting an argument $B1$ (see fig. 6.2), that instantiates $AS2$ as follows:

$B1$: `argue(`$\mathcal{C}$`,`$\mathcal{A}$`,contra({d_p(d,s_h)},{reject(r,lung)},`
`grft_fail(r)));`

Let us now identify $AS2$'s critical questions. That is, which lines of attack can be pursued in order to, for example, attack argument $B1$. For that purpose, let us highlight what is being asserted by an argument instantiating $AS2$, taking into account that $\mathcal{C}$ has been updated (*e.g.* in argument $B1$, $\mathcal{C}$ = {`av_org(d,lung),p_recip(r,lung),` `d_p(d,s_h)`}) while $\mathcal{A}$ remains the same and that *ProCLAIM* arguments only assert a relation among the sets $\mathbf{R}$, $\mathbf{A}$, $\mathbf{S}$ and $\mathbf{G}$:

1. $\mathcal{C} \wedge \mathcal{A} \mid\!\sim_{\Gamma}$ `side_effect(`S`)`; and
2. `side_effect(`S`)` $\wedge\mathcal{C} \mid\!\sim_{\Gamma}$ `und_goal(`g`)`;

Firstly, whether these two relations hold is evaluated first at the *Proxy Level* (§5.1) where the MA validates the incoming arguments and latter at the *Resolution Stage* (§5.2.5) where a relative strength of the accepted arguments is assigned. Secondly, under the assumptions presented at the beginning of this section, the local contexts are such that $\mathcal{C} \subseteq \mathbb{C}_F$ and $\mathcal{A} \subseteq \mathbb{C}_A$(Assum_2 and Assum_3a resp) and thus they are taken to be the case (*e.g.* in argument $B1$ `d_p(d,s_h)` holds). And with Assum_4 we have that if `und_goal(`g`)` holds as consequence of the action this should be deemed unsafe. What can be done to attack an argument instantiating scheme $AS2$ is an update to either $\mathcal{C}$ or $\mathcal{A}$ so that either of the two relations does not hold ($\not\mid\!\sim$ `side_effect(`S`)` or $\not\mid\!\sim$ `und_goal(`g`)`). Since each fact in $\mathcal{C}$ and each action in $\mathcal{A}$ has to be in $\mathbb{C}_F$ and $\mathbb{C}_A$ respectively, and $\mathbb{C}_F$ and $\mathbb{C}_A$ do not allow for inconsistencies, any update on the local contexts has to be truth preserving. Retracting or negating an element of $\mathbb{C}_F$ or $\mathbb{C}_A$ is done at the *Context Stage* and the effect of such moves is discussed in §6.2.4. Since neither $\mathbf{R}$ or $\mathbf{A}$ have an internal structure (we discuss relaxation of Assim_1 in §6.2.2), truth preserving updates on $\mathcal{C}$ or $\mathcal{A}$ can only be done by adding a new sets of (relevant) facts R to $\mathcal{C}$ or complementary courses of actions A_c to $\mathcal{A}$. Therefore, what can be questioned on arguments instantiating scheme $AS2$ is whether there exists a set $R \subseteq \mathbb{C}_F$ such that in the new context $\mathcal{C} \cup R$ the side effect S is no longer expected ($AS2\_CQ1$); or in which the undesirable goal g would not be realised ($AS2\_CQ2$)[7]. Note that $\mathcal{A}$ only appears in the first assertion. Thus, changes in $\mathcal{A}$ ($\mathcal{A} \cup A_c$) can only be proposed in order to argue that the complementary course of action A_c can prevent the side effect S ($AS2\_CQ3$). These three critical questions have only practical use if the appropriate relevant R or A_c are provided. The critical questions $AS2\_CQ1$, $AS2\_CQ2$ and $AS2\_CQ3$ are therefore addressed as attacking arguments respectively instantiating schemes $AS3$, $AS4$ and $AS5$, and so introducing the relevant Rs and A_cs.

[7] That is, given the new context $\mathcal{C} \cup R$ the degree by which S realises g is too weak.

AS3

Preconditions: $R \subseteq \mathbb{C}_F$, S the side effect of the target argument
$\mathcal{C}$ and $\mathcal{A}$ the updated context of facts and actions of the target argument.

Body:

In circumstances $\mathcal{C}$
Because R holds, the side effect S is not expected as caused by $\mathcal{A}$.

$R \mid\!\not\sim_{\mathcal{C}\wedge\mathcal{A}\wedge\Gamma}$ `side_effect`(S)

`argue(`$\mathcal{C}$`,`$\mathcal{A}$`,no_side_effect(R,S));`

Critical Questions:

CQ1: Are current circumstances such that an undesirable side effect will occur?

Context Updating Rule: $\mathcal{C} := \mathcal{C} \cup R$; $\mathcal{A} := \mathcal{A}$.

AS4

Preconditions: $R \subseteq \mathbb{C}_F$, S and g of the target argument replied to
$\mathcal{C}$ and $\mathcal{A}$ the updated context of facts and actions of the target argument.

Body:

In circumstances $\mathcal{C}$
And assuming $\mathcal{A}$ will be performed
It is because R holds, that S does not realises `g`

`side_effect`$(S) \wedge R \mid\!\not\sim_{\mathcal{C}\wedge\Gamma}$`und_goal(g)`

`argue(`$\mathcal{C}$`,`$\mathcal{A}$`,not_realised_goal(R,S,g));`

Critical Questions:

CQ1: Are current circumstances such that the side effect will realise the undesirable goal?

Context Updating Rule: $\mathcal{C} := \mathcal{C} \cup R$; $\mathcal{A} := \mathcal{A}$.

AS5	
Preconditions: $A_c \subseteq \mathbb{C}_A$, $R_p \subseteq \mathbb{C}_F$ preconditions to perform A_c, S of the target argument; $\mathcal{C}$ and $\mathcal{A}$ the updated context of facts and actions of the replied argument.	
Body:	In circumstances $\mathcal{C} \cup R_p$ The complementary course of action A_c Prevents actions $\mathcal{A}$ from causing the side effect S.
	$A_c \wedge R_p \mathrel{\vert\!\sim}_{\mathcal{C}\wedge\mathcal{A}\wedge\Gamma}$ `side_effect`(S)
	`argue(`$\mathcal{C}$`,`$\mathcal{A}$`,preventive_action(A`$_c$`,R`$_p$`,S));`
Critical Questions: **CQ1**: Are current circumstances such that an undesirable side effect will be achieved?	
Context Updating Rule: $\mathcal{C} := \mathcal{C} \cup R_p$; $\mathcal{A} := \mathcal{A} \cup A_c$.	

Figure 6.2 illustrates the use of these three argument schemes. Argument $B2$, instantiating $AS3$, attacks $B1$, indicating that because the donor does not have a Chronic Obstructive pulmonary disease (R = {`d_p(d,no_copd)`}) the donor's smoking history is no longer a contraindication. Argument $C2$, instantiating $AS4$, attacks argument $C1$, indicating that because the potential recipient already has HIV (`p_r_p(r,hiv)`), the infection cannot be deemed as a severe infection caused by the lung transplant[8]. Finally, argument $D2$ illustrates an instantiation of scheme $AS5$ proposing to administrate *penicillin* to the recipient (`treat(r,penicillin)`) of a lung of a donor whose cause of death was a *streptococcus viridans endocarditis* (`d_p(d,sve)`) so as to prevent an infection of that same bacteria (`r_p(r,svi)`). The set of preconditions R_p in argument $D2$ is empty. It is assumed in this scenario that there is an availability of penicillin and means to administrate the antibiotic. Otherwise such facts should be added in the set of preconditions.

Note that the attacks made on argument A by $B1$, $C1$ and $D1$ are asymmetric (one way attacks), whereas the attacks on $B1$, $C1$ and $D1$ made respectively by $B2$, $C2$ and $D2$ are symmetric. The reason for these differing attack relations is that in the former case, arguments *in favour* of the proposed action are always based on an assumption that no contraindication exists; an assumption that is undermined by the attacking arguments. (e.g., $D1$ undermines A's default assumption of no contraindication by identifying a contraindication ($d\_p(d, svi)$). In the second case, complementary course of action are proposed to prevent undesirable side effects, where whether or not such prevention will be realised may still be a matter of debate. Hence, $D2$ attacks $D1$ by proposing $treat(r, penicillin)$ to prevent

[8] Given that `p_r_p(r,hiv)`, the degree by which `side_effect(r_p(r,hiv))` realises a `sev_inf(r)` is too weak.

ID	Type	Argument
A	AS1	argue({},{}, propose({av_org(d,lung), p_recip(r,lung)}, {transp(r,lung)}))
B1	AS2	argue(C,A, contra({d_p(d,s_h)}, {reject(r,lung)}, graft_failure))
B2	AS3	argue(C,A, no_side_effect({¬d_p(d,copd)}, {reject(r,lung)}))
C1	AS2	argue(C,A, contra({d_p(d,hiv)}, {r_p(r,hiv)}, sever_infect))
C2	AS4	argue(C,A, not_realised_goal({p_r_p(r,hiv)}, {r_p(r,hiv)}, sever_infect))
C3	AS6	argue(C,A, contra({r_p(r,superinf)}, sever_infect))
D1	AS2	argue(C,A, contra({d_p(d,sve)}, {r_p(r,svi)}, sever_infect))
D2	AS5	argue(C,A, preventive_action({treat(r,penicillin)}, {r_p(r,svi)}))
D3	AS2	argue(C,A, contra({p_r_p(r,pen_allergy)}, {r_p(r,anaphylaxis)}, sever_infect))

a)

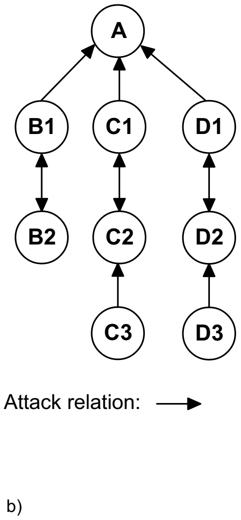


Figure 6.2: Example of three lines of argumentation structured reasoning: the B arguments which address the donor's smoking history ($\texttt{d\_p(d,s\_h)}$), the C arguments addressing the donor's HIV ($\texttt{d\_p(d,hiv)}$); and the D arguments which address the fact that the donor's cause of death was *streptococcus viridans endocrditis* ($\texttt{d\_p(d,sve)}$) which may result in the recipient of the lung contracting a *streptococcus viridans infection* ($\texttt{r\_p(r,svi)}$). Each argument's $\mathcal{C}$ and $\mathcal{A}$ is updated according to the schemes' context updating rules.

$reject(r, lung)$, where the efficacy of this preventative measure may still be debatable (implicitly then, $D2$ and $D1$ disagree on whether $d\_p(d, svi)$ is or not a contraindication). This disagreement is made explicit with a symmetric attack. To resolve whether the transplant is safe or not will require a decision as too whether or not $d\_p(d, svi)$ is a contraindication, that is, whether $D2$ is preferred to $D1$ or vice versa (this is further discussed in §8). Note however, that if a fourth argument $D3$ is submitted attacking argument $D2$, by indicating for instance that the potential recipient is allergic to penicillin, such an attack will again be asymmetrically directed on an assumption of argument $D2$ that no other contraindication exists. And so argument $D2$ does not defend itself against (i.e attack) $D3$ as would be the case with a symmetric attack.

Let us return to schemes $AS3$, $AS4$ and $AS5$ in order to identify their CQs. An argument instantiating schemes $AS3$ or $AS4$ introduces a new set of relevant facts R. An argument instantiating $AS5$ introduces a complementary course of actions A_c with a possibly empty set of preconditions R_p. At this stage R, R_p and A_c are taken to be the case (resp. intended), under assumptions Assum_2 and Assum_3a. As with arguments instantiating $AS2$, whether R and A_c are *relevant* is decided first at the Proxy Level (*e.g.* should argument $B2$ be accepted, *i.e.*, does $\{\texttt{d\_p(d,no\_copd)}\}\mathrel{|\!\!\sim}_{\mathcal{C}\wedge\mathcal{A}\wedge\Gamma}$ $\texttt{side\_effect}(\{reject(r, lung)\})$ make sense) and latter at the Resolution Stage (*e.g.* does argument $B2$ defeats argument $B1$, *i.e.*, would {reject(r,lung)} be prevented).

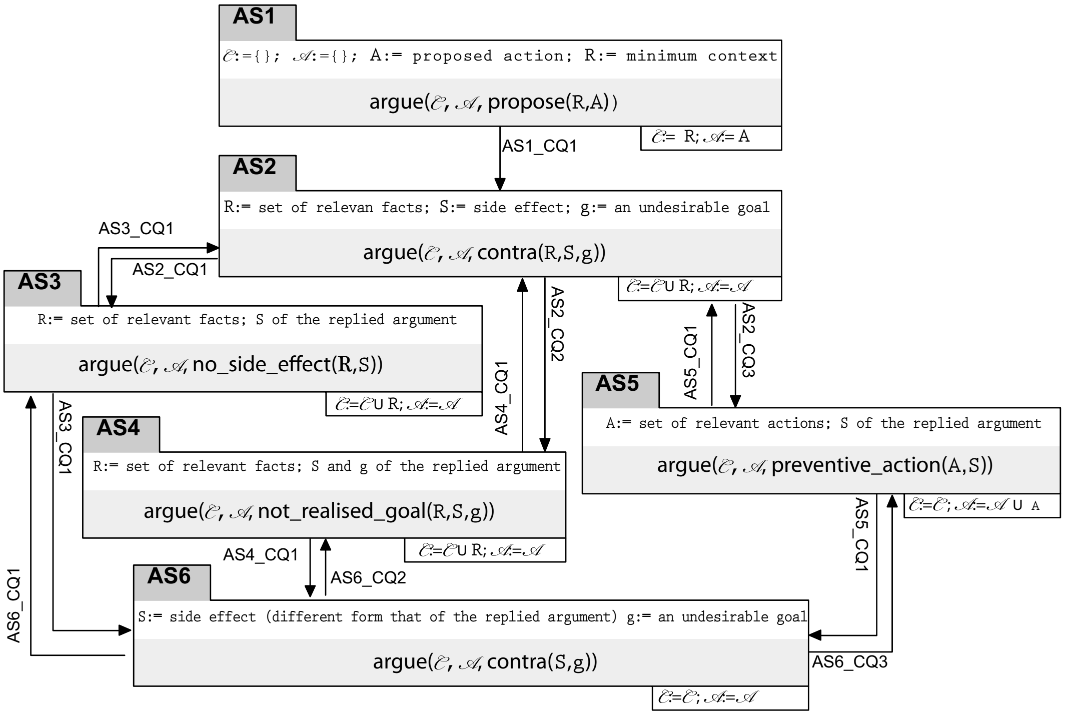

AS1
$\mathcal{C}$:={}; $\mathcal{A}$:={}; A:= proposed action; R:= minimum context
argue($\mathcal{C}$, $\mathcal{A}$, propose(R,A))
$\mathcal{C}$:= R; $\mathcal{A}$:= A
AS1_CQ1
AS2
R:= set of relevan facts; S:= side effect; g:= an undesirable goal
argue($\mathcal{C}$, $\mathcal{A}$, contra(R,S,g))
$\mathcal{C}$:=$\mathcal{C}$ ∪ R; $\mathcal{A}$:=$\mathcal{A}$
AS3_CQ1
AS2_CQ1
AS3
R:= set of relevant facts; S of the replied argument
argue($\mathcal{C}$, $\mathcal{A}$, no_side_effect(R,S))
$\mathcal{C}$:=$\mathcal{C}$ ∪ R; $\mathcal{A}$:=$\mathcal{A}$
AS4_CQ1
AS2_CQ2
AS5_CQ1
AS2_CQ3
AS5
A:= set of relevant actions; S of the replied argument
argue($\mathcal{C}$, $\mathcal{A}$, preventive_action(A,S))
$\mathcal{C}$:=$\mathcal{C}$; $\mathcal{A}$:=$\mathcal{A}$ ∪ A
AS3_CQ1
AS4
R:= set of relevant facts; S and g of the replied argument
argue($\mathcal{C}$, $\mathcal{A}$, not_realised_goal(R,S,g))
$\mathcal{C}$:=$\mathcal{C}$ ∪ R; $\mathcal{A}$:=$\mathcal{A}$
AS5_CQ1
AS4_CQ1
AS6_CQ2
AS6_CQ1
AS6
S:= side effect (different form that of the replied argument) g:= an undesirable goal
argue($\mathcal{C}$, $\mathcal{A}$, contra(S,g))
AS6_CQ3
$\mathcal{C}$:=$\mathcal{C}$; $\mathcal{A}$:=$\mathcal{A}$

Figure 6.3: Argument Schemes connected via their associated Critical Questions

An argument, say Arg, that instantiates scheme $AS3$, $AS4$ or $AS5$, assumes (as in the case of the first submitted argument) that no (other) contraindication exists for performing the main action. This assumption is questioned by $AS3\_CQ1$, $AS4\_CQ1$ and $AS5\_CQ1$. As in $AS1$, such critical questions can only be addressed as attacks identifying the contraindications and the associated undesirable side effects. Such attacks can thus be embodied by arguments instantiating scheme scheme $AS2$, analogous to attacks on the first submitted argument by arguments instantiating $AS2$. However, this time, as a way to defend the main action's safety, Arg introduces a new set of factors (facts or actions) which themselves may warrant, respectively cause, some undesirable side effect. That is, this time, an attack can be made via $AS3\_CQ1$, $AS4\_CQ1$ and $AS5\_CQ1$ without having to introduce a new set of facts. Such attacks are embodied by argument scheme $AS6$ which differs from $AS2$ in that it does not require introducing an additional set of relevant facts R:

AS6

Preconditions: $S \subseteq \mathbf{S}$, non-empty and **different** from the replied argument's stated effect, $g \in \mathbf{G}$;
$\mathcal{C}$ and $\mathcal{A}$ the updated context of facts and actions of the replied argument.

Body:

In circumstances $\mathcal{C}$
The actions $\mathcal{A}$ will cause a side effect S
which will realise some undesirable goal g.

- ○ $\mid\!\sim_{\mathcal{C}\wedge\mathcal{A}\wedge\Gamma}$ `side_effect`(S); and
- ○ `side_effect`$(S) \mid\!\sim_{\mathcal{C}\wedge\mathcal{A}\wedge\Gamma}$ `und_goal(g)`;

`argue(`$\mathcal{C}$`,`$\mathcal{A}$`,contra(S,g));`

Critical Questions:

CQ1: Are current circumstances such that the stated side effect will not be achieved?

CQ2: Are current circumstances such that the achieved side effect will not realise the stated goal?

CQ3: Is there a complementary course of action that prevents the achievement of the stated effect?

Context Updating Rule: $\mathcal{C}$:= $\mathcal{C}$; $\mathcal{A}$:= $\mathcal{A}$.

We can continue with our medical example to illustrate the use of schemes $AS2$ and $AS6$ in order to attack arguments instantiating schemes $AS3$, $AS4$ or $AS5$ (see fig. 6.2). Suppose, for instance, that the recipient to whom the lung is intended is *allergic to penicillin*. Thus, if as a way to prevent the recipient's bacterial infection penicillin is administered ($D2$), the allergic reaction may be quite severe, (*anaphylaxis*). Such an argument against the action's safety is embodied by $D3$ which instantiates scheme $AS2$. To illustrate the use of scheme $AS6$, let us continue with the argumentation line A, $C1$ and $C2$, where it has been argued that the lung may safely be transplanted despite the donor having *HIV* because the potential recipient already has the same viral infection. It is currently believed that in most cases such transplants will cause a *superinfection* [242], which is an uncontrolled, severe infection. Note that no new factors were introduced in order to attack argument $C2$. Thus, such an attack can be embodied by an argument $C3$ that instantiates $AS6$. In this basic circuit of schemes and critical questions, $AS6$'s critical questions are the same as those for $AS2$.

Figure 6.3 depicts the circuit of argument schemes connected via their associated critical questions presented in this section. In the following subsections we relax some of the assumptions introduced in this subsection so as to address required extensions to this basic circuit.

6.2.2 Accounting for Specificity

Let us suppose now that a DA offers for transplantation the lung of a donor with a history of cancer (`d_p(d,h_cancer)`). The DA herself may argue that with such record the recipient will result having as a side effect cancer. As depicted in figure 6.4 this argument ($E1$) can be instantiated using scheme $AS2$. Let us suppose as well that the DA have added to $\mathbb{C}_F$ the fact `d_p(d,h_nonmel_skin_c)` meaning that the donor had a nonmelanoma skin cancer. A history of cancer is in general an excluding criteria for being a donor. However, for some kind of past malignancies, such as nonmelanoma skin cancer, the risk of transmitting the malignancy to the recipient is believed to be marginal [126]. Let us suppose the RA believes that to be the case and would wish to argue that for this particular type of cancer the transplant is safe. At first sight it may seam that this argument could be constructed by instantiating scheme $AS3$ with R = {`d_p(h_nonmel_skin_c)`} being the new relevant set of facts. And so updating the local context of facts to be:

$\mathcal{C}$ = {`av_org(d,lung),p_recip(r,lung),d_p(d,h_cancer),`
`d_p(d,h_nonmel_skin_c)`}

Although clearly $\mathcal{C}$ holds ($\mathcal{C} \subseteq \mathbb{C}_F$), there is a bit of information that despite being important is not captured if $AS3$ is to be used. That is, `d_p(d,h_cancer)` and `d_p(d,h_nonmel_skin_c)` are not independent facts, the latter is a subclass of the former. Furthermore, there is an implicit assumption that donor had a history nonmelanoma skin cancer and no other type of cancer.

In order to account for this we need first to relax Assum_1a by associating to $\mathbf{R}$ a relation of specificity $\prec$ so as to account for the fact that, for instance, {`d_p(d,h_nonmel_skin_c)`} $\prec$ {`d_p(d,h_cancer)`}.

Having defined a taxonomy in $\mathbf{R}$ the circuit of schemes and CQs is extended. The CQs of the kind – *Are the current circumstances such that...?* – (*i.e.* $AS2\_CQ1$, $AS2\_CQ2$, $AS3\_CQ1$, $AS4\_CQ1$, $AS5\_CQ1$, $AS6\_CQ1$ and $AS6\_CQ2$) can now be embodied as an attack not only by schemes $AS2$, $AS3$ and $AS4$ but also by their *specific* versions $AS2s$, $AS3s$ and $AS4s$. Below we introduce only scheme $AS3s$, schemes $AS2s$ and $AS4s$ are defined analogously:

AS3s	
Preconditions: $R_g \subseteq \mathcal{C}$, $R_s \subseteq \mathbb{C}_F$, S of the replied argument $\mathcal{C}$ and $\mathcal{A}$ the context of facts and actions of the replied argument.	
Body:	Because R_s, a particular case of R_g, holds in circumstances $(\mathcal{C} - R_g)$ the side effect S is not expected as caused by $\mathcal{A}$.
	○ $R_s \prec R_g$ ○ $R_s \not\mid\!\sim_{(\mathcal{C}-R_g)\wedge\mathcal{A}\wedge\Gamma}$ `side_effect(`S`)`
	`argue(`$\mathcal{C}$`,`$\mathcal{A}$`,no_side_effect(replace_s(`R_g, R_s`),`S`));`
Critical Questions: Same as $AS3$	
Context Updating Rule: $\mathcal{C}$:= $(\mathcal{C} - R_g) \cup R_s$; $\mathcal{A}$:= $\mathcal{A}$.	

The main change in these new schemes is the way the local context of facts $\mathcal{C}$ is updated. Instead of introducing an additional set of facts R (as it is the case with $AS2$, $AS3$ and $AS4$) a subset $R_g \subseteq \mathcal{C}$ is replaced by a more specific set of facts R_s ($R_s \prec R_g$). In this way, it is made explicit that R_g does not holds by itself, independent of R_s. Rather, R_g is the case only because R_s holds, since R_s entails R_g. Thus, for example, `d_p(d,h_cancer)` would hold only because `d_p(d,h_nonmel_skin_c)` is the case.

To continue with our example, argument $E1$ can now be attacked by an argument $E2$ instantiating scheme $AS3s$ as follows:

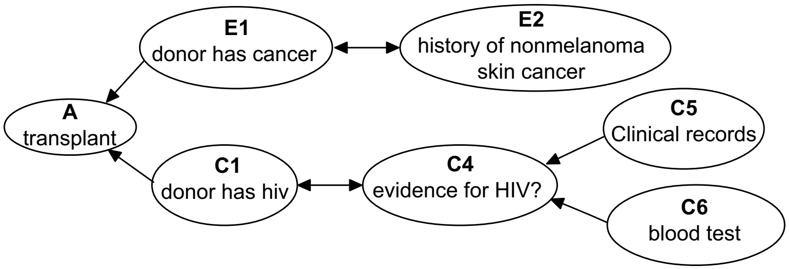
```
E2: argue(
   {av_org(d,lung),p_recip(r,lung),d_p(d,h_cancer)},
   {transp(r,lung)},
   no_side_effect(
      replace_s({d_p(d,h_cancer)}, {d_p(d,h_nonmel_skin_c)}),
      {r_p(r,cancer)}));
```

With its updated local context of facts being:

$\mathcal{C}$ = {`av_org(d,lung),p_recip(r,lung),d_p(d,h_nonmel_skin_c)`}

6.2.3 Accounting for Uncertainty

In §5.2.2 we have said that any dispute regarding whether a fact in $\mathbb{C}_F$ holds or not should be resolved outside *ProCLAIM*. However, it is still important for the decision making to



ID	Type	Argument
A	AS1	argue({},{}, propose({av_org(d,lung), p_recip(r,lung)}, {transp(r,lung)}))
E1	AS2	argue(C,A, contra({d_p(d,cancer)}, {r_p(r,cancer)}, cancer))
E2	AS3s	argue(C,A, no_side_effect(replace_s({d_p(d,cancer)},{d_p(d,h_nonmel_skin_c)}), {reject(r,lung)}))
C1	AS2	argue(C,A, contra({d_p(d,hiv)}, {r_p(r,hiv)}, sever_infect))
C4	AS2_CQ1	challenge(evidence(d_p(d,hiv)))
C5	AS2ev	argue(C,A, contra(replace_ev(d_p(d,hiv),{clin_rec(d,hiv)}), {r_p(r,hiv)}, sever_infect))
C6	AS2ev	argue(C,A, contra(replace_ev(d_p(d,hiv),{test(d,blood,hiv)}), {r_p(r,hiv)}, sever_infect))

Figure 6.4: Example illustrating the use of argument scheme AS3s and of a challenge.

account for the evidence that supports facts asserted within the *Argumentation Layer*. Thus participants should be able to request for and provide such evidence. For example, the RA may want to know the evidence that supports the fact that the donor has HIV.

To enable this Assum_1a has to be relaxed by associating to $\mathbf{R}$ a defeasible consequence relation $\mid\!\sim_{ev}$ where $Ev\mid\!\sim_{ev}Fact$ indicates that a set of facts $Ev \subseteq \mathbf{R}$ is evidence in support of the fact $Fact \in \mathbf{R}$. And so, for example {`clin_rec(d,hiv)`} $\mid\!\sim_{ev}$`d_p(d,hiv)` indicating that donor's clinical records support the fact that the donor has HIV.

Secondly, the circuit of schemes and CQs has to be extended so that argument schemes that introduce relevant set of facts R[9], for each asserted fact $r_i \in R$, there is an associated CQ of the form – *Is there evidence to believe r_i is the case?* –. Now, this CQ is indeed intended to question r_i so that participants have to provide evidence in its support. However, it is not intended for participants to argue that r_i is false, for this should be resolved outside *ProCLAIM*, and if resolved that $\neg r_i$ is the case, $\mathbb{C}_F$ should be updated. Hence, these CQs are not formalised as attacking arguments asserting $\neg r_i$, but only as challenge locutions questioning the evidence for r_i:

challenge(evidence(r_i))

In reply of these challenges is expected an argument that would provide the evidence, a set of facts in support of r_i, that is a set $Ev \subseteq \mathbb{C}_F$, such that $Ev\mid\!\sim_{ev}r_i$. Therefore, if a challenge

[9] These are schemes $AS2$, $AS2s$, $AS3$, $AS3s$, $AS4$ and $AS4s$.

is directed on argument $C1$ as:

```
challenge(evidence(d_p(d,hiv)))
```

The supporting set of facts that may meet the challenge may well be $\{$`clin_rec(d,hiv)`$\}$.

The purpose of these CQs is to allow bringing in the evidence on which the introduced facts are based. In so doing the inherent uncertainty of the facts conforming to the circumstances in which the decision making takes place is made explicit. In this way, decisions are made accounting for this uncertainty, which may, of course, motivate further enquiries in order to make more informed decisions. For example, doctors may proceed to perform a serological (blood) test on the donor in order to have more conclusive evidence on whether the donor does actually have `HIV`. However, while the results of any such enquiry can be fed into *ProCLAIM*'s deliberation by updating $\mathbb{C}_F$, the actual enquiry is not formalised by *ProCLAIM*.

As stated above these CQs are associated to any argument scheme that defines the introduction of a new set of facts, *i.e.* to schemes $AS2$, $AS2s$, $AS3$, $AS3s$, $AS4$ and $AS4s$. Here we present only scheme $AS2ev$ which should be instantiated to construct an argument in reply to a challenge made on an argument instantiating $AS2$ or $AS2s$. The other schemes ($AS3ev$ linked to $AS3$ and $AS3s$ and scheme $AS4ev$ linked to $AS4$ and $AS4s$) are defined analogously:

AS2ev

Preconditions: r_i the questioned fact, $R_{ev} \subseteq \mathbb{C}_F$, S and g of the argument being challenged, and $\mathcal{C}$ and $\mathcal{A}$ its updated context of facts and actions.

Body:

R_{ev} is evidence for r_i being the case, and such that
in circumstances $(\mathcal{C} - \{r_i\}) \cup R_{ev}$
actions $\mathcal{A}$ will cause a side effect S
which will realise some undesirable goal g.

- ○ $R_{ev} \mid\!\sim_{ev} r_i$
- ○ $R_{ev} \mid\!\sim_{(\mathcal{C} - \{r_i\}) \wedge \mathcal{A} \wedge \Gamma}$ `side_effect`(S); and
- ○ `side_effect`$(S) \mid\!\sim_{R_{ev} \wedge \mathcal{C}_i \wedge \Gamma}$ `und_goal(g)`

`argue(`$\mathcal{C}$`,`$\mathcal{A}$`,contra(replace_ev(`r_i, R_{ev}`),`S`,`g`));`

Critical Questions: Same as $AS2$ and $AS2s$ to which we now add the CQs

CQ4$_i$: Is there evidence to believe r_i is the case? ($r_i \in R$,
R the new introduced set of facts)

Context Updating Rule: $\mathcal{C}$:= $(\mathcal{C} - \{r_i\}) \cup R_{ev}$; $\mathcal{A}$:= $\mathcal{A}$.

Note that an argument instantiating scheme $AS2ev$ not only provides the evidence (R_{ev}) supporting the challenged fact, but its claim is that if the asserted fact is replaced by the evidence on which it is based on the same undesirable side effects will be caused (see figure 6.4.). Analogously arguments instantiating scheme $AS3ev$ will claim that the side effect is not expected and; arguments instantiating scheme $AS4ev$ will claim that the side effect are not undesirable in this updated circumstances.

The lack of evidence to support a challenged fact may motivated participants to get the required evidence during the deliberation (*e.g.* perform a serological test on the donor: `test(d,blood,hiv)`). However, it may well be the case that such evidence cannot be acquired, so leaving a challenge weakly replied, or even unreplied. This may lead PAs to retract the challenged fact and so subtract it from $\mathbb{C}_F$, in which case, the challenged argument becomes *hypothetical*, and the challenge is removed from $\mathbb{T}$. Of course if an unknown fact eventually becomes known to be false, the argument is overruled. We discuss all this in the next subsection.

Note that because of the collaborative setting in *ProCLAIM*, the fact that an argument is challenged does not imposes any commitment or *burden* on the agents that submitted or endorsed the challenged argument. As discussed in §2.2 typically when an argument is challenged and left unreplied it is deemed defeated[10]. In *ProCLAIM*, having an unreplied challenge only highlights the uncertainty under which a decision has to be taken. Whether $\mathbb{T}$ is left with uncertain or unknown facts, decision makers will still have to decide what to do. Having resolved which the preferred arguments are in $\mathbb{T}$, if the safety of the action amounts to deciding whether some uncertain and/or unknown facts are the case or not, such resolution would plausibly aim to assess the likelihood of these facts being the case, accounting for the risk involved in them being or not the case. While *ProCLAIM* aims to identify the relevant facts and the risk involved in them being or not the case, by indicating the possible undesirable side effects, it is not intended for addressing the resolution process of weighting likelihood versus risk. We continue this discussion in §8, where we describe the model's argument evaluation process.

6.2.4 Accounting for Incomplete Information

Players may start the deliberation with a set of facts believed to be the case, $\mathbb{C}_F$, and during the argumentation process realise that some potentially relevant information, say r, is miss-

[10] In [101] a more detailed study of the effect of a challenge is made identifying in which condition a challenge has the effect of shifting the burden of proof, see §2.2

ing. That is, $\neg r, r \notin \mathbb{C}_F$. But still, even if some facts are unknown, a decision needs to be made on whether or not to perform the proposed action. Decision makers should be made aware that potentially relevant information is missing. To account for this situation, the argumentation circuit is extended so that participants can submit arguments that introduce a set of fact R as relevant, despite $R \nsubseteq \mathbb{C}_F$. That is, while it is argued that R is relevant, it is unknown whether it holds or not. In that way, participants can make explicit that some data, presumed to be relevant, is missing. And so, they can submit *hypothetical* arguments. Arguments of the form *–If R were the case, then...–*.

These hypothetical arguments are formalised in exactly the same manner as those presented above, the only difference is that we now have relaxed the precondition that facts used in an argument must be in $\mathbb{C}_F$. That is, we relax the assumption Assum_2. In general, updates at the *Argumentation Layer* can be made independently from to those at the Context Stage, and vice versa. This independence results in the definition of three types of arguments:

Definition 6.3 *Suppose $\mathcal{C}$ is the updated local context of facts of an argument Arg, then:*

- *If $\mathcal{C} \subseteq \mathbb{C}_F$, Arg is a **factual argument**.*
- *If $\exists r \in \mathcal{C}$ s.t. $\neg r \in \mathbb{C}_F$, Arg is a **overruled argument**.*
- *Otherwise, Arg is a **hypothetical argument**.*

The arguments presented thus far are all factual, as a precondition we required that their local contexts $\mathcal{C}$ would be in $\mathbb{C}_F$. We have now added the possibility of hypothetical and overruled arguments. Broadly speaking, hypothetical arguments allows PAs to stress the need to check whether or not a relevant facts holds. Overruled arguments indicate that these highlighted facts were contemplated but deemed false. Of course, overruled arguments do not change the acceptability status of the arguments they attack.

To illustrate the requirement for hypothetical arguments, let us introduce a new organ acceptability criterion from [143]: "*For pancreas transplantation, guidelines suggest that donor age should be less than 45 yr; nonetheless, using pancreas with good appearance on inspection after retrieval from donors aged 45-62 yr; can achieve the same graft survival as pancreas from donors aged under 45 ys.*". Hence, if a donor is elderly (over 45 years, for the pancreas case) and her pancreas is transplanted it is likely that it will be rejected, and so realising a graft failure. Unless, the pancreas has *good appearance*. However, in order to check the pancreas' appearance, the organ must first be retrieved. Hence, the transplant should have been deemed safe, at least provisionally.

Let us suppose that a pancreas of a 60 year old donor is available with the donor having `hcv`. Suppose the DA offers the pancreas (argument A, see figure 6.5) and argues that: *1)* because the donor is elderly, the recipient will reject the organ (argument $G1$, instantiating scheme $AS2$), and *2)* that the donor's `hcv` is a contraindication (argument $H1$, instantiating $AS2$), unless the recipient already has this same infection (hypothetical argument $H2$, instantiating $AS4$). Suppose that, in response to DA's submitted arguments the RA adds to $\mathbb{C}_F$ the fact `p_r_p(r,hcv)` (the recipient has `hcv`) and so making argument $H2$ factual.

ID	Type	Argument
A	AS1	argue({},{}, propose({av_org(d,pancreas), p_recip(r,pancreas)}, {transp(r,pancreas)}))
G1	AS2	argue(C,A, contra({d_p(d,elderly)}, {reject(r,pancreas)}, graft_failure))
G2	AS3	argue(C,A, no_side_effect({**o_p(d,pancreas,good_app)**}, {reject(r,lung)}))
H1	AS2	argue(C,A, contra({d_p(d,hcv)},{r_p(r,hcv)}, sever_infect))
H2	AS4	argue(C,A, not_realised_goal({p_r_p(r,hcv)}, {r_p(r,hcv)}, sever_infect))

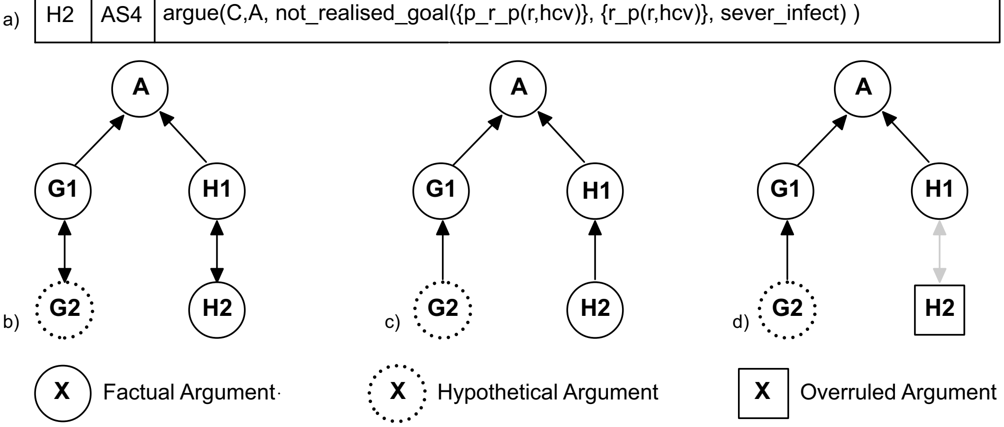


Figure 6.5: Example illustrating the use of hypothetical arguments.

Also, let us suppose the RA submits the hypothetical argument $G2$ that instantiates $AS3$ as follows:

$G2$ = argue($\mathcal{C},\mathcal{A}$, no_side_effect({o_p(d,pancreas, good_app)},
{reject(r,pancreas)}))

with o_p(d,pancreas, good_app) indicating that the donor's pancreas has good appearance. Argument $G2$ can only become factual once the organ is retrieved. Taking this into account, and supposing arguments $G2$ and $H2$ are deemed preferred to $G1$ and $H1$ respectively (fig. 6.5 *c.*), the pancreas will be deemed suitable for this recipient, subject to the organ's appearance on retrieval. That is, if after retrieval o_p(d,pancreas, good_app) holds, the organ can be transplanted, otherwise the transplant should not be performed, argument $G2$ would become overruled.

Note that, if the potential recipient does not have hcv (¬p_r_p(r,hcv) $\in\mathbb{C}_F$), the transplant should be deemed unsafe, irrespective of the pancreas' appearance (fig. 6.5 *d.*). Or similarly, if $H1$ would have been submitted as a hypothetical (it is unknown whether the donor has hcv or not) and $H2$ as factual, what becomes irrelevant, for deciding the action's safety, is whether the donor has or not hcv. Namely, hypothetical and factual arguments together indicate which of the unknown facts are worth checking to see whether they hold.

The independence between the elements of $\mathbb{C}_F$ and $\mathcal{C}$ makes each argument potentially factual, overruled or hypothetical. To allow for such independence we have relaxed the precondition that each additional set of facts R must be in $\mathbb{C}_F$. Because we have defined $\mathbb{C}_F$ such that it has to be a consistent set of facts ($\mathbb{C}_F \nvdash_\Gamma \bot$) this precondition enforced that each argument's context has to be, in turn, consistent. To preserve such a property with the hypothetical arguments, we must ensure that each additional set of facts R is consistent with the elements of the argument's local context $\mathcal{C}$. To do so, to schemes $AS2$, $AS2s$, $AS2ev$, $AS3$, $AS3s$, $AS3ev$, $AS4$, $AS4s$, $AS4ev$, we add the precondition:

The introduced set of relevant facts R must be such that $R \cup \mathcal{C} \nvdash_\Gamma \bot$

Hypothetical arguments have mainly been studied in the legal domain, where the argumentation may involve sophisticated reasoning intended to identify inconstancies in one of the parties' hypothetical arguments [30, 84, 42]. The approach taken here is of course much more humble. It is only intended to make visible facts not known to be the case but which should still be taken into account as being potentially relevant for the decision making. In a somewhat similar way, in [163] hypothetical arguments can be included in an argumentation intended for a classical decision making. In this work, the more hypothetical elements the arguments contains the weaker their relative strength will be.

6.2.5 Discussing Further Extension

There are a number of extensions that can be propose to this circuit of schemes and CQs. Any extension involves *1)* identify the motivating set of examples that needs to be addressed; *2)* relax the appropriate assumptions and finally; *3)* define the procedure through schemes and CQs for capturing the right relation among the sets $\mathbf{R}$, $\mathbf{A}$, $\mathbf{S}$ and $\mathbf{G}$ while appropriately updating the sets $\mathcal{C}$ and $\mathcal{A}$. Each such procedure, argument scheme, must be motivated by a change in the assessment on the main action's safety (within the local contexts of facts an actions). In this subsection we describe a few extensions we are currently formalising.

The first required extension is intended to allow PAs to point at actions that are incompatible across different local contexts of actions. Take for example two complementary actions A_{c1} and A_{c2} that are proposed each to mitigate or prevent different side effects highlighted in a different branch of $\mathbb{T}$. Each action corresponds to a different local contexts: say $< \mathcal{C}_1, \mathcal{A}_1 >$ and $< \mathcal{C}_2, \mathcal{A}_2 >$. Suppose that A_{c1} and A_{c2} are such that when performed together they cause an undesirable side effect. Firstly to address this example assumption Assum_3c has to be relaxed, so that complementary actions can be deemed in conflict. Secondly a procedure must be defined by which an undesirable side effect is caused when the A_{c1} and A_{c2} are jointly performed. This suggest that the update of the local contexts $< \mathcal{C}_1, \mathcal{A}_1 >$ and $< \mathcal{C}_2, \mathcal{A}_2 >$ is for them to be merged, capturing the fact that these local context are no longer *independent*.

Another extension related with actions involves making a distinction between *intending* and merely *proposing*/suggesting an action. For example, it may seam reasonable that while a RA can argue that he *intends* to treat the recipient with antibiotics to prevent a certain infection, the DA can only *suggest* treatments on the recipient. This can be formalised in a similar fashion as we did in §6.2.4 to address the problem of incomplete information.

Relaxing Assum_3a, so that an argument instantiating scheme $AS5$ can use complementary actions that are not in $\mathbb{C}_A$, redefine appropriately the $AS5$'s preconditions and identifying which are the factual, hypothetical and overruled arguments.

The last extension we discuss here is intended to allow addressing the fact that in some circumstances any alternative to performing the main proposed action will derive in more undesirable consequences than the side effects caused by the proposed action. Thus, PAs should be able to question the degree of undesirability of goals. Questioning, for example, whether `cancer` is undesirable enough as a side effect of a organ transplant when any alternative to the organ transplant will result in the death of the potential recipient. To address this example Assum_1d must be relaxed by associating to $\mathbf{G}$ a relation of undesirability, next Assum_4 needs to be relaxed so that not any realised $g \in \mathbf{G}$ is reason enough so as to abort the proposed action. Finall the appropriate procedure has to be defined.

6.3 Discussion

In this chapter we have developed a circuit of argument schemes connected via their critical questions, which aim to capture reasoning patterns useful to argue over whether or not a given action can safely performed. We tailor this reasoning to circumstances where the action's desirability in *ideal* circumstances is beyond dispute. The question is then, whether the circumstances in which the action is to be performed are indeed ideal or there exist some contraindications. That is, whether there are facts because of which the proposed action is believed to cause severe undesirable side effects that motivates not performing it.

We start by introducing the structure of arguments in *ProCLAIM*, which formalisation is motivated by the scheme for action proposal of Atkinson *et al.* [34]. We arrange the arguments' internal structure so that each introduced set of facts or actions must be relevant (it must shift the action's safety assessment, at least within the local contexts) and it is highlighted as such, which is important for the case comparison at the CBRc reasoning cycle. Nonetheless, in defining the arguments' internal structure we leave implicit much of the rational as to *why* some consequences of the main action will or will not be arrived to in certain circumstances (*e.g.* why a lung transplant may end up in a graft failure if the donor has smoking history). We belief this is well motivated for the deliberations *ProCLAIM* is intended (which are time constraint, possibly under stress and where participants are domain experts), where the dominant question that needs to be resolved is whether or not the proposed action will bring about undesire side effects in the current circumstances[11]. Thus, we believe that the underlying rationale of the cause-effect relations used in a deliberation can be left outside the argumentation. This is mainly because, while participant agents may disagree about any stated cause-effect relation, as illustrated in this chapter, as domain experts they require no explanations to understand the underlying consequence relations, which on the other hand are manyfold. For instance, the gap between smoking history and graft failure, can be filled by referencing past cases (*e.g.* a significant number of lung transplants from donors with smoking history have ended up in a graft failure); by referencing guidelines and regulations, or by providing a domain explanation, *e.g.* smoking history is

[11] Of course, accounting for the perspectives of the different knowledge resources

a risk factor for structural damages in the lungs, which in turn, may affect their function, which, if implanted, may hinder the chances for a correct graft. Of course, the domain explanation can be more or less fine grained, introducing variations according to the experts' interpretations.

The choice to limit the deliberation to a cause-effect relation is only a practical one, that is to keep the deliberation focused and not divert it by including other kinds of reasonings (*e.g.* other argument schemes that might be more appropriate for an offline deliberation). The choice to define a rather shallow internal structure for the arguments is motivated to promote the participation of heterogeneous agents. Each PA (or any other knowledge resource) may have its own way to fill the gap between cause and effect (their own version of Γ). It is worth emphasising that there are only tow basic relations (and their negation) defined in *ProCLAIM*'s arguments structure. Thus, if R, A and S are respectively sets of facts, actions and effects, g a goal, $\Rightarrow$ a defeasible rule and $\neg$ the negation, then the two basic relations with their negations are:

1. $R \wedge A \Rightarrow S$ and $\neg(R \wedge A \Rightarrow S)$
2. $R \wedge S \Rightarrow g$ and $\neg(R \wedge S \Rightarrow g)$

The constructs introduced in the schemes' definition, beyond these basic relations among the four dimensions (*i.e.* facts,actions, effects and goals), are intended as a means to decompose Atikinson's *et al.* [34] scheme for practical reasoning, into the specialised schemes tailored for reasoning over safety critical actions. As we see later, in §7, these schemes then become the basis from which the scenario specific schemes are constructed. Namely, the introduced constructs are required for later providing support to developers in constructing the ASR. It is worth noting that other approaches, such as Araucaria [192] or Carneades [100] also provide users support in constructing argument schemes. However, their support is formal, in the sense that they help users build well formed schemes with their associated CQs following the frameworks' specification. Namely, these systems provide no particular support in identifying the appropriate reasoning patterns for a given application. This is exactly what *ProCLAIM* does, for the type of application it is intended for. Furthermore, the produced scenario specific scheme developed within *ProCLAIM* enables the automation of deliberation dialogues between agents (human or software) in a manner which is structured and orderly. Once the scenario specific schemes are produced we anticipate no difficulty in coding them in any of the above systems' specifications. Two points should be added here, in §11.2 we discuss how, from our experience in assisting developers not familiar with argumentation, we learned that the creation of scenario specific schemes is difficult, even when a number of indications are given and providing Atikinson's *et al.* [34] scheme for practical reasoning as a starting point. Secondly, scenario specific schemes can indeed be produced in an *ad-hoc* fashion, as we did in [17], however, the structured procedure we propose in the following chapter, based on the schemes developed in this chapter, helped us construct the medical ASR in more organised fashion and thus identify more reasoning lines not contemplated before. Furthermore, as we show in §10.1.3, the scenario-specific schemes developed with the proposed procedure are more effective in capturing the scenario reasoning patterns than the *ad-hoc* schemes we developed earlier. What required the exchange of four arguments in the *ad-hoc* version, requires only two in the current version (see §10.1.3).

We wish to further emphasise that the argumentation-based deliberative model that we propose is tailored to safety critical domains. As such there is an obligation to ensure users are guided to exhaustively pursue paths of reasoning. It is thus crucial that specialised domain specific guidance is provided. Walton's more abstract schemes are essentially domain independent patterns of reasoning, and thus needed to be specialised to provide more fine-grained domain specific guidance.

Argument schemes have been envisioned some years ago as an important tool for building practical applications in a wide set of fields [194], most approaches either address case modelling in the legal domain [102, 230, 42], or they are actually intended for relatively open scenarios [101, 34, 182]. Both cases thus, have strong requirements on expressivity and generality. Thus most works using argument schemes address an almost opposite problematic as that of *ProCLAIM* . Broadly speaking, while *ProCLAIM* aims to constrain deliberations to make them efficient, most approaches aim at generality.

Our hypothesis is that while the more abstract argument schemes (*e.g.* those of Walton [231] or of Atkinson *et al.* [34]) help structuring the argumentation, the specialised schemes help participants in the submission and exchange of arguments. The large scale demonstrator we describe in §10.1, the CBR system presented in §10.4 and the environmental scenario (see §11.2.1) illustrate the added value of scenario specific schemes. In this chapter we provided the basis to address the question: *how* to identify the right set of reasoning patterns appropriate for any given *ProCLAIM*-based application.

Chapter 7

Argumentation into Practice

The main focus of this chapter is to provide a clear view of how the MA can guide the participant (human or artificial) agents at each stage of the deliberation on what can be argued and how, and so, providing a setting for an efficient and effective deliberation among heterogeneous agents. Thus, in this chapter we put into practice the deliberation dialogue game introduced in §5 and the circuit of argument schemes and critical questions defined in §6. This circuit of schemes and CQs is tailored for deliberating over safety-critical actions and so, it provides a good basis for guiding the argumentation process identifying which are potentially relevant argumentation moves. However, when put into practice, the scheme instantiation at this stage is not yet a transparent process.

To illustrate this point let us suppose a deliberation over the safety of an organ transplant is in progress. Suppose then that the participants, DA and RA, are guided to reconsider the transplant safety via scheme $AS2$, thus they are questioned:

Is there a set of facts R such that the transplant will cause a side effect S which will realise the undesirable goal G?

Indeed, as we saw throughout §6, such schemes structure de argumentation process. However, as we can see in this case, it is not at all obvious with which values R, S and G can be instantiated. In consequence, the argument construction may involve too much overhead for the PAs and so disrupting the deliberation and providing no guaranty that PAs will succeed in constructing the desired arguments within the time constraints of the particular application. Consider this time, however, the following scheme specialised for the transplant scenario:

The donor's $C1$ will cause $C2$ to the recipient, which is a $severe\ infection$

To construct an argument using this scheme requires only instantiating $C1$ and $C2$ with the appropriate donor and recipient conditions respectively. For example, both taking the value *Hepatitis C*. In other words, instantiating this scheme involves little or no overhead at all. It is scenario-specific schemes like this one that the MA uses for guiding the PAs on what is to be argued about and how in the deliberation. That is, these are the schemes and CQs encoded in *ProCLAIM*'s Argument Scheme Repository (ASR).

In the following section we describe a step by step procedure for the ASR construction, by way of illustrating with the transplant scenario. In §7.2 we show how the MA, using *ProCLAIM*'s dialogue game and referencing the ASR, performs his guiding task both on artificial and human agents. In order to further focus the deliberation on the subject matter, *ProCLAIM* defines a validation process in which each PA submitted arguments is evaluated by the MA before it is added to the tree of arguments and broadcasted to all participants. In this way spurious arguments or arguments that are too weak to be accounted for in the deliberation are rejected. The validation process is described in §7.3.

7.1 Constructing an ASR

Once the circuit of schemes and CQs is defined, and tailored to encode stereotypical reasoning patterns for deliberating over safety-critical actions, we can further specialise this circuit to a particular application, *e.g.* the transplant or environmental scenario in order to construct the ASR.

To illustrate this process, let us consider the argument scheme $AS1$ in which, given the preconditions R_p, an action A_m is proposed. In the transplant scenario the proposed action is always the same: *transplant an organ*, and the preconditions are: to have an *available organ* for the *potential recipient*. Of course, in each instance the donor, the recipient and the organ are different. Thus, tailoring $AS1$ to the transplant scenario involves capturing this recurrent pattern while allowing for different donor, organ and recipient instantiation. This can be done by ungrounding the predicates `av_org(donor,organ)`, `p_recip(recipient,organ)` and `transp(recipient,organ)`. So denoting variables with upper-case letters we can define the tailored version of $AS1$ as:

$AS1_T$: `argue({},{},propose({av_org(D,O),p_recip(R,O)},`
`{transp(R,O)}))`

The MA references the ASR in order to provide the legal replies to an argument. In so doing, *ProCLAIM* facilitates a highly focused deliberation, paramount for its intended applications. This is not only because participants are directed in their argument submission to a degree where they only need to fill in some blanks (as in $AS1_T$), but also, in referencing the ASR the MA can easily identify the arguments that though logically valid, make little sense in the application scenario. Furthermore, the specialisation of the ASR plays an important role in the CBRc retrieval process, helping identify potentially similar cases (broadly speaking, cases in which the same reasoning patterns – specialised schemes – were used). We will discuss this in detail in the final Thesis version when we describe the CBRc.

Firstly, the ASR developers[1] must identify the type of information to be used in the deliberation. This information is encoded in the sets $\overline{\mathbf{R}}$, $\overline{\mathbf{A}}$, $\overline{\mathbf{S}}$ and $\overline{\mathbf{G}}$, which respectively denote the ungrounded versions of $\mathbf{R}$, $\mathbf{A}$, $\mathbf{S}$ and $\mathbf{G}$. That is, if for example `av_org(d,lung)` $\in \mathbf{R}$ then, `av_org(D,O)` $\in \overline{\mathbf{R}}$. Table 7.1 collects a sample of these sets.

[1] Most naturally, the construction of the ASR will be carried out mainly by computer science developers under the supervision of domain experts.

Set	Ungrounded Predicate	Description
$\overline{\mathbf{R}}$	`av_org(D,O)`	The organ `O` of the donor `D` is available for transplantation
	`p_recip(R,O)`	`R` is a potential recipient for an organ `O`
	`d_p(D,P)`	The donor `D` has the property `P`
	`org_p(D,O,P)`	The organ `O` of donor `D` has the property `P`
	`p_r_p(R,P)`	The potential recipient `R` has property `P`
	`blood_type(Pt,Bt)`	The patient `Pt` has blood type `Bt`
	`test(Pt,Tst,Res)`	Test `Tst` on patient `Pt` has result `Res`
	`clin_rec(Pt,P)`	Clinical records indicate that patient `Pt` has property `P`
$\overline{\mathbf{A}}$	`transp(R,O)`	Transplant organ `O` to the potential recipient `R`
	`treat(Pt,T)`	Treat patient `Pt` with treatment `T`
$\overline{\mathbf{S}}$	`r_p(R,P)`	The recipient `R` will have property `P`
	`reject(R,O)`	The recipient `R` will reject the organ `O`
	`death(R)`	The recipient `R` will die
$\overline{\mathbf{G}}$	`sev_inf(R)`	The recipient `R` will have a severe infection
	`grft_fail(R)`	The recipient `R` will have a graft failure
	`cancer(R)`	The recipient `R` will have cancer
	`toxic_shock(R)`	The recipient `R` will have a toxic shock
	`death(R)`	The recipient `R` will die

Table 7.1: A subset of the elements of $\overline{\mathbf{R}}$, $\overline{\mathbf{A}}$, $\overline{\mathbf{S}}$ and $\overline{\mathbf{G}}$

The next step is to choose the (type of) safety-critical action to be argued about (*e.g.* $\{\texttt{transp(R,O)}\}\subseteq \overline{\mathbf{A}}$) and identify a set of preconditions required for the action's performance (*e.g.* $\{\texttt{av\_org(D,O),p\_recip(R,O)}\}\subseteq \overline{\mathbf{R}}$). For each chosen set of actions $A_i \subseteq \overline{\mathbf{A}}$ and their set of preconditions $R_{p\_i} \subseteq \overline{\mathbf{R}}$ developers can define the specialised versions of $AS1$:

$AS1_i$: argue({},{},propose(R_{p_i}, A_i))

To each such $AS1_i$ there is associated the CQ $AS1_i\_CQ1$: *–Is there any contraindication for performing action A_i?–*, which can be embodied as an attack by a specialised version of $AS2$. Thus, given a specialisation of $AS1$ developers must produce specialised versions of $AS2$. Any specialised version of $AS2$ that replies to an argument instantiating $AS1_T$ is of the form:

$AS2_T$: argue({av_org(D,O),p_recip(R,O)},{transp(R,O)},
contra($R, S, \boldsymbol{g}$))[2]

Now, for each undesirable goal (see Table 7.1) that the action can bring about, (*e.g.* sev_inf, cancer, grft_fail, death,...) there is a partially specialised version of $AS2$, e.g.:

$AS2_{T\_gf}$: argue({av_org(D,O),p_recip(R,O)},{transp(R,O)},
contra(***R***, ***S***, grft_fail(R)))

Developers must now identify the type of side effects S ($S \subseteq \overline{\mathbf{S}}$) that realise each of these undesirable goals, and in turn, identify which are the type of contraindications R ($R \subseteq \overline{\mathbf{R}}$) that may lead the main action to cause these side effects. Thus, for example, a graft failure occurs when a recipient rejects the organ ({reject(R,O)}) which may be because of a donor property ({d_p(D,P)}, *e.g.* d_p(d,s_h)), due to a blood mismatch ({blood_type(D,BtypD), blood_type(R,BtypR)}) or because of a combination of the organ property and recipient property ({org_p(D,O,Po), p_r_p(R,Pr)} *e.g.* the lung is too big for the recipient's thoracic cavity), *etc*... Each of these combinations constitutes a specialised version of $AS2$:

$AS2_{T\_gf1}$: argue({av_org(D,O),p_recip(R,O)},{transp(R,O)},
contra({d_p(D,P)},{reject(R,O)},grft_fail(R)))

$AS2_{T\_gf2}$: argue({av_org(D,O),p_recip(R,O)},{transp(R,O)},
contra({blood_type(D,BtypD), blood_type(R,BtypR)},
{reject(R,O)},grft_fail(R)))

$AS2_{T\_gf3}$: argue({av_org(D,O),p_recip(R,O)},{transp(R,O)},
contra({org_p(D,O,Po),p_r_p(R,Pr)}, {reject(R,O)},
grft_fail(R)))

[2] Note that R is a set of facts and R is a variable bounded by p_recip(R,O) and transp(R,O).

Now, to each such specialised schemes there are in turn associated the CQs of the scheme $AS2$, which should direct developers in further constructing the ASR. For example, respectively embodying the Critical Questions $AS2_{T\_gf1}\_CQ1$, $AS2_{T\_gf1}\_CQ2$ and $AS2_{T\_gf1}\_CQ3$, $AS2_{T\_gf1}\_CQ4_1$ are the specialised schemes and challenge:

$AS3_{T\_gf1\_1}$: argue({av_org(D,O),p_recip(R,O),d_p(D,P)},{transp(R,O)}, no_side_effect({d_p(D,P2)},{reject(R,O)}))

$AS4_{T\_gf1\_1}$: argue({av_org(D,O),p_recip(R,O),d_p(D,P)},{transp(R,O)}, not_realised_goal({p_r_p(R,Pr)},{reject(R,O)},grft_fail(R)))

$AS5_{T\_gf1\_1}$: argue({av_org(D,O),p_recip(R,O),d_p(D,P)},{transp(R,O)}, preventive_action({treat(R,T)},{},{reject(R,O)})))

$AS2_{T\_gf1\_CQ4_1}$: challenge(evidence(d_p(D,P)))

The process continues in a similar way with each of these specialised schemes. Developers are thus directed in the construction of the ASR by the circuit of schemes and CQs described in §6. To continue with the example, we know from §6.2.3 that in reply to challenge $AS2_{T\_gf1}\_CQ4_1$, we can submit an argument instantiating scheme $AS2ev$:

argue({av_org(D,O),p_recip(R,O)},{transp(R,O)}, contra(replace_ev(d_p(D,P),$\boldsymbol{R}$),reject(R,O), grft_fail(R)));

Just as we did before, to specialise this scheme is to semi-instantiate the elements of the scheme that are not yet scenario-specific, in this case the set of facts R. As defined for the scheme $AS2ev$, the set of facts R has to be such that it provides evidence for d_p(D,P), that is $R {\mid\!\sim}_{ev}$ d_p(D,P). From the predicates in $\overline{\mathbf{R}}$ collected in table 7.1, R may be either {test(D,Tst,Res)} or {clin_rec(D,P)}. Therefore, the specialised versions of the argument scheme $AS2ev$ in reply to $AS2_{T\_gf1}\_CQ4_1$, are:

$AS2ev_{T\_gf1\_CQ4_1\_1}$: argue({av_org(D,O),p_recip(R,O)},{transp(R,O)}, contra(replace_ev(d_p(D,P),{test(D,Tst,Res)}),reject(R,O), grft_fail(R)));

$AS2ev_{T\_gf1\_CQ4_1\_2}$: argue({av_org(D,O),p_recip(R,O)},{transp(R,O)}, contra(replace_ev(d_p(D,P),{clin_rec(D,Inf)}),reject(R,O), grft_fail(R)));

The schemes we have developed thus far are intended for *artificial* agents. However, as mentioned in previous chapters, the ASR also encodes the schemes in Natural Language (NL) representation intended for the *human* agents. Before we present these NL schemes, it is important to note that when a specialised scheme is presented to a PA as a legal move it is almost completely instantiated (or even completely instantiated). Hence, in order to submit an argument the PA needs only to instantiate a few variables. To illustrate this, suppose PAs are guided to reply to the challenge:

```
   challenge(evidence(d_p(d,s_h)))
```

Thus, requesting for evidence for the fact that the donor has a smoking history. Thus supposing that `lung` is the offered organ and `r` the potential recipient then the two legal replies facilitated to the PAs are:

```
argue({av_org(d,lung),p_recip(r,lung)},{transp(r,lung)},
contra(replace_ev(d_p(d,s_h),{test(d,Tst,Res)}),reject(r,lung),
grft_fail(r)));
```

```
argue({av_org(d,lung),p_recip(r,lung)},{transp(r,lung)},
contra(replace_ev(d_p(d,s_h),{clin_rec(d,Inf)}),reject(r,lung),
grft_fail(r)));
```

The former legal reply requires instantiating the variables `Tst` and `Res` in order to construct a legal argument. That is, indicate a test `Tst` which result `Res` shows that the donor has a smoking history. The latter legal reply requires to introduce only the information `Inf` that appears on the donor's clinical records justifying that the donor has smoking history. Therefore, when constructing the schemes in a NL representation, much of the schemes' contextual information can be omitted in order to focus on the few variables that need to be instantiated. That is, `Tst` and `Res` in the former legal move and `Inf` in the latter. Thus, the NL representation of these two schemes may be:

$AS2ev_{T\_gf1\_CQ4_1\_1}$:*The donor has* `P` *since test* `Tst` *gave* `Res` *as a result*

$AS2ev_{T\_gf1\_CQ4_1\_2}$: *The donor has* `P` *since clinical records indicate that* `Inf`

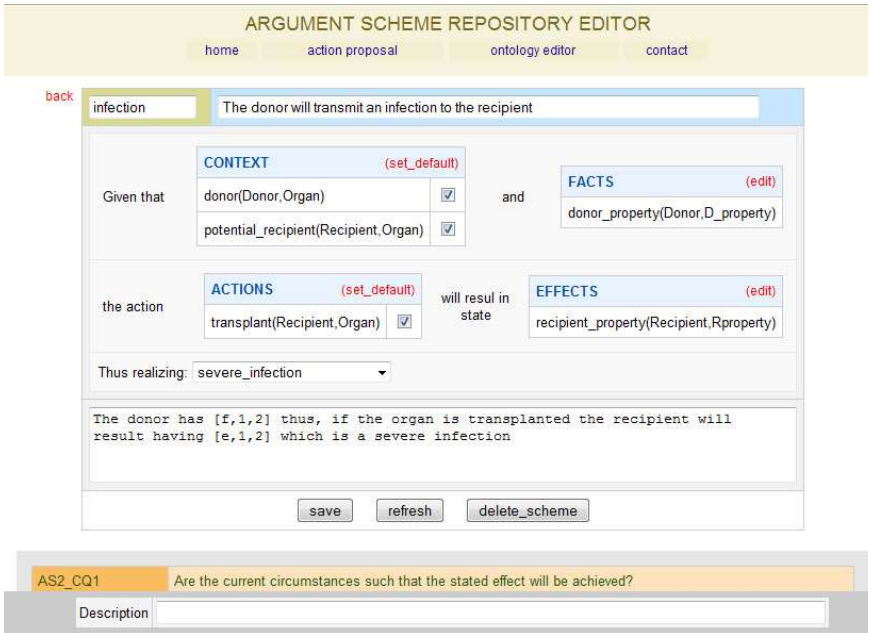
In a similar fashion we can define the above presented schemes $AS2_{T\_gf1}$ and $AS3_{T\_gf1\_1}$ as:

$AS2_{T\_gf1}$: *The donor's* **`P`** *causes a graft failure.*

$AS3_{T\_gf1\_1}$: *There wont be a graft failure because the donor has* **`P2`**

The main purpose of these NL schemes is to elicit from the experts the relevant factors throughout the deliberation. Hence it should be clear and transparent for the end users what is expected from them to feed in. It is important to recall that we take the PAs to be experts in the problem domain, namely, these human agents have a good understating of the problem at hand. Therefore, the role of argumentation as a educational artifact, very important in other works, as we have seen in §2.5, plays a minor part here in contrast to that of a tool intended for problem solving. In other words, efficiency is promoted sometimes in detriment of well formed NL arguments. An example of this is scheme $AS2_{T\_gf1}$. Strictly speaking, 'the donor's **`P`**' does not *cause* the graft failure. However, the scheme effectively conveys the idea that the donor has a condition **`P`** because of which, if the organ is transplanted, the organ may be rejected by the recipient.

The reason we can take the liberty of defining the NL schemes in such informal way is because these schemes are only informal at the surface, underlying these informal schemes a formal definition is given. It is the formal schemes that define the rules for the exchange of arguments. And so, the informal schemes are only used to interface with the human agents in order to effectively obtain the relevant factors for the decision making. Another important aspect of these informal schemes is that they are shorter than their full formal version. This is because we omit much of the contextual information, that though useful and sometimes necessary for artificial agents, it is redundant for human agents and at times may be disruptive.



infection — The donor will transmit an infection to the recipient

Given that

CONTEXT	(set_default)
donor(Donor,Organ)	☑
potential_recipient(Recipient,Organ)	☑

and

FACTS	(edit)
donor_property(Donor,D_property)	

the action

ACTIONS	(set_default)
transplant(Recipient,Organ)	☑

will resul in state

EFFECTS	(edit)
recipient_property(Recipient,Rproperty)	

Thus realizing: severe_infection

```
The donor has [f,1,2] thus, if the organ is transplanted the recipient will
result having [e,1,2] which is a severe infection
```

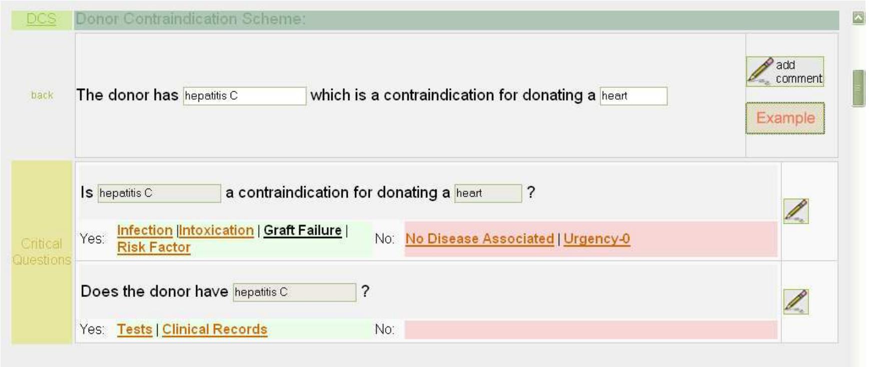
AS2_CQ1 — Are the current circumstances such that the stated effect will be achieved?

Description

Figure 7.1: ASR builder.

The construction of an ASR is a structured process in which, once the sets $\overline{\mathbf{R}}$, $\overline{\mathbf{A}}$, $\overline{\mathbf{S}}$ and $\overline{\mathbf{G}}$ are defined, developers are guided by the circuit of schemes and CQs defined in §6.2 step by step in the construction of the specialised, scenario specific, schemes. For each defined scenario specific scheme, developers must also define their NL counterpart. These NL schemes are not only important for the deliberation itself, but they are highly important for the validation process, where the domain experts check that the ASR does contain the appropriate schemes. In so doing, the end users should also help developers to refine the NL schemes so that they encode the right reasoning pattern in a clear and succinct way.

To facilitate these tasks we have developed two online tools: the first one intended to assist developers in the step by step ASR construction[3] (see figure 7.1) and another tool[4] which allows domain experts to navigate ASR's schemes and CQ (see figure 7.2) using the NL schemes. These tools are currently in a prototype phase of development and provides a useful proof of concept illustrating the potential value of our approach. We discuss these tools in more detail in §10.



Donor Contraindication Scheme:

The donor has hepatitis C which is a contraindication for donating a heart

Is hepatitis C a contraindication for donating a heart ?

Yes: Infection |Intoxication | Graft Failure | Risk Factor No: No Disease Associated | Urgency-0

Does the donor have hepatitis C ?

Yes: Tests | Clinical Records No:

Figure 7.2: ASR Browser.

In this subsection we have seen how the full space of argumentation can be codified in the ASR in a form useful for artificial and human agents. In the following subsection we show how this effort enables a highly focused deliberation process among heterogeneous agents.

7.2 MA's guiding task

In this section we show how the MA can perform his guiding task by following the rules of the dialogue game presented in §5 and referencing the ASR. Facilitating in this way the agents' participation.

A *ProCLAIM* deliberation begins with an argument proposing the main action, through instantiation of a specialised version of $AS1$ in the ASR. The basic idea is that an action (*e.g.* {transp(R,O)}) can only be proposed if the precondition (*e.g.* {av_org(D,O), p_recip(R,O)}) are met. In the transplant scenario, as soon as there is an available organ (*e.g.* `liver`) of a donor (*e.g.* `d`) for a potential recipient (*e.g.* `r`) $AS1_T$ can be instantiated automatically and the a deliberation triggered with the `open_dialogue` locution at the Open Stage:

```
inform(ma,all,conv_id,-1,0,open_dialogue( propose(
     {av_org(d,liver),p_recip(r,liver)},{transp(r,liver)})))
```

[3] http://www.lsi.upc.edu/~tolchinsky/newASR
[4] http://www.lsi.upc.edu/~tolchinsky/ASR

Agent	Submitted Facts	Description
DA	`av_org(d,liver)`	There is ana available liver from donor `d`
	`p_gender(d,male)`	The donor is a male
	`p_age(d,65)`	The donor is 65 year old
	`blood_type(d,A`$^+$`)`	The donor's blood type is A$^+$
	`bd_cause(d,ba)`	The donor's cause of brain death is *brain anoxia*
	`d_p(d,hbv)`	The donor has Hepatitis B
	`loctn(d,hosp1)`	The donor is located in hospital `hosp1`
RA	`p_recip(r,liver)`	`r` is a potential recipient for the available liver
	`p_gender(r,male)`	The recipient is a male
	`p_age(r,43)`	The recipient is 43 year old
	`blood_type(r,A`$^+$`)`	The recipient's blood type is A$^+$
	`p_r_prim_pat(r,cirrhosis)`	The recipient's primary pathology for the liver transplant is cirrhosis
	`loctn(r,hosp2)`	The recipient is located in hospital `hosp2`

Table 7.2: Information made available by the DA and the RA of the potential donor and the potential recipient

The DA that offers the organ and the RA responsible for the potential recipient may then enter the dialogue, for which they must first submit the following requests:

```
request(da_id,ma,conv_id,0,1,
   enter_dialogue(proposal,DA, d_basic_info))

request(ra_id,ma,conv_id,0,2,
   enter_dialogue(proposal,RA, r_basic_info))
```

With these `request` locutions the DA and RA request the MA to enter the deliberation over the stated proposal. The participant agents indicate the role they intent to play, that is DA and RA, and they also provide a list of facts they believed to be relevant for the deliberation: `d_basic_info` and `r_basic_info`, which respectively are the donor's and recipient's information. Table 7.2 shows the content of these sets for this example.

Supposing the MA accepts the two requests the MA must inform all participants of the incorporation of each PA into the deliberation indicating the role they will enact and the information they have introduced:

```
inform(ma,all,conv_id,1,3,
   enter_dialogue(proposal, da, d_basic_info, {ma}, ℂ_{F∧A}, 𝕋,
   legal_replies))

inform(ma,all,conv_id,2,4,
```

`enter_dialogue(proposal, ra, r_basic_info,{ma,da}` $\mathbb{C}_{F \wedge A}$`,` $\mathbb{T}$`,`
`legal_replies))`

The MA also provides PAs with updated information on the contextual information $\mathbb{C}_{F \wedge A}$ and state of the deliberation $\mathbb{T}$. Hence, if a PA enters the deliberation long after it has began, this PA is provided with the required information for an effective participation. At this stage, the set of facts $\mathbb{C}_F$ is updated to contain `d_basic_info` and `r_basic_info`. The set of actions $\mathbb{C}_A$ contains only action `transp(r,liver)` and $\mathbb{T}$ contains only the initial proposal.

Note that in these broadcasted messages the MA already informs the participants of the possible lines of attack on each argument in $\mathbb{T}$. In this example these are the replies to the initial proposal, say argument $A1$ with id *1*. To obtain these legal replies, the MA references the ASR and can retrieve the legal replies both in *code*-like representation, useful for artificial agents, or in NL, intended for human agents. Among these legal replies are the specialised schemes:

$AS2_{T\_inf1}$: `argue(`$\mathcal{C}$`,`$\mathcal{A}$`, contra({d_p(d,`**`Pd`**`)},{r_p(r,`**`Pr`**`)},sev_inf(r)))`

| *The donor's* **`Pd`** *will cause* **`Pr`** *to the recipient, which is a* $severe\ \ infection$

$AS2_{T\_gf1}$ `argue(`$\mathcal{C}$`,`$\mathcal{A}$`, contra({org_p(liver,`**`Pd`**`)},{reject(r,liver)},`
`grft_fail(r)))`

| *The donor's* **`Pd`** *will cause a* $graft\ failure$

$AS2_{T\_gf3}$: `argue(`$\mathcal{C}$`,`$\mathcal{A}$`, contra({org_p(liver,`**`Po`**`),p_r_p(r,`**`Pr`**`)},`
`{reject(r,liver)},grft_fail(r)))`

| *The organ property* **`Po`** *and the recipient's* **`Pr`** *will jointly cause a* $graft\ failure$

$AS2_{T\_cncr3}$: `argue(`$\mathcal{C}$`,`$\mathcal{A}$`, contra({o_p(liver,`**`Po`**`)},{r_p(r,cancer)},`
`cancer(r)))`

| *The organ property* **`Po`** *will cause* $cancer$ *to the recipient*

Where $\mathcal{C}$ = `{av_org(d,liver),p_recip(r,liver)}` and $\mathcal{A}$ =
`{transp(r,liver)}`.

Once the PA are given the legal replies they can move into the Argumentation Stage. Let us suppose, for instance, that the DA wishes to highlight a contraindications for donating the liver based on the donor's Hepatitis B, for which the DA will select the scheme $AS2_{T\_inf1}$ to construct argument:
$A2$: *The donor's* `Hepatitis B` *will cause a* `Hepatitis B` *to the recipient, which is a* $severe\ infection$

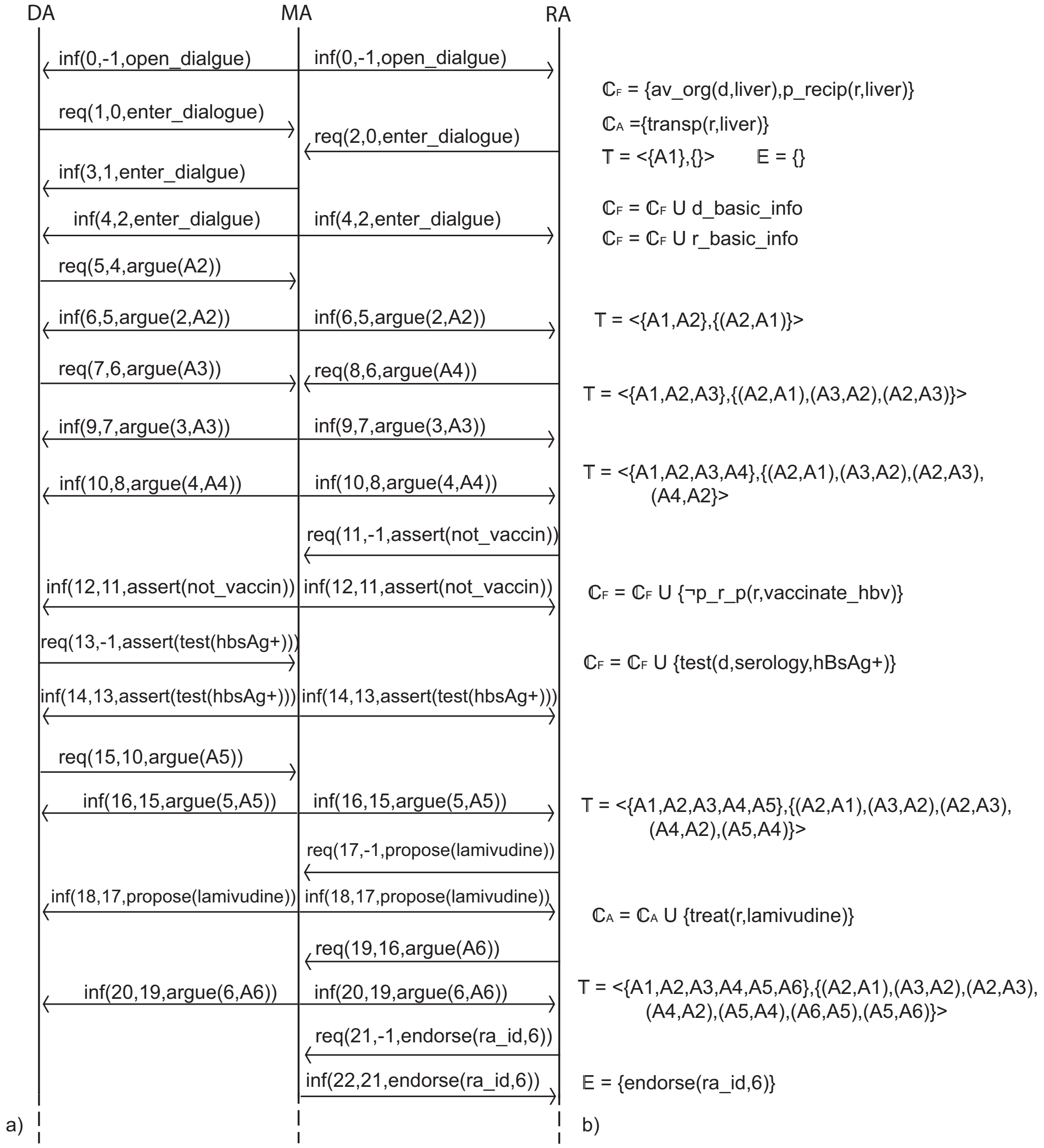

DA
MA
RA
inf(0,-1,open_dialgue)
inf(0,-1,open_dialgue)
ℂF = {av_org(d,liver),p_recip(r,liver)}
req(1,0,enter_dialogue)
ℂA ={transp(r,liver)}
req(2,0,enter_dialogue)
T = <{A1},{}>
E = {}
inf(3,1,enter_dialgue)
inf(4,2,enter_dialgue)
inf(4,2,enter_dialgue)
ℂF = ℂF U d_basic_info
ℂF = ℂF U r_basic_info
req(5,4,argue(A2))
inf(6,5,argue(2,A2))
inf(6,5,argue(2,A2))
T = <{A1,A2},{(A2,A1)}>
req(7,6,argue(A3))
req(8,6,argue(A4))
T = <{A1,A2,A3},{(A2,A1),(A3,A2),(A2,A3)}>
inf(9,7,argue(3,A3))
inf(9,7,argue(3,A3))
inf(10,8,argue(4,A4))
inf(10,8,argue(4,A4))
T = <{A1,A2,A3,A4},{(A2,A1),(A3,A2),(A2,A3),
(A4,A2}>
req(11,-1,assert(not_vaccin))
inf(12,11,assert(not_vaccin))
inf(12,11,assert(not_vaccin))
ℂF = ℂF U {¬p_r_p(r,vaccinate_hbv)}
req(13,-1,assert(test(hbsAg+)))
ℂF = ℂF U {test(d,serology,hBsAg+)}
inf(14,13,assert(test(hbsAg+)))
inf(14,13,assert(test(hbsAg+)))
req(15,10,argue(A5))
inf(16,15,argue(5,A5))
inf(16,15,argue(5,A5))
T = <{A1,A2,A3,A4,A5},{(A2,A1),(A3,A2),(A2,A3),
(A4,A2),(A5,A4)}>
req(17,-1,propose(lamivudine))
inf(18,17,propose(lamivudine))
inf(18,17,propose(lamivudine))
ℂA = ℂA U {treat(r,lamivudine)}
req(19,16,argue(A6))
inf(20,19,argue(6,A6))
inf(20,19,argue(6,A6))
T = <{A1,A2,A3,A4,A5,A6},{(A2,A1),(A3,A2),(A2,A3),
(A4,A2),(A5,A4),(A6,A5),(A5,A6)}>
req(21,-1,endorse(ra_id,6))
inf(22,21,endorse(ra_id,6))
E = {endorse(ra_id,6)}
a)
b)

Figure 7.3: Figure illustrating the agents exchanged messages and how the sets $\mathbb{C}_F$, $\mathbb{C}_A$, $\mathbb{T}$, and $\mathbb{E}$ are accordingly updated.

To submit this argument the DA sends it to the MA as `request` locutions:

```
request(da_id,ma,conv_id,4,5,argue(
   argue(𝒞,𝒜, contra({d_p(d,hbv)},{r_p(r,hbv)},sev_inf(r))),
   1))).
```

Assuming this argument is accepted by the MA it will be added to $\mathbb{T}$ with id *2* and broadcasted to all participants:

```
inform(ma,all,conv_id,5,6,argue(2,
   argue(𝒞,𝒜, contra({d_p(d,hbv)},{r_p(r,hbv)},sev_inf(r))),
   1,legal_replies))).
```

MA attaches the appropriate legal replies to argument $A2$, among which are:

$AS3_{T\_inf1\_1}$: argue($\mathcal{C}$,$\mathcal{A}$,no_side_effect({**d_p(d,Pd)**},{r_p(r,hbv)}))

> *The recipient will not be infected because the donor has* ***Pd***

$AS3_{T\_inf1\_2}$: argue($\mathcal{C}$,$\mathcal{A}$,no_side_effect({**p_r_p(r,Pr)**},{r_p(r,hbv)}))

> *The recipient will not be infected because he has* ***Pr***

$AS4_{T\_inf1\_1}$: argue($\mathcal{C}$,$\mathcal{A}$,not_realised_goal({p_r_p(r,Pr)},{r_p(r,hbv)}, sev_inf(r)))

> *The infection is not sever taking into account that the recipient has* ***Pr***

$AS5_{T\_inf1\_1}$: argue($\mathcal{C}$,$\mathcal{A}$,preventive_action({**treat(r,T)**},{}, {r_p(r,svi)})))

> *The infection can be prevented by treating the recipient with* ***T***

$AS2_{T\_inf1\_CQ4_1}$: challenge(evidence(d_p(d,hbv)))

> *Provide evidence for the donor's Hepatitis B*

Where $\mathcal{C}$ = {`av_org(d,liver)`,`p_recip(r,liver)`, `d_p(d,hbv)`} and $\mathcal{A}$ = {`transp(r,liver)`}.

Suppose now, the DA wishes to indicate that if the potential recipient has been vaccinated for HBV (`p_r_p(r,immunised_hbv)`), the transplant can safely be performed. And at the same time the RA wants to request the evidence for the donor's HBV. Hence, while DA will instantiate scheme $AS3_{T\_inf1\_2}$ to construct the argument $A3$, the RA will submit the CQ $AS2_{T\_inf1\_CQ4_1}$ as the challenge $A4$.

$A3$: argue($\mathcal{C}$,$\mathcal{A}$,no_side_effect({**p_r_p(r,vaccinated_hbv)**},{r_p(r,hbv)}))

> *The recipient will not be infected because he is* ***vaccinated_hbv***

$A4$: challenge(evidence(d_p(d,hbv)))

> *Provide evidence for the donor's Hepatitis B*

Since ¬p_r_p(r,vaccinated_hbv), p_r_p(r,vaccinated_hbv) $\notin \mathbb{C}_F$, argument $A3$, if accepted, would be hypothetical. Supposing it is accepted and that the recipient is not immunised against HBV, the RA should add ¬p_r_p(r,vaccinated_hbv) to $\mathbb{C}_F$ making argument $A3$ overruled:

```
request(ra_id, ma,conv_id,11,-1,
   assert(¬p_r_p(r,vaccinated_hbv))).
```

If challenge $A4$ is accepted, the MA will facilitate the following schemes for its reply:

$AS2ev_{T\_inf1\_CQ4_1\_1}$: argue($\mathcal{C}$,$\mathcal{A}$,
contra(replace_ev(d_p(d,hbv),{test(d,Tst,Res)}),r_p(r,hbv),
sev_inf(r))));

$AS2ev_{T\_inf1\_CQ4_1\_2}$: argue($\mathcal{C}$,$\mathcal{A}$,
contra(replace_ev(d_p(d,hbv),{clin_rec(d,hbv)}),r_p(r,hbv),
sev_inf(r))));

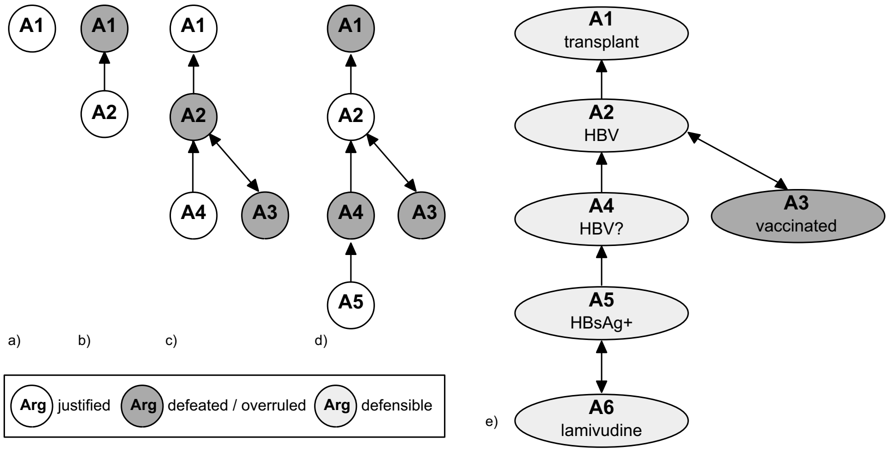


Figure 7.4: The tree of arguments $\mathbb{T}$ as the deliberation progresses.

The DA may then reply to challenge $A4$, firstly at the context level, by requesting to add test(d,serology,hBsAg+) to the set of facts $\mathbb{C}_F$, indicating that a blood test on the donor gave positive to HBsAg[5] and later at the argumentation level, by requesting to submit argument $A5$ instantiating scheme $AS2ev_{T\_inf1\_CQ4_1\_1}$:

```
request(da_id,ma,conv_id,13,-1,
   assert(test(d,serology,hBsAg+))).
```

```
request(da_id,ma,conv_id,10,15,argue(
   argue(𝒞,𝒜,contra(replace_ev(d_p(d,hbv),
   test(d,serology,hBsAg+)), r_p(r,hbv),sev_inf(r)),2))).
```

In a similar fashion, if argument $A5$ is accepted, the MA will direct the PAs to consider the schemes to instantiate, which are similar to those proposed in reply of argument $A2$, but this time $\mathcal{C}$ = {`av_org(d,liver),p_recip(r,liver),` `test(d,serology,hBsAg+)`}. Among which is again the scheme:

$AS5_{T\_inf1\_1b}$: `argue(𝒞,𝒜,preventive_action({`**`treat(r,T)`**`},{},`
`{r_p(r,hbv)})))`

| *The infection can be prevented by treating the recipient with* ***T***

Following these schemes, the RA may then propose treating the recipient with lamivudine which may prevent the recipients infection [68], prior to which the RA has to add `treat(r,lamivudine)` to $\mathbb{C}_A$(because of Assum_3a, see §6.2):

```
request(da_id, ma,conv_id,-1,17,propose(treat(r,lamivudine))).
```

```
request(da_id,ma,conv_id,16,19,argue(
   argue(𝒞,𝒜,preventive_action({treat(r,lamivudine)},{},
   {r_p(r,hbv)})), 2)).
```

Assuming these moves are accepted by the MA, `treat(r,lamivudine)` will be added to $\mathbb{C}_A$ and the argument, say $A6$, will be added to $\mathbb{T}$ with id *6*:

```
inform(ma,all,conv_id,17,18 propose(treat(r,lamivudine))).
```

```
inform(ma,all,conv_id,19,20,argue(6,
   argue(𝒞,𝒜, preventive_action({treat(r,lamivudine)},{},
   {r_p(r,hbv)})), 3,legal_replies)).
```

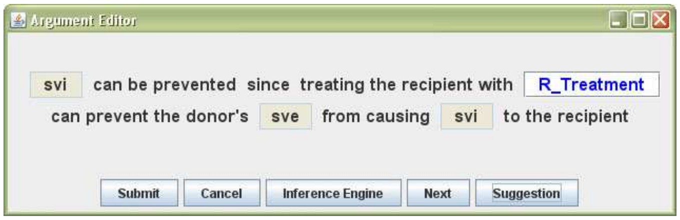
At this stage $\mathbb{T}$ contains the six arguments $A1$, $A2$, $A3$, $A4$, $A5$ and $A6$ organised as depicted in Figure 7.4e. The MA has provided the legal replies to each of these arguments to the PAs to further argue *for* or *against* the transplant safety. Concurrently, PAs may check whether there are new facts they may want to add to $\mathbb{C}_F$. For instance, the RA should check

[5] HBsAg is the surface antigen of the Hepatitis-B-Virus, if positive it indicates current Hepatitis B infection.

for any contraindications for administrating lamivudine to the recipient (such as allergy to lamivudine, a kidney disease or an incompatibility with other concurrent treatments), if that is the case, the RA may add this fact to $\mathbb{C}_F$ at the Context Stage and argue against $A6$ at the Argumentation Stage instantiating the appropriate legal reply facilitated by the MA. At this stage, the safety of the liver transplant depends on the efficacy of the lamivudine treatment, that is on deciding whether argument $A6$ is preferred to $A5$, as depicted Figure 7.4e. This will be addressed at the Resolution Stage, in §8. Note that in Figure 7.3, the RA has endorsed argument $A6$, thus, if RA is deemed as representing a prestigious transplant unit, this move will bias the final decision favouring argument $A6$ over $A5$. This we discus in §8 and in §11.2.1 we illustrate a complete *ProCLAIM* deliberation, this time in the environmental scenario.

The purpose of this section was to show how the deliberation can be put into practice led by *ProCLAIM*'s dialogue game and the ASR. While a critical point of this section was to show the PAs are guided at each stage of the deliberation on what can be argued about and how, thus making the deliberation highly focused on the subject matter, another important aspect worth highlighting is the provision of schemes tailored both for human and artificial agents. In §10.1 we present a prototype application that makes use of an ASR to facilitate the deliberation among a human agent and an artificial agent. In this prototype the human agent is assisted by an artificial agent that guides her in the argument submission, proposing alternative argument instantiation and validating them against the artificial agent's knowledge base (see fig.7.5), where the human agent can override any of artificial agent suggestions. While the human agent is presented with the arguments in NL, the assisting agent uses the schemes formatted in PROLOG code. At the other end, an artificial agent was guided by the MA with the provision of the ASR schemes also formatted in PROLOG code. This work has been presented in [11].

While the provision of specialised schemes defining the legal moves at the Argumentation Stage ensures, to some degree, that the agents' submitted arguments are relevant for the deliberation, there is still a risk of constructing spurious arguments if a legal reply is instantiated with unappropriate values. To prevent this from happening, the *ProCLAIM* model defines a validation process in which the MA checks the argument submission before adding them to $\mathbb{T}$ and broadcast them to all PAs. We present this validation process in this following section.



svi can be prevented since treating the recipient with R_Treatment can prevent the donor's sve from causing svi to the recipient

Figure 7.5: The argument editor of the application presented in [11]. This panel provides a human user with a legal reply ($AS5_{T\_inf1}$) to an argument. The *Inference Engine* button will validate the argument according to the knowledge base of an artificial agent that aids the user in the deliberation. The *Next* button provides the user with another legal reply (another scheme, *e.g.* $AS4_{T\_inf1}$) and button *Suggestion* proposes a scheme instantiation suggested by the artificial agent's knowledge base.

7.3 Argument Validation

As a way to focus the argumentation process on the problem at hand we have seen how the MA guides the PAs on what can be argued at each stage of the deliberation. With each broadcasted argumentation move the MA attaches the semi-instantiated schemes and challenges the PAs may use in order to reply the broadcasted move. However, even with this guidance the deliberation may still be disrupted with the submission of spurious arguments or arguments that are too week to be accounted for in a safety-critical deliberation. In order to prevent this from happening *ProCLAIM* defines a validation process in which each submitted argument must be evaluated before it is added to $\mathbb{T}$ and broadcasted to all participants. To perform this validation process the MA references the three knowledge resources: DCK, AEM and CBRc. Broadly speaking, for an argument to be accepted it must comply with guidelines and regulations, that is, it has to be validated by the DCK. However, if the argument is not validated by the DCK it may still be accepted if the PA that submitted the argument is trustworthy (validated by the AEM) and/or if similar arguments where successfully used in previous deliberations (validated by CBRc).

In this section we outline a formalisation that organises the validation process in order to facilitate its implementation, which is application dependent as it ultimately depends on how each of these knowledge resources is implemented. Let us first introduce the three following mappings:

Definition 7.1 *Let A be the set of instantiated schemes of the ASR of a particular ProCLAIM instantiation and P the set of all the participant agents, then:*

- *V_k is a mapping associated to the DCK such that V_k: $A \rightarrow (-1, 1)$ where for $arg \in A$ if $V_k(arg)$ is close to -1 the argument is spurious (*e.g.* the term instantiation is ontologically wrong) and if close to 1, it is certainly valid.*
- *V_t is a mapping associated to AEM such that V_t: $A \times P \rightarrow [0, 1]$ where for $arg \in A$ and $p \in P$ if $V_t(arg, p) = 0$ the player is untrustworthy w.r.t. to the argument's*

domain area and if $V_t(arg, p) = 1$ *this agent is empowered to submit any (sensible) argument w.r.t. to the argument's domain area.*

- V_e *is a mapping associated to the CBRc such that* V_e*:* $A \rightarrow [0, 1)$ *where given an argument* arg*, if there is no case stored in the CBRc where arguments* similar *to* arg *were used and were undefeated then* $V_e(arg) = 0$*. The more incidences of justified uses of arguments* similar *to* arg *the closer* $V_e(arg)$ *gets to 1. We will describe the notion of argument similarity in with more detail in the final thesis when describing the CBRc.*

Thus V_k, V_t and V_e provides an assessment on whether to validate an argument on the basis of three independent perspectives. Those respectively given by the DCK, the AEM and the CBRc. Though this may differ from scenario to scenario, it is reasonable to take the DCK's assessment as the dominant perspective. That is, an argument is accepted if validated by guidelines or else, it is only *exceptionably* accepted if the agent submitting the argument is trustworthy and/or the argument was successfully used in previous deliberations. In order to capture this let us introduce the following two threshold values t_v and t_r. Let t_v be such that if $V_k(arg) \geq t_v$ then argument arg is validated by DCK, irrespective of the assessments given by AEM and CBRc. Most naturally t_v would be set to take a value between zero and one ($0 < t_v < 1$). Now, tel t_r be such that if $V_k(arg) \leq t_r$ the argument is rejected irrespective of AEM and CBRc's valuation. That is, arg is deemed spurious. Most naturally, t_r would be set to take a value between minus one and zero ($-1 < t_r < 0$).

Arguments that fall between the two thresholds, that is $t_r < V_k(arg) < t_v$ are neither rejected nor validated by the DCK. These arguments may still be exceptionally accepted if validated by AEM and/or the CBRc. Nonetheless, given the safety-critical nature of the deliberation, each such exceptional decision requires first to be validated by a responsible (human) agent.

Let us then introduce the mapping $valid$ from $(t_r, t_v) \times [0, 1] \times [0, 1)$ into $(-1, 1)$ such that if arg is an argument between the two thresholds, $t_r < V_k(arg) < t_v$, and p a player then, if $valid(V_k(arg), V_t(arg, p), V_e(arg)) > 0$ the argument is deemed valid, otherwise it is rejected.

Therefore, to recapitulate, for an argument arg submitted by a player p the validation process goes as follows:

- if $V_k(arg) \geq t_v$ then the argument is accepted and added to $\mathbb{T}$; else,
- if $V_k(arg) \leq t_r$ then the argument is rejected as being spurious; else
- if $valid(V_k(arg), V_t(arg, p), V_e(arg)) > 0$ then arg is proposed for acceptance by the MA and if validated by the responsible (human) agent, arg is added to $\mathbb{T}$; else
- the argument arg is rejected as being too week.

Submitted arguments should be accepted if they make sense, not only from a logical point of view, but also from the context in which the deliberation takes place. Of course, the argument:

$Arg1$: *The donor's* **`black hair`** *will cause a* $graft\ failure$

Is logically sound, but would be spurious in a safety critical deliberation. Identify the boundaries for which argument should be accepted and which rejected is not a trivial task. Take for instance the argument:

$Arg2$: *There will be a* $graft\ failure$ *because the recipient is a* **`young afro-american`**

Although nowhere in the literature we have found arguments suggesting that being a young Afro-American is a contraindication for being a recipient, there are ongoing research studies (*e.g.* [65]) which highlight the low graft survival rates among Afro-American patients who are transplanted a kidney, particularly among pediatric recipients. Hence, while it is clear that argument $Arg1$ should be rejected right from the outset, *i.e.* $V_k(Arg1) \leq t_r$, argument $Arg2$, though would most probably be rejected, serves to illustrate that there are no clear boundaries, that is, some arguments will fall in between t_r and t_v. By defining a flexible account for validating the submitted arguments from multiple perspectives, *ProCLAIM* aims to prevent, on the one hand, the inclusion of arguments that can disrupt the deliberation process, while on the other hand, ensure that arguments that are potentially useful for the decision making are not rejected because they do not comply with standard guidelines.

7.4 Discussion

In this chapter we brought the first insight of how *ProCLAIM* can take argumentation into practice. We first illustrate how an ASR can be constructed for a given application, and how these defined application-specific reasoning patterns can guide PAs in an argumentation where the argument submission becomes a transparent process. A process where transplant professionals are requested for medical information (*e.g.* a donor condition or a prophylactic treatment) as opposed to abstract concepts as *facts*, *actions*, *effects* and *goals*, which as discussed in this chapter, they indeed help structure the argumentation, but as we discuss in §11, their instantiation is not immediately obvious and may be disruptive in a real-time deliberation.

An important aspect of the schemes defined by *ProCLAIM* is that they are defined both in NL and formally, which bridges the human-computer interaction. Both in assisting users in the argument instantiation, evaluation and submission and in the actual deliberation among human and artificial agents, we continue this discussion in §10.1.

To the best of our knowledge, all existing systems that propose a repository of schemes as a source for the argument construction, and actually provide such repository, are not designed for real-time dialogues. Examples of this are the Araucaria system [193] intended for analysing textual arguments, the Carneades system [101], intended to assist users in construct a wide variety of arguments in order to improve their ability to protect their interests in dialogues, or Rahwan *et al.*'s ArgDF [186] and *Avicenna* [182], which final aim is to facilitate the exchange of semi-structured arguments among different argument analysis and argumentation-support tools. Thus, while these systems aim to direct users to analyse the arguments' structure and relations, *ProCLAIM* is intended to focus users on the actual problem at hand.

Another approach is taken in the Magtalo system [195],in which a repository of fully instantiated arguments is used to help users express their position regarding a subject of public debate (in particular, whether *Identity cards are a bad idea*). The user can direct a dialogue among different artificial agents which allows them explore the system's knowledge base following the natural flow of a dialogue. In this way the user can view different arguments for and against each point made (where these arguments are already precoded in the system knowledge base). The user may then agree with the different exposed arguments shaping in this way her position. Furthermore, the user may select an argument from the argument store if it better matches her position and as a last resource, the user can type her own arguments in natural language (though with no support in therm of how the argument should be structured). The user interaction proposed by Magtalo is presented as non intrusive mode for eliciting knowledge from the players, in particular, their position with regard to a particular topic. This claim is based on what Walton and Krabbe call the *maieutic function* of dialogue [232]. That is, because users are immerse in a dialogue they do not feel being interrogated. It should be noted that Magtalo is not intended for decision making. In particular, it does not provide any argument evaluation mechanism. As just pointed out its, focuses in eliciting the user's position with regard to a given topic by reusing preexisting arguments.

We have concluded this chapter by anticipating uses of the knowledge resources DCK, CBRc and AEM, which play an important role in validating the PAs' submitted arguments. We propose a flexible account for deciding when an argument should be deemed as relevant for the deliberation and when it should be discarded. This validation process prevents the inclusion in $\mathbb{T}$ of arguments that are spurious or too weak to be considered in a safety-critical decision making, thus promoting a focused and efficient deliberation. At the same time, it is flexible enough so as to exceptionally accept arguments that while deemed too weak by guidelines, they have shown to be relevant in previous cases, or were submitted by a sufficiently trusted agent.

Once the tree of interacting argument is build, the MA has to propose a solution which accounts to each of the parties submitted arguments (whether factual, hypothetical or challenged and weakly replied) the PAs' endorsements, and to the assessments of the DCK, CBRc and AEM. This we discuss in the next chapter.

Chapter 8

ProCLAIM's Argument Evaluation

Once the PAs have submitted all their arguments, introduced all the facts they believe to be relevant and have endorsed the arguments they believe should be preferred, the deliberation enters into the Resolution Stage. At this stage, the MA has to propose a solution, on the basis of $\mathbb{T}$, $\mathbb{C}_F$, $\mathbb{C}_A$ and $\mathbb{E}$, on whether or not the main action can safely be performed. Broadly speaking, this involves applying Dung's theory in order to identify which are the rejected and justified arguments in $\mathbb{T}$. If the argument at the root of $\mathbb{T}$ is evaluated to be justified, the action is deemed safe, whereas if rejected, then the lines of arguments that lead to the rejection identify the contraindications that warrant deeming the action to be unsafe.

However, prior to computing the arguments' status MA has to: *1)* reference the DCK and the CBRc as these component may submit additional arguments relevant for the decision making; *2)* assign a preference relation between arguments that mutually attack each other, since symmetric attacks may prevent a definitive status evaluation of the main proposed argument; and finally *3)* appropriately deal with arguments that are hypothetical or that are not well defended from a challenge made on them.

Once these three tasks are performed, the MA returns a solution proposal for the deliberation. Where a solution is a justification to deem the action safe or unsafe. Nonetheless, when there is not enough knowledge about the particular problem at hand so as to provide a conclusive assessment over the action safety, a partial solution is presented to the decision makers (human) ultimately responsible for deciding the actions safety. This partial solution can be presented in $\mathbb{T}$'s graphical representation with additional features which all together facilitates the decision makers to see what makes the proposal uncertain and what are the risks involved in performing the safety critical action in the give circumstances. Once the decision makers resolve $\mathbb{T}$, it is returned to the PAs as *ProCLAIM*'s solution.

In the following section we discuss the MA's task of submitting additional arguments intended to ensure that all the factors relevant for the decision making are taken into account, not only from the view point of the PAs but also from domain guidelines and regulations and from similar past deliberations. In §8.2 we describe the argument preference assignment indented to disambiguate the symmetrical attacks that may preclude deciding which are the winning arguments. Again, the preference assignment process is derived from the PAs, DCK and CBRc's assessments. That is, it accounts for experts' opinion, domain consented knowledge and past collected evidence. In §8.3 we describe a labelling process by which to identify whether the final decision is dependent upon any unknown or uncertain fact. In §8.4 we discuss how the final solution is presented to the end users so that they can ultimately take the best decision with the available knowledge and information. In §8.4 we give a short discussion over the main issues introduced in this chapter.

8.1 MA's Additional Arguments Submission

Throughout the deliberation PAs submit the arguments they believe are relevant for the decision making. As experts in the domain their views are of course important. However, given the nature of the decision to be taken it is important to ensure, in as far as possible, that no potentially relevant factor is overlooked. To this end, the MA is assigned the task to submit any additional argument not accounted by the PAs but deemed relevant from the perspectives of the DCK and/or the CBRc. In other words, the MA is assigned the role of two additional PAs: an expert or specialist in domain consented knowledge, and another specialist in reusing evidence collected from past deliberations. The MA will perform this task as soon as the timeout is triggered, or when all the PAs submitted the `no_more_moves()` locution at the Resolution Stage. That is, when they inform they have no further moves to submit that may change either $\mathbb{C}_F$, $\mathbb{C}_A$, $\mathbb{T}$, or $\mathbb{E}$.

Let us first discuss how the MA references the DCK to submit additional arguments, which in its basic approach is a fairly simple process. Later we briefly discuss how the CBRc can propose additional arguments for submission. This latter, slightly more sophisticated task is addressed in §9.

Any additional argument submitted by the MA must instantiate a legal reply provided by the ASR. Thus, when playing the role of the specialist in domain consented knowledge, the MA has to check whether there are possible instantiations of legal replies that while validated by the DCK are not in $\mathbb{T}$. Thus, for each argument arg in $\mathbb{T}$, MA obtains its legal replies, that is, a list of specialised schemes sch_1,....sch_n, if the DCK is able to instantiate any such schemes into a valid argument, that is, an argument arg_{dck} such that $V_k(arg_{dck}) > t_v$, and such that arg_v is not in $\mathbb{T}$, then arg_{dck} should be added to $\mathbb{T}$.

For an additional argument arg_{dck} to be added to $\mathbb{T}$ it has to be applicable to the particular circumstances. That is, any introduced relevant fact must be in $\mathbb{C}_F$. Hence, additional arguments not only have to conform to the legal replies facilitated by the MA, but their instantiation is bounded by $\mathbb{C}_F$.

In its basic implementation the DCK can codify the guidelines and regulation of a given domain (relevant for the argumentation process) in the form of semi-instantiated schemes of the ASR, where the only ungrounded variables are the identifiers of the particular instances in the deliberation. For instance, the *donor* and the *recipient*. In fact, one could regard the ASR as an abstraction of the DCK. Thus, for example, if guidelines indicate that a condition `x` of the donor is a contraindication for a heart transplant because the transplant may cause `y` to the recipient which is a severe infection. This can be encoded in the DCK as the scheme:

AS_{dck}: `argue({av_org(D,heart),p_recip(R,heart)},` {transp(R,heart)}, contra({d_p(D,x)},{r_p(R,y)},sev_inf(R)))

It is thus, easy to see that if during a deliberation over a heart transplant, PAs have added d_p(d,x) to $\mathbb{C}_F$ but ignored it as a contraindication, by referencing the DCK the MA can instantiate scheme AS_{dck} to submit the additional argument and add it to $\mathbb{T}$. This of course opens the possibility to add hypothetical arguments thus recommending PAs to check if a given property holds or not. In a similar fashion the DCK can propose complementary courses of actions by instantiating specialised versions of scheme $AS5$, the only requirement is that the preconditions for the complementary actions are not negated in $\mathbb{C}_F$.

While a DCK submitted argument represents guidelines recommendations, arguments submitted by the CBRc derive from previous recorded deliberations. Hence, the process to obtain the additional arguments is slightly more sophisticated. As we will discus when presenting the CBRc, there are two aspects of the schemes that facilitate the CBRc task: *1)* the specificity of the schemes in the ASR (as described in §7) and *2)* that *relevant* facts and complementary courses of actions are introduced in a structured fashion, each singled out and introduced step by step. The schemes' specificity allows identifying potentially similar cases with little computational cost. The idea is that cases in which the same specialised schemes (reasoning patterns) were used, may be similar. Thus, by organising the case-base in terms of the ASR, a set of broadly similar cases can effectively be retrieved. The latter aspect of the schemes facilitates a more detailed comparison between cases on the basis of the similarity between the cases' introduced *relevant* facts and actions. We illustrate this with a simple example from the medical scenario.

Suppose the deliberation consisted only of the arguments $A1$, $A2$, $A3$ and $A4$ where a lung of a donor whose cause of death was *streptococcus viridans endocarditis* (d_p(d,sve)) is offered for transplantation, and the donor's sve is believed to be a contraindication ($A2$) because the recipient may be infected by this bacteria. Argument $A3$ indicates that the infection can be prevented by administrating *penicillin* to the recipient and argument $A4$ indicates that the recipient is allergic to such treatment. Arguments $A1$, $A2$, $A3$ and $A4$ respectively instantiate schemes $AS1_T$, $AS2_{T\_inf1}$, $AS5_{T\_inf1}$ and $AS2_{T\_inf1\_alrgy}$ encoding the following reasoning pattern:

*An organ **O** was intended for transplantation. The donor had some condition **P** which would bring about a severe infection in the recipient. Treatment **T** for the recipient was proposed to prevent this infection. However, the recipient is allergic to **T***

Thus by retrieving from the case-base all the deliberations which consisted of these four schemes we obtain cases that are already quite similar to our target case. So now, if we take from these past cases those where the organ O is a lung, the condition P is *similar* to sve (*e.g. streptococcus bovis endocarditis*) and where the treatment T is *similar* to penicillin, we obtain the desired set of cases from which to evaluate the target case on an evidential basis. Thus, while the argument schemes are used as a heuristics for a first, broad case retrieval, the similarity between cases is ultimately derived from a similarity between the facts and actions highlighted as relevant for the decision making. The similarity between facts and between actions can be derived from a distance measure between terms in an ontology. So, for instance, if the distance in a medical ontology between the terms `streptococcus bovis` and `streptococcus viridans` is below a given threshold, it can be derived that these two bacteria types are *similar*. And thus, if two arguments instantiate the same scheme of the ASR, and the used terms for their instantiation are *similar*, we can then say that these two arguments are similar (see §9.5).

Now, the similar cases retrieved by the CBRc may contain additional arguments that though applicable to the target application were not submitted by the PAs nor the DCK. Thus, for example, if in similar deliberations PAs have successfully argued that an alternative treatment to prevent the recipient's infection is `teicoplanin`, this argument will be proposed by the CBRc as an additional argument to be submitted by the MA (as discussed in §9).

It is worth mentioning that as soon as the PAs facilitate the contextual information $\mathbb{C}_F$, the MA can already construct a provisional $\mathbb{T}$ by referencing first the DCK so as to add the arguments that represent the guidelines recommendations and then the CBRc to submit the additional arguments deemed relevant by the CBRc. Thus, as soon as the PAs enter the deliberation they are presented a tentative solution. And it is only when they disagree with the proposed solution, or they believe they have additional potentially relevant information to submit, that the experts opinion comes to play an important role. *ProCLAIM* can account for this possibility simply by advancing the Resolution Stage.

Once all the arguments have been submitted, the following process in the arguments' evaluation involves assigning a preference relation between the mutually attacking arguments

8.2 Preference Assignment

While the tree of arguments map out all the relevant factors for the decision, arranging them in a convenient way, symmetrical attacks between arguments may preclude any definitive resolution. The MA's task at this stage is to resolve these symmetrical attacks by assigning a preference between mutually attacking arguments. Because the preference assignment is derived from thee independent knowledge resources (DCK, CBRc and AEM[1]), an important part in this process is to provide a clear account of the preference assignment, specially when the symmetrical attacks cannot be resolved in an *uncontroversial* way, that is, following the scenario's defined guidelines. In this section we will assume that $\mathbb{T}$ contains no hypothetical arguments and all challenges are successfully replied to, in the following section we address these cases.

ProCLAIM derive the preference assignment between arguments from its the three knowledge resources DCK, CBRc and AEM:

1. The **DCK** provides a preference assignment on the basis of the domain consented knowledge, that is guidelines, standard criteria, standard intervention plans, *etc*. For

[1] Argument Endorsement Manager, defined in §4.1

example, if penicillin is a recommended treatment for preventing the bacterial infection that a recipient may develop from a lung transplant of a donor who died because of streptococcus viridans endocarditis, the $A3$ will be deemed preferred to $A2$. If on the other hand is deemed inappropriate, $A2$ will be deemed as preferred to $A3$.

2. The **CBRc** provides a preference assignment based on evidence gathered from previous similar cases. Thus, if evidence exists in the CBRc memory for assigning $A3$ as preferred to $A2$, the CBRc will propose the transplant as safe based on the evidence that penicillin is an effective treatment for preventing the recipient's infection.

3. The **AEM** provides a preference assignment based on the trust in the experts endorsing one or another argument. Thus if the agent endorsing argument $A3$ is trustworthy (*e.g.* of a prestigious hospital) the AEM will propose deeming the transplant as safe based on expert opinion.

Each of these three preference assignment embodies different and independent perspective from which a proposal can be evaluated (that is, consented knowledge, evidence and expert opinion). Hence the preference assignments may all be in agreement (*e.g.* that $A2$ is preferred to $A3$) or not, for example, the DCK may prefer $A2$ to $A3$ deeming the treatment ineffective, CBRc may have no evidence for any (mutual attack between $A2$ and $A3$), while the AEM may indicate that the agent endorsing $A3$ is highly trustworthy ($A3$ preferred to $A2$). Furthermore, not only may the different knowledge resources yield conflicting preferences, but their preference assignments may vary in degrees of confidence.

To address these issues, we maintain the independence of the preference assignments, rather than aggregate them, since we believe that merging the three preference assignment would reduce the solution's quality. To this end, we define the preference assignment as a mapping:

$$pref : (\mathcal{A} \times \mathcal{A}) \mapsto ([-1,1] \times [-1,1] \times [-1,1])$$

Thus, pref($Arg1$,$Arg2$) = (a,b,c), where a is the preference assignment of the DCK, b of the CBRc and c of the AEM, and where positive values express a preference for the first argument over the later ($Arg1$ preferred to $Arg2$) and negative values the opposite. Zero means there is no preference at all. The bigger the absolute value of the number, the more the confidence in the preference assignment. Thus if pref($Arg1$,$Arg2$) = $(-1,-1,-1)$ then $Arg2$ is deemed preferred to $Arg1$ with full confidence. When the preference assignments are not all in agreement, say for instance pref($Arg1$,$Arg2$) = $(0.2,-0.6,-0.5)$, then decision makers must decide whether or not to override guidelines ($A1$ preferred to $A2$ with confidence 0.2), and trust the PA's assessment knowing that she is a reliable expert ($Arg2$ preferred to $Arg1$ with confidence 0.5) and her opinion is backed by evidence ($Arg2$ preferred to $Arg1$ with confidence 0.6). It is worth noting that symmetric attacks are only important to resolve when they preclude definitive evaluation of the status of the root argument proposing the main action. For example, in figure 8.1b. determining the direction of an asymmetric attack (based on a preference) between $A2$ and $A3$ is not relevant, as irrespective of such a determination, $A1$ is justified, since $A2$ is defeated by argument $A5$.

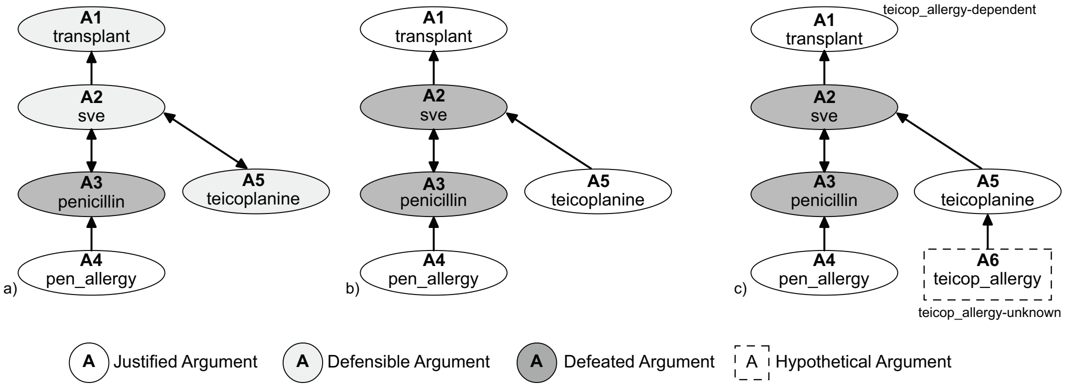


Figure 8.1: *a)* Because argument $A3$ is defeated by argument $A4$, the only symmetrical attack we must resolve is that between $A2$ and $A5$. *b)* Argument $A1$ will be justified irrespective of the preference relation between $A2$ and $A3$. *c*) The final solution depends on whether or not the recipient is allergic to teicoplanine.

Present decision makers with a graph filled with tuples of the form $(0.2, -0.6, -0.5)$, $(-0.4, 0.2, -0.1)$ or $(0.8, -0.1, 0.3)$ may not be the most clear way to present the proposed solution. To begin with, it is natural to take the DCK assessment as central to de decision making. Where, broadly speaking, argument $Arg1$ preferred to $Arg2$ if DCK deems it so. Other knowledge resources assessments can be regarded as supporting or contradicting the DCK assessment. In particular, it should be regarded as exceptional to override guidelines. This suggest a threefold partition of $pref$'s range:

- P = $(0, 1] \times [-1, 1] \times [-1, 1]$;
- Z = $\{0\} \times [-1, 1] \times [-1, 1]$;
- N = $[-1, 0) \times [-1, 1] \times [-1, 1]$;

Where, if $Arg1$ and $Arg2$ are two mutually attacking arguments, then if $pref(Arg1, Arg2)$ falls in P the DCK deems $Arg1$ preferred to $Arg2$, if it falls in Z no preference is given by the DCK, and finally, if it falls in N it is $Arg2$ that is preferred to $Arg1$. We can now propose another partition on P, so that if $pref(Arg1, Arg2)$ falls in one of following subsets then:

- P^+: DCK's assessment is supported by the other knowledge resources. *E.g.* $(0.8, 0.5, 0.6) \in P^+$
- P^0: DCK's assessment is not in significant conflict with the other knowledge resources. *E.g.* $(0.5, 0, -0.1) \in P^0$
- P^-: DCK's assessment is in conflict with the other knowledge resources. *E.g.* $(0.6, -0.4, -0.3) \in P^-$

- P^{--}: DCK's assessment is strongly contradicted by other knowledge resources. *E.g.* $(0.3, -0.7, -0.6) \in P^{-}$

So, if $pref(Arg1, Arg2) \in P^{+}$ then decision makers can deem $Arg1$ preferred to $Arg2$ with full confidence, whereas if it falls in P^{-} the symmetric attack is presented as unresolved and decision makers should take a position. However, if $pref(Arg1, Arg2) \in P^{--}$ ***ProCLAIM***'s recommendation is to override guidelines and deem $Arg2$ as preferred to $Arg1$. Defining the actual partition, that is, which elements of P corresponds to each of the four subsets is application dependent.

In a similar fashion we can define the partitions on Z, as follows, where if
$pref(Arg1, Arg2)$ falls in one of these sets then:

- Z^{0}: No knowledge resource provides a preference assignment, *E.g.* $(0, 0.1, -0.1) \in Z^{0}$
- $\pm Z$: Knowledge resources are in conflict, *E.g.* $(0, 0.5, -0.6) \in \pm Z$
- $+Z$: knowledge resources suggest $Arg1$ should be deemed preferred to $Arg2$, *E.g.* $(0, 0.1, 0.2) \in +Z$
- $+Z^{+}$: knowledge resources strongly suggest $Arg1$ should be deemed preferred to $Arg2$, *E.g.* $(0, 0.7, 0.5) \in +Z^{+}$
- $-Z$: knowledge resources suggest $Arg2$ should be deemed preferred to $Arg1$, *E.g.* $(0, -0.2, -0.1) \in -Z$
- $-Z^{+}$: knowledge resources strongly suggest $Arg2$ should be deemed preferred to $Arg1$, *E.g.* $(0, -0.6, -0.7) \in -Z^{+}$

And finally the partition on N (N^{+}, N^{0}, N^{-} and N^{--}) can be defined in a symmetrical way as that of P.

In this way, we can say that a symmetrical attack between the two arguments $Arg1$ and $Arg2$ is resolved uncontroversially if $pref(Arg1, Arg2) \in P^{+} \cup P^{0} \cup N^{0} \cup N^{+}$. If $pref(Arg1, Arg2) \in +Z^{+} \cup -Z^{+}$ decision makers are positively advised to resolve the symmetrical attack in one way or another, despite the DCK provided no assessment. If $pref(Arg1, Arg2) \in P^{--} \cup N^{--}$, decision makers would be strongly advice to resolve the symmetrical attack overriding guidelines. If $pref(Arg1, Arg2) \in +Z \cup -Z$ decision makers are merely suggested how to resolve the symmetrical attack, however it would be deemed as unresolved. And finally, if $pref(Arg1, Arg2) \in P^{-} \cup N^{-} \cup Z^{0} \cup \pm Z$ decision makers are given no advice at all.

Another step forward in assisting end users is to provide a short description of the actual preference assignment valuation. This is because, not only it is different for
$pref(Arg1, Arg2)$ to fall into P^{-}, N^{-}, Z^{0} or $\pm Z$, though in all four cases the symmetrical attack is deemed unresolved (at least provisionally), the situation is different if

$pref(Arg1, Arg2)$ is $(0.1, -0.5, 0.6)$, $(0.1, 0.6, -0.5)$ or even $(0.1, -0.9, 0.8)$, though all three values may belong to P^-.

The idea is to divide the intervals [-1,0] and [0,1] into subintervals with a qualitative description of the confidence with which the assessment is given. For example, using the following labels $\{$`top`, `strong`, `weak`, `none`$\}$ and thus, in this way the preference assignments $(0.1, 0.6, -0.5)$ and $(0.1, -0.6, 0.5)$ can respectively be presented as:

*A **weak** assessment from guidelines is **strongly** supported by evidence but **strongly** contradicted by experts' opinion*

*A **weak** assessment from guidelines is **strongly** support by experts' opinion but **strongly** contradicted by evidence*

At this stage the confidence the final decision makers have on each of the knowledges resource may balance their decision in one side or another. If they strongly rely on the CBRc assessment, evidence will be weighted as being more important. Though, it may also be the case that the experts' opinion will be deemed as more reliable. Of course, the decision makers themselves may have their own opinions regarding the preference between the mutually attacking arguments, which may clearly play a critical part in the final decision[2]. At this point it is worth emphasising that *ProCLAIM* is not intended for providing a the final decision to which stakeholders should stick to. Rather, it is intended to first gather and later present the available information and knowledge, relative to the problem at hand, in a clear fashion so as to facilitate the final decision making task. We discus how *ProCLAIM* presents a final decision in §8.4.

Before we continue, it is important to note that, if all symmetrical attacks are resolved, even if it is overriding guidelines, it is a trivial task to compute the acceptability status (defeated or justified) of all the arguments in $\mathbb{T}$, using Dung's theory. Furthermore, as we have seen in figure 8.1b, even if not all symmetric attacks are resolved a solution can be proposed, since what is important is to determine the acceptability status of the root argument of $\mathbb{T}$. Nonetheless, even if some unresolved symmetrical attacks still preclude computing the root argument's acceptability status, a partial resolution of $\mathbb{T}$ (labelling the defeated, justified and defensible arguments) is of great help, because, as illustrated in figure 8.1a, it enables to easily identify which are the symmetrical attacks that need to be resolved. We continue this discussion in §8.4 where we discuss how MA presents a solution proposal for a deliberation.

Let us now recall that $\mathbb{T}$ may contain hypothetical arguments and challenges that are either weakly replied or even unreplied. So even if the root argument is deemed a winning argument, the final decision may still depend on partial or incomplete information. We address this issue in the following section where we assume that a $\mathbb{T}$ is partially, if not completely, resolved, *i.e.* the arguments are labelled as defeated, justified or defensible.

[2] All this suggests some learning-by-reinforcement mechanism so as to improve the initial classification of $pref(Arg1, Arg2)$ into the sets P^+, P^0,..., N^- and N^{--}. This is beyond the scope of this study.

8.3 Accounting for Incomplete and Uncertain Information

As discussed both in §6.2.3 and §6.2.4, the purpose of *ProCLAIM*'s deliberation is not to decide whether or not uncertain or unknown facts are the case, but whether these are relevant for the actions' safety, and if so, what is the risk involved in these facts being or not the case. The risk involved in a fact being or not the case is highlighted by the arguments when indicating what undesirable side effects may or may not be expected, so let us discus now how *ProCLAIM* identify the uncertain or unknown facts that decision makers should be ware of.

Once the preference assignment process has taken place with a complete or partial resolutions of $\mathbb{T}$, where hypothetical arguments and weakly replied challenges are taken as regular elements of $\mathbb{T}$, the following labelling process takes place:

- Arguments whose updated local context of facts contain an *unknown* fact, f (*i.e.* $f, \neg f \notin \mathbb{C}_F$) are labelled as f-*unknown*;
- Arguments whose updated local context of facts contain an *uncertain* fact f, *i.e.* while $f \in \mathbb{C}_F$, f has been challenged but not well defended. These arguments are labelled f-*uncertain*;
- Arguments and challenges which acceptability status (defeated or justified, or defensible in presence of unresolved symmetrical attacks) depend on arguments labelled either as f-*unknown* or f-*uncertain* are labelled as f-*dependent*. That is, if their acceptability status change depending on whether f holds or not.

Let us continue with above example, where now, as depicted in 8.1c the hypothetical argument $A6$ has been submitted requiring to check whether or not the recipient is allergic to teicoplanine (`p_r_p(r,teicop_allergy)`) before deciding on the transplant safety. Hence, argument $A6$ is `p_r_p(r,teicop_allergy)`-*unknown* and both arguments $A1$ and $A5$ are `p_r_p(r,teicop_allergy)`-*dependent*. This is because, if `p_r_p(r,teicop_allergy)` is taken to be the case, both $A5$ and $A1$ become defeated, whereas if it is taken to be false, $A6$ would be overruled, and both $A5$ and $A1$ would be justified. Namely, both $A1$ and $A51$'s acceptability status depends on `p_r_p(r,teicop_allergy)` being or not the case.

8.4 Proposing a Solution

Once the additional arguments have been submitted, the preference between arguments assigned and $\mathbb{T}$ has been labelled with the f-dependencies the MA has to present a solution proposal. Of course, if all symmetrical attacks are satisfactorily resolved and the root argument is not dependent on any unknown or uncertain fact, there is no question about whether or not to perform the safety critical action and thus, the resolved tree of arguments $\mathbb{T}$ can be used as a solution to be sent to the PAs. However, if that is not the case, the decision makers empowered to validate exceptional choices should intervene to make a final decision, for which they should be presented with the available informational and knowledge about the problem at hand in clear fashion so as to facilitate their decision making.

Firstly it is worth emphasising again the explanatory power of argumentation, especially when presented in its graphical mode as we showed throughout this paper (see §2.5). Presenting the trees of argument, with the labelled nodes and their attack relations enables end users (experts in the domain) to easily grasp the full content of the deliberation. By colouring the nodes to indicate their acceptability status (justified, defeated and undecided) if the root argument is coloured as defeated (resp. justified) it is easy to understand why the action was deemed unsafe (resp. safe), if coloured undecided, it is easy to identify the symmetrical attacks that precludes selecting a final decision. What remains to be presented to the end users is a clear account of the unresolved symmetrical attacks valuation in order to facilitate their decision making and the dependencies the final decision may have on incomplete or uncertain information.

ProCLAIM extends the tree of arguments' graphical representation with additional information. The initial classification of the symmetrical attacks in one of the sets defined in §8.2 enables a first classification in five categories:

1. The symmetrical attack is uncontroversially resolved (*i.e.* P^+, P^0, N^+ or N^0)

2. A resolution is well supported, though no assessment is given by guidelines (*i.e.* $+Z^+$ and $-Z^+$)

3. A resolution, though conflicting with guidelines, is well supported (*i.e.* P^{--} and N^{--})

4. There is a tame recommendation on how to resolve the symmetrical attack (*i.e.* $+Z$ and $-Z$).

5. No recommendation can be given (*i.e.* P^-, Z^0, $\pm Z$ or N^-).

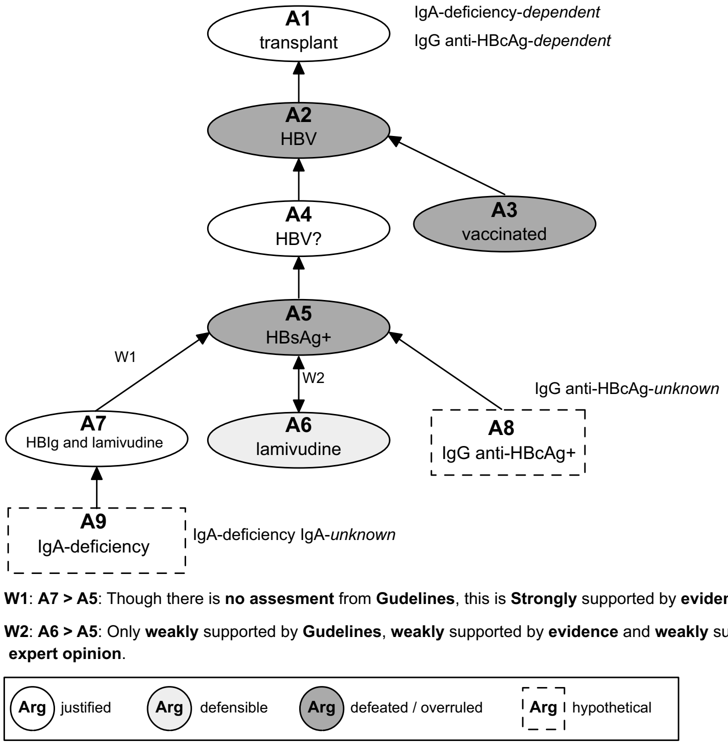
If the preference assignment falls in the first, second or third category, the symmetrical attack can be deemed as resolved, though it may require validation from final decision makers when it falls in the latter two categories[3]. For the fourth and fifth categories, decision makers can benefit from viewing the descriptive definition of preference assignment and thus see the three independent assessments of *ProCLAIM*'s knowledge resources. It is worth noting, again, that if no definitive status of acceptability can be assigned to the root argument, not all symmetrical attacks need to be resolved, thus decision makers should only focus on those that preclude computing the root argument's status of acceptability, which are easy to identify thanks to the nodes colouring. At the same time, decision makers would be advised to resolve those symmetrical attacks that fall into the fourth category before addressing those in the fifth category.

In addition to resolving symmetrical attacks, which correspond to deciding whether or not an undesirable side effect will be caused in certain circumstances, decision makers may

[3] Any preference assignment that falls in the second or third category should trigger a revision on DCK. This is beyond the scope of this research.

need to decide what to do when part of these circumstances is uncertain or unknown. As we have discussed earlier (in §6.2.3 and §6.2.4), while it is beyond *ProCLAIM*'s scope to valuate the certainty of the facts put forward in the argumentation process, decision makers should have a clear account of the presence of the uncertain and unknown facts and their impact in the action's safety. To this end, as we discussed in §8.3, *ProCLAIM* defines a labelling process to identify the arguments which definitive acceptability status depend on unknown or uncertain facts. Therefore, $\mathbb{T}$'s graphical representation can be extended so that it highlights the nodes that are f-dependent, for some $f \in \mathbf{R}$ (see figure 8.2). Presented in this way, decision makers can easily identify which f*-dependencies* they should address, as the only relevant f*-dependencies* are those of the root argument.

To address an f-dependency is to make a choice to take the uncertain or unknown fact as either the case, or as false. Where, as stated in §8.3 the risk involved in such decision (the undesirable side effect that may be caused) is highlighted by the argument introducing the f*-dependency*. That is, an argument either labelled as f*-unknown* and so hypothetical, or labelled as f*-uncertain*, namely an argument not well defended from a challenge made on the introduced fact f. Needless to say that an argument may be dependent on more than one fact.



W1: **A7 > A5**: Though there is **no assesment** from **Gudelines**, this is **Strongly** supported by **evidence**.

W2: **A6 > A5**: Only **weakly** supported by **Gudelines**, **weakly** supported by **evidence** and **weakly** support from **expert opinion**.

Figure 8.2: A *ProCLAIM* solution proposal

Figure 8.2 illustrates a possible presentation of partial solution given by *ProCLAIM*of the example case given in this chapter, while Figure 8.2 illustrate a possible solution to the example case developed in the previous chapter, where the MA have added three additional arguments, $A7$ as an alternative treatment believed to be more effective than lamivudine to prevent HBV on the recipient, which involves combining the lamivudine treatment with hepatitis B immunoglobulin (HBIg) [124]. However, to safely perform such treatment the RA should first check that the recipient does not have Immunoglobulin A deficiency (IgA-deficiency) which may lead to an adverse reaction to the immunoglobulin treatment, this is embodied by the hypothetical argument $A9$. The third, also hypothetical argument is $A8$ that indicates that if the potential recipient happens to be IgG anti-HBcAg-positive[4], then the transplant can safely be performed [143].

For a partially resolved $\mathbb{T}$, finale decision makers must disambiguate, in one way or another, the symmetrical attacks and the possible f-dependencies. Such solution is then returned to the PAs using the `solution` locution at the Resolution Stage. This given solution is the resolved $\mathbb{T}$, where the root argument is either justified or defeated. Now, if PAs accept the proposed solution they should submit the `accept` locution, otherwise, they can simply submit an additional move which may change either $\mathbb{T}$, $\mathbb{C}_F$ or $\mathbb{E}$ and that may subsequently change MA's proposed solution. In the hepatitis B example, depicted in Figure 8.2, if the RA adds to $\mathbb{C}_F$ either $\neg$`p_r_p(r,igA_def)`, overruling argument $A9$, or `p_r_p(r,igG_Anti_HBc+)` making argument $A8$ factual, the transplant would be deemed safe. Otherwise, decision makers would have to decide whether or not lamivudine alone is effective in preventing the infection to the recipient. Once a solution is proposed which all PAs accept the deliberation concludes.

8.5 Discussion

In this chapter we have described the nature of *ProCLAIM* proposed solutions, which ultimately involves the integration of the knowledge derived form the diverse and heterogenous knowledge resources that take part in the decision making. In the two previous chapters we have focused on the submission and exchange of arguments for or against the safety of a proposed action. In the previous chapter we have described how the MA can elicit knowledge from the PAs about the problem at hand by providing them scenario-specific argument schemes which can be instantiated with little overhead, both for human and artificial agents. In this chapter we continued this story by discussing how the MA itself can submit additional arguments by referencing the DCK and the CBRc. We noted that while the DCK argument submission may be implemented in a somewhat similar fashion as of an artificial PA[5], the CBRc implementation is rather more sophisticated and will be described in §9.

We then discuss the preference assignment derived from the DCK and CBRc. Dealing with preferences is a known difficult problem in decision making [79, 59] and while argumentation has developed frameworks for arguing about preferences [159], the problem clearly remains there: where do preferences come from? In that respect we believe that regulated environment, as the ones addressed by *ProCLAIM* are more likely to have

[4] The potential recipient has the IgG antibodies for HBV and may be immune to this virus. This may be caused from a previous HBV infection that has spontaneously been cured.

[5] We give some insight on the PAs implementation in §10

intrinsic motivations to organise and prioritise their guidelines and regulations from were these preferences may stem. In the transplant domain, for example, guidelines are commonly classified in terms of the evidence level that support them. Standards are elaborated to express these levels, for example the Oxford Centre for Evidence-based Medicine[6] proposes nine levels of evidential support (1a, 1b, 1c, 2a, 2b, 2c, 3a, 3b, 4 and 5)[7] and four recommendation grades (A,B,C and D), which relate to the evidence level supporting the recommendation. A more succinct classification (though following very similar criteria) can be find the transplantation guidelines published by the European Dialysis and Transplant Association[8] or the International Society for Heart and Lung Transplantation[9]:

Evidence Level A : Guidelines are supported by at least by one large published randomised control trial or more.

Evidence Level B : Guidelines are supported by large open trials or smaller trials with consensus results.

Evidence Level C : Guidelines are derived from small or controversial studies, or represent the opinion of a group of experts. [10]

So, to each criterion or recommendation in these guidelines an evidence level or grade of recommendation is associated, *e.g.*:

- Kidneys from HBV-infected living or cardiac donors may be offered to already HBsAg-positive recipients or HBV well protected recipients (active or passive immunisation) with their consent and when permitted by the national law.

  (*Evidence Level C*) [11]

- Intravenous ganciclovir may be administered to intermediate and high-risk patients, whereas patients at low-risk for Cytomegalovirus infection may only receive anti-herpes simplex virus prophylaxis with acyclovir.

  (*Evidence Level A*) [75]

- In kidney transplantation Anti-CD20 (rituximab) may be efficacious to treat acute humoral rejection. However, firm evidence on efficacy and side-effects are lacking.

  (*Grade of Recommendation B*) [12]

[6] http://www.cebm.net/

[7] http://www.cebm.net/index.aspx?o=1025e

[8] http://www.uktransplant.org.uk

[9] http://www.ishlt.org/

[10] http://ndt.oupjournals.org/cgi/reprint/15/suppl_7/2.pdf

[11] http://ndt.oupjournals.org/cgi/reprint/15/suppl_7/3.pdf

[12] From the European Association of Urology:
http://www.uroweb.org/gls/pdf/Renal%20Transplantation%202010.pdf

While this does not imply that obtaining the preference assignment for the DCK will be a trivial task, it certainly is a very good starting point. Note that artificial PAs can derive their argument preferences (which are manifested when endorsing an argument) following the same idea. In the transplant case, the artificial DA and RA can derive their arguments and preferences from the guidelines specifics of their hospitals. The human PA can then modify any stated preference by endorsing the arguments they believe should be stronger at that time (*e.g.* by mouse clicking on the appropriate node in $\mathbb{T}$). That is, PAs do not have to assign a strength but only indicate which of the competing arguments they endorse/prefer. The AEM will then convert these endorsements into a preference assignment, on the basis of some trust measure which accounts to the confidence in PA's judgement. Note that this process bypasses two problems of the preference assignment. First, users do not have to choose among a scale of options (either qualitative or numerical) and we do not face a normalisation problem inherent to the subjectivity in the PAs perceived certainty on their assessment.

We should note that each of the perspectives by which the arguments are evaluated: *guidelines*, *evidence* and *expert opinion* can in turn be formalised as Walton's proposed schemes [231]: argument from established rules, from evidence and argument from expert witness testimony, respectively. However, we believe that widening the *ProCLAIM* deliberations to explicitly include a discussion over the guidelines, the evidence or the experts themselves would be impractical within a real-time deliberation over the safety of a critical action. That is, questioning whether *–the applied guideline is the right one, or should some other guideline be the right one? Could there be more than one guidelines involved, with some doubt on which is the more appropriate one?–* is impractical when deciding in real-time the safety of an organ transplant or the safety of a toxic spill. Similarly, including into the deliberation questions over the PAs' *bias*, *honesty* or *conscientiousness* may not be desirable. To a great degree these questions are implicitly accounted for by *ProCLAIM* in allowing for a deliberation and assigning different weights to each knowledge resource assessment based on its reputation. It might be interesting for future work to map out how exactly the CQs of these schemes are actually addressed, because it is clear that these CQs are relevant.

Once we described how the MA can submit additional arguments and assign the preference relations between mutually attacking arguments, by referencing the DCK, CBRc and the AEM, we discussed how to organise all the available knowledge about the problem at hand in order to present it in a comprehensible way to the final decision makers. The objective is to limit the presentation to only the relevant aspects that need to be addressed. When the proposed action falls in a situation where guidelines resolve it with full confidence and the acceptability status of the root argument is not dependent on any uncertain or unknown facts, $\mathbb{T}$ is presented as a justification to why the main action is safe or unsafe. Otherwise, each aspect that prevents a definitive resolution is highlighted. Whether it is an unresolved symmetrical attack or an uncertain or unknown fact or facts. Also, symmetrical attacks that are not resolved following the DCK assessment are highlighted to indicate proposals that deviate from the guidelines.

We would like to emphasise that *ProCLAIM* results are always presented as proposed solutions. The purpose of *ProCLAIM* is to lay out the available information and knowledge at hand so as to support the decision makers in their task. The final decision is of course that of the human user. Thus, the role of *ProCLAIM* is to support informed decisions. Hence, we assume there is a gap, which we do not intend to fill, between the given support and the final decision-making.

It is worth noting that *ProCLAIM* allows for the inclusion of any other Knowledge Resource (KR) which assessment on the safety of the critical action at stake is deemed relevant. The only requirement for any KR to partake in the deliberation is to be able to judge whether or not, given a state of affairs, the intended course of action will or will not cause an undesirable effects. The strength of the KR's endorsed arguments would then be based on the associated trust on the KR's judgement. Thus, in particular, *ProCLAIM* allows including KRs that are based on any form of reasoning (*e.g.* probabilistic methods) in so far the above mentioned requirement is fulfilled. In that sense *ProCLAIM* is best viewed as a meeting point where the different views of the heterogeneous KRs are integrated in a structured and orderly fashion by means of a deliberation dialogue. In proposing a solution the model accounts for the independence assessment derived from the different KRs. The quality of the proposed solution will then depend on the quality of the KRs involved in the decision making and the associated trust in their assessment. We should note that in some sense this approach is shared with Carneades' [101] view of the potential of argument schemes along with argumentation framework to provide an open architecture for integrating multiple forms of reasoning. Carneades identifies each form of reasoning with a differed kind of scheme, thus, an argument from practical reasoning is a different form of reasoning form that of argument from evidence. From this perspective *ProCLAIM* defines only one mode of reasoning which is that from practical reasoning. The multiple instances of this particular scheme, those which conform the ASR, enable *ProCLAIM* elicit form the different KRs the relevant knowledge always in the form of a cause-effect relation. Other forms of reasoning (*i.e.* argument from established rules, from evidence and argument from expert witness testimony) are accounted for by *ProCLAIM*, but are used to evaluate the submitted arguments (all instantiating the scheme for practical reasoning).

While we obtained positive feedback from transplant professionals and the environmental engineers regarding *ProCLAIM*'s solution proposals as a comprehensible presentation to end users two important tasks are required for future work: improve the quality of the KRs involved in the argument evaluation (particularly of the DCK) so that solution proposals are more realistic and deliver a more evolved GUI that will facilitate the readability and interaction with the proposed solution.

Chapter 9

ProCLAIM's Case-Based Reasoning Component

In this chapter we present *ProCLAIM*'s Case-Based Reasoning component, which allows reusing past stored deliberation in order to provide an assessment over the safety of an action on an evidential basis. In the following section we introduce the basic background of Case-Based Reasoning and Argumentation and discuss related works. In §9.2 we sketch out the general principles on which the CBRc operates and illustrate these ideas by means of a simple example. Then, in the following sections we provide a more detailed description of the CBRc. We begin by defining, in §9.3 how the CBRc represents each *case*. In §9.4 we describe the CBRc memory organisation. That is, we show how the stored cases are organised in the case-base so as to facilitate their retrieval when solving a new case, and the storage of a new resolved cases. Once we define how case are represented and how they are organised, we present in §9.5 the CBRc's reasoning cycle. That is, how the CBRc proposes solutions to a target deliberation on an evidential basis, as well as, how resolved deliberations (cases) are then retained by the CBRc to help resolve future similar cases. In §9.6 we describe the CBRc validation process and finally, in §9.7 we conclude with a discussion regarding the CBRc.

9.1 Background

Case-Based Reasoning [133], broadly construed, is the process of solving new problems based on the solutions of similar past problems. For a computer system to carry out this reasoning, CBR has been formalised as a four-step process, commonly known as the four Rs (Retrieve, Reuse, Revise and Retain) [19].

Retrieve: Given a target problem, retrieve cases from memory that are relevant to solving it. A case consists of a problem, its solution, and, typically, annotations about how the solution was derived.

Reuse: Map the solution from the retrieved case to the target problem. This may involve adapting the solution as needed to fit the new situation.

Revise: Having mapped the retrieved solution to the target situation, test the new solution in the real world (or a simulation) and, if necessary, revise.

Retain: After the solution has been successfully adapted to the target problem, store the resulting experience as a new case in memory.

CBR traces its roots to the work of Schank in the early 1980s. Schank's model of dynamic memory [200] was the basis for the earliest CBR systems such as Kolodner's *CYRUS* [134] or Lebowitz's *IPP* [140]. During the 1990s CBR has consolidated as a mainstream research area in AI. CBR technology has produced a number of successful deployed systems, the earliest being *CLAVIER* [147], a system for laying out composite parts to be baked in an industrial convection oven.

Concurrently other closely allied fields to CBR emerged. Such is the case of memory-based reasoning or, more relevant here, the exploration of legal reasoning through formal models of reasoning with cases, particularly in the context of the *Common law system*, a legal system based on unwritten laws developed through judicial decisions that create binding precedent.

The inherent argumentative nature of the legal domain and the Common law system particularity, has provided the scenario for developing models and systems for reasoning and arguing with precedents, *i.e.* past cases.

Broadly, the legal reasoning which these models and systems aim to capture involve a number of stages. First, identifying the relevant features for resolving a new case. This features are called *factors*. Defendant and plaintiff most commonly identify different factors in a case, more in line with their positions. Secondly, parties may argue over the factors that should be used to describe the case. Once this is done the case has a number of reasons to resolve it in one way and a number of reasons to resolve it in the other way. Thirdly, precedent cases are introduced. Precedents represent past situations where these competing factors were weighed against one another and a decision was taken. At this stage, if a precedent is found by one of the parties (defendant or plaintiff) such that it has the same factors as the current case and such that it support the party's position, this party may justify its position by using this precedent. If no precedents exactly match or subsume the current case, parties may argue about the importance of the differences.

Despite the great synergy between legal reasoning, as just described, and CBR, both the theoretical works (*e.g.* [45, 198, 37]) and the developed systems (*e.g.* [29, 210, 22]) that addresses reasoning with legal precedents focus their attention in integrating cases, precedents, within the reasoning process, rather than actually proposing CBR applications. Although in this kind of legal reasoning, past cases influence the final decisions, the degree of interpretation to which cases are subjected makes a statement such as *-solving new problems based on the solutions of similar past problems-* unsuitable to capture what past cases represent in reasoning with legal precedents. Thus, as it is done in [45] we believe it is more appropriate to regard these works as formalisations for *reasoning with cases* rather than CBR, as they are more communally referred to.

More in line with our approach is Sycara's PERSUADER System [215]. This seminal work combines concepts of negotiation and CBR. PERSUADER proposes a mediated system in which two *human* agents, one representing a *company* and another the *trade union*, can negotiate over a contract intended to resolve labour management disputes, where a contract consists of a number of attributes such as salaries, pensions and holidays. The negotiation begins by a contract proposed by PERSUADER based in great part on the participants' submitted individual objectives. If the participants accept the proposed contract the negotiation ends. Otherwise, PERSUADER may either issue a new contract or try to persuade the agents to accept a proposed contract. To address this persuasion problem, the mediator builds models of the two participants, this will help select different persuasion strategies intended to promote the agents' perceived value of the objectives realised by the contract and demote the perceived value of the agent's objectives not met by proposed contract. These persuasion strategies are realised through the submission of arguments. Therefore, for a given contract, the task of the mediator is to find the most effective argument that will increase the agent's perceived payoff for that contract.

These strategies not only involve choosing which issues to address but also, how to address them. Based on persuasion psychology [123] PERSUADER associates different *convincing power* to different kinds of arguments. In particular, arguments with little or non logical appeal may be selected because they are deemed effective from a psychological point of view. For instance, arguments of the kind *Appeal to a theme* such as '*The offer is intended to help all of us, in AT&T to work together to build the future. Let's get on with it!*' are deemed very effective.

Arguments may be constructed from scratch (using objective graph search and multi-attribute utility) or extracted from a case-base, that is using CBR. To use arguments from the case-based, PERSUADER's memory is organised so that to an argument are associated: the addressed contract issue, the agent to which is intended (*i.e.* the union or the company agent) and which argumentation goal and strategy it fulfills (*e.g.* increase the importance of an objective). Also is associated the kind of argument it embodies (*e.g. Appeal to a theme* or *Appeal to universal principle*) as well as information about the effectiveness of the argument depending on various external economic conditions.

Therefore, the idea is that given a new negotiation context, PERSUADER can retrieve from the case-based a list of arguments that have been used in previous similar negotiations, indicating the target contract issue, the agent the argument is intended to persuade and the intended strategy. That is, the CBR is used as a library of previously used arguments at the mediator agent's disposal.

While valuable, PERSUADER's approach deviates substantially form our purpose by addressing primarily on strategies and the psychological impact of the arguments in a competitive environment. Furthermore, being a very early proposal (ten years before influential works in argumentation such as Dung's argumentation framework [80] or Walton's argument schemes [231]), PERSUADER's underlying argumentation model is rather weak.

A somewhat similar use of CBR is proposed in [122] within the context of the HERMES system [121] intended for collaborative deliberations among *human* agents. HERMES proposes a structured forum, following the IBIS [138] tradition (see §2.2). Hence, arguments can be *pro* or *con* an issue, or a statement, where arguments are written in free text. HERMES' CBR is intended to provide support to the users in exploring previous deliberations (stored as structured discussion forums) and with a number of available filters, help the user identify potentially useful arguments for their target deliberation. One important issue of this proposal is that the CBR presented in [122] leaves most of the reasoning cycle unspecified.

Other works proposing CBR and some form of argumentation are for example [212] or [168]. The former proposes the use of CBR to support autonomous agents to negotiate the use of their resources (sensors that track several mobil objects). In this work one agent may require occasionally some resources (sensors) from a third agent for which she will have to persuade her to share her resources. The CBR will help select the appropriate information that will persuade that particular agent to collaborate. The later work proposes a collaborative decision making among software agents framed as a classification problem. A solution to a target problem is arrived to by properly classifying the problem at hand into a solution class, a set of similar cases that share the same solution. Each agent has its own case base, containing different cases and thus yielding different classification of the target problem. Given a problem, the decision making begins with one of the agents proposing a classification based on its stored cases. Other agents may then propose different classifications (counter-argue) or share cases which contradict a given classification (counter-example). Through this *argumentative* process agents help improve the quality of the initially proposed classification.

Case Based Reasoning (CBR) has proven to be an appropriate reasoning and learning approach for ill-structured domains, where capturing experts' knowledge is difficult and/or the domain theory is weak or incomplete. However, CBR developers still have to face problems such as having to decide *how* to represent a case, *what* are the relevant factors for comparing them and *how* to retain new cases that encode, in a useful way, both the success and failure of the cases' proposed solutions. On the other hand, argumentation has proven to be a suitable approach for reasoning under uncertainty, with inconsistent knowledge sources, and in dialog based communication. However, one important problem in argumentation is *how* to reuse the knowledge encoded in the arguments used in previous dialogs in a somewhat automated fashion. Applications from the legal domain require high degree of sophistication in human interpretation regarding identifying what are the relevant factors, in particular, cases are not added or updated automatically. Systems such as PERSUADER or HERMES rely on human agents to asses the relevance of retrieved arguments.

In the following sections we introduce the models CBRc, in which previous stored *ProCLAIM* deliberations can be reused to help resolve target deliberations, where the full reasoning cycle can be automatised.

9.2 Introducing the CBRc

Once a deliberation regarding the safety of proposed action has concluded, the tree of arguments $\mathbb{T}$ contains **all** the facts and actions deemed *relevant* for assessing the main proposed action's safety, from the view point of domain experts, guidelines, regulations and past collected evidence. Furthermore, if the proposed action is deemed safe and eventually performed, $\mathbb{T}$ can then be updated by the appropriate PAs so as to record the actual outcome of the action's performance. For instance, if the recipient of a lung of a donor with smoking history and no chronic obstructive pulmonary disease (COPD) rejects the transplanted organ, the RA updates $\mathbb{T}$ so that argument $A2$ is preferred to $A3$ as depicted in figure 9.1. That is, the RA changes $\mathbb{T}$ from that in figure 9.1b to that in figure 9.1c. Note that after this update the arguments in $\mathbb{T}$ are no longer *presumptive* but *explanatory* in nature. These arguments describe the actual outcome of the performed action. And so, the updated $\mathbb{T}$ can be reused as evidence for resolving future similar deliberations, which is the CBRc's role. Indeed as we discuss in §9.3 the produced tree of argument $\mathbb{T}$ is what best represents a case in the CBRc.

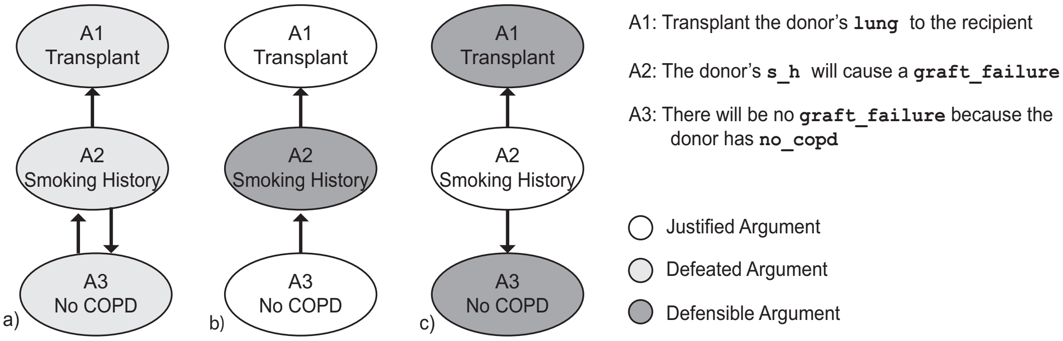


Figure 9.1: The tree of arguments $\mathbb{T}$ encoding the reasoning regarding the safety of a lung transplant when the donor has a smoking history but no COPD

There are two aspects of the schemes defined here that facilitate CBRc's task: *1)* the specificity of the schemes in the ASR (as described in §7) and *2)* that *relevant* facts and actions are introduced in a structured fashion, each singled out and introduced step by step. The schemes' specificity allows to efficiently identify potentially similar cases. The idea is that cases in which the same specialised schemes (reasoning patterns) were used, may be similar. Thus, by organising the case-base in terms of the argument schemes, a set of broadly similar cases can effectively be retrieved. The latter aspect of the schemes facilitates a more detailed comparison between cases on the basis of the similarity between the cases' introduced *relevant* facts and actions. We illustrate with the above example from the medical scenario.

Suppose the deliberation consisted only of the arguments $A1$, $A2$ and $A3$ depicted in figure 9.1. All three arguments instantiate the schemes $AS1_T$, $AS2_{T\_gf1}$ and $AS3_{T\_gf1\_1}$ (see §7.1) encoding the following reasoning pattern:

*–An organ **O** was intended for transplantation, and while the donor's condition **P1** was considered as a risk factor for causing a graft failure, the donor's condition **P2** was thought to minimise this risk–*

Thus by retrieving all the deliberations which consisted of these three schemes we obtain cases that are already quite similar to our target case. So now, if we take from these past cases those where the organ `O` is a `lung`, the condition `P1` is *similar*[1] to `s_h` and where the treatment `P2` is *similar* to ¬`COPD`, we obtain the desired set of cases from which to evaluate the target case on an evidential basis. Thus, while the schemes are used as a heuristics for a first, broad case retrieval, the similarity between cases is ultimately derived from a similarity between the facts highlighted as relevant for the decision making. This process, which is part of the CBRc reasoning cycle, is described in detail in §9.5. In this same section, we also describe how, having retrieved the set of similar cases, represented by trees of arguments, the CBRc can derive its preference assignment on mutually attacking arguments. The retrieved trees of arguments represent cases where the action was already performed, and thus it only contains asymmetric attacks. In the example this may results in two types of retrieved trees of arguments: $\mathbb{T}^{+}$, where the arguments *similar* to $A3$ asymmetrically attacks and so defeat those *similar* to $A2$, *i.e.* the action was successful; and $\mathbb{T}^{-}$, where the arguments *similar* to $A2$ defeat those *similar* to $A3$, *i.e.* the lung was rejected despite the donor not having have had a COPD. If the incidence of $\mathbb{T}^{+}$ cases *significantly* outnumber the $\mathbb{T}^{-}$ cases then argument $A3$ would be deemed preferred to $A2$, otherwise either argument $A2$ would be deemed preferred to $A3$ or, if there is not enough evidence so as to prefer one argument over the other, their conflict will remain unresolved. In this same process, where past cases are *reused* to resolve a target deliberation, the CBRc may propose the submission of additional arguments, this is further discussed in §9.5.2.

Assuming the proposed action is performed, the CBRc reasoning cycle concludes when the outcome of the action is fed back into the case's $\mathbb{T}$, and then, this *revised* tree of argument is *retained* in the CBRc' memory. Then this resolved case can later be *retrieved* and *reused* to help solving similar target cases.

Having sketched out the basic ideas on how the CBRc operates, let us now describe each of these aspects in more detail. Let us begin by introducing the CBRc's case description.

9.3 Case Description

A case in *ProCLAIM* is mainly defined by the tree of argument $\mathbb{T}$ and the set of facts $\mathbb{C}_F$ constructed during the deliberation, which may be updated at a later stage (*e.g.* after the proposed action is performed). So to recall, $\mathbb{T}$ is a tree of interacting arguments (a Dung-like graph) where each argument instantiate a scheme of the ASR. While $\mathbb{C}_F$ is the set of facts the PAs have submitted during the deliberation. The broad idea is that $\mathbb{T}$ indicates which are the relevant factors for the decision making (and so for he case comparison) and $\mathbb{C}_F$ is attached to the case description to provide additional contextual information when adaptations are required as we discuss in §9.5.2.

We would also like to add a notion of evidential weight to the description of a case. Note that once a deliberation concludes, the proposed action is either deemed safe or unsafe. If deemed safe, the action may then be performed. Once an action is performed, the outcome counts as evidence for the action's safety. As we will see in §9.5.3, the PAs directly involved in the action performance, will feedback the relevant outcomes of the performed action into $\mathbb{T}$. This time, however, PAs' submitted arguments are no longer *presumptive* in

[1] We discuss the similarity between facts, actions and arguments in §9.4.

nature but they *explain* the reasons why the action turned out to be safe or unsafe. In other words, arguments submitted after the action is performed are based on evidence, and so $\mathbb{T}$s' associated with performed actions carry an evidential weight. We can thus denote as **F** the **phase** at which the action assessment is made, where phase 0 denotes assessments made at deliberation time and so carrying no evidential weight, and where assessments made at later phases ($F > 0$) will have associated more evidence.

At the transplant scenario we define three phases: $F = 0$ for the initial, on deliberation, assessment; $F = 1$ for assessments made after the organ is extracted from the donor but before it is implanted into the recipient and $F = 2$ for assessments made after the organ is implanted into the recipient. Hence, cases with $F = 2$ carry more evidential weight than cases with $F = 1$, while cases with $F = 0$ do not provide any evidence. In future work we intend to define a more fine grained distinction among cases of phase $F = 2$ distinguishing, in particular, between long and short term survival success.

Therefore, a case description in *ProCLAIM* is defined by the tuple: $<\mathbb{T}$, $\mathbb{C}_F$, $F>$.

9.4 The CBRc' Memory Organisation

The case base is organised as a hierarchical structure based on the cases' associated trees of arguments. This hierarchical relation accounts only for the structure trees of arguments, ignoring both the schemes instantiation and the direction of the attack relation. In order to define the case base organisation we must introduce the following definitions.

Definition 9.1 *Let $\mathbb{T}$ be a tree of arguments, each argument instantiating schemes of the ASR. Let us define p_S as the canonical projection of $\mathbb{T}$ which removes from $\mathbb{T}$ the schemes instantiation and the attack relations. That is, $p_S(\mathbb{T})$ is an undirected tree labelled only with the schemes' and CQs' ids.*

Let us now consider $\mathbb{T}_1$ and $\mathbb{T}_2$ be two trees of arguments, we can define the partial ordering $\preceq_S$ such that:

$\mathbb{T}_1 \preceq_S \mathbb{T}_2$ *if and only if* $p_S(\mathbb{T}_1)$ *is a subtree of* $p_S(\mathbb{T}_2)$.

In the above section we have described a case as the tuple $<\mathbb{T}$, $\mathbb{C}_F$, $F>$. To refer to a case's C associated tree of arguments we write $tree(C)$. Let $C1$ and $C2$ be two cases such that $p_S(tree(C1)) = p_S(tree(C2))$, then we say that in both cases the same reasoning lines were used. Returning to the example illustrated in figure 9.1, let us take $tree(C1)$ and $tree(C2)$ to include only the chain of schemes $AS1_T - AS2_{T\_gf1} - AS3_{T\_gf1\_1}$. Any case $C3$ such that $tree(C1) \prec_S tree(C3)$, would contain the chain $AS1_T - AS2_{T\_gf1} - AS3_{T\_gf1\_1}$ and at least one more argument, in reply to either the instantiation of $AS1_T$, $AS2_{T\_gf1}$ or $AS3_{T\_gf1\_1}$.

This structural relation among cases allows the CBRc organise cases based only on the used chain or reasoning.

Definition 9.2 *Let CB be the set of all cases in CBRc. Let $\mathcal{T}$ be the set of all tree structures. That is $\mathcal{T} = \{T \mid T = p_S(tree(C)), C \in CB\}$. Let $\mathcal{F}$ a function such that, for $T \in \mathcal{T}$, $\mathcal{F}(T) = \{C \mid C \in CB, p_S(tree(C)) = T\}$. Then, we organise the case base as a tuple: $< M, \preceq >$, where $M = \{< T, \mathcal{F}(T) > \mid T \in \mathcal{T}\}$ and $\preceq$ is a partial ordering such that $< T1, \mathcal{F}(T1) > \preceq < T2, \mathcal{F}(T2) >$ if and only if $T1$ is a subtree of $T2$.*

Hence, for any given case C we can obtain the set of all cases, that share the same reasoning lines with C be retrieving the tuple $< p_S(tree(C)), S > \in M$. Through the partial ordering $\preceq$ we can retrieve all cases which subsume the reasoning lines used in C, which is the first step in the CBRc reasoning cycle.

9.5 The CBRc' Reasoning Cycle

9.5.1 Retrieve

The reasoning cycle begins with the retrieval process, in which, given a target problem the relevant cases for solving it are retrieved from the Case-Base. Before we describe the retrieval process we must first provide some definitions. In particular, we introduce a notion of cases similarity, which is based on a distance measure between the terms instantiating the schemes and CQs.

Definition 9.3 *Let O be the ontology whose terms instantiate the argument schemes of ASR. We assume O to have a tree structure, with its nodes being the terms of the ontology and with weighted edges representing the distance between terms. We take the children of node, to be* more specific than *its parent.*

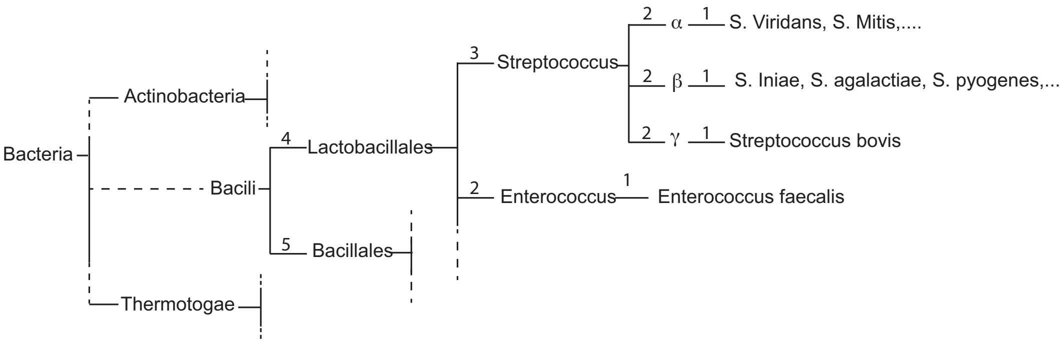


Figure 9.2: Fragment of a medical Ontology

Figure[2] 9.2, depicts a fragment of a medical ontology. For convenience to refer to a term of the ontology we will write $t \in O$.

[2] Extracted from the Bacterial Bioinformatics Resource Center (PATRIC): http://patricbrc.org/. The numerical values are only for illustrative purposes.

Definition 9.4 *Let $t1, t2 \in O$, we denote $LCA(t1, t2)$, the* lower common ancestor *of $t1$ and $t2$. Let $path_w(t1, t2)$ be the sum of the weights of the edges connecting the two nodes $t1$ and $t2$. Then, distance between two terms $t1, t2 \in O$ is defined as:*

$$\delta_O(t1, t2) = max(path_w(LCA(t1, t2), t1), path_w(LCA(t1, t2), t2))$$

According to figure 9.2 the distance between the two types of bacterias *streptococcus viridans* and *enterococcus faecalis* is 6. The lower common ancestor of these two terms is *lactobacillales* and the maximum value of $path_w$(`s_viridans`, `lactobacillales`) and $path_w$(`lactobacillales`, `enteroc_faecalis`) is 6. Similarly, we can derive from figure 9.2 that δ_O(`s_viridans`,`s_bovis`) =3.

For terms that need not be compared because they are particularities of each case. Such as the donor and recipient in the transplant scenario, the distance between two instances would always be zero. That is, for any two donors `d1` and `d2` or two recipients `r1` and `d2`, then δ_O(`d1`,`d2`) =0 and δ_O(`r1`,`r2`) =0.

In order to define the distance between arguments, let us represent argument instatiation in a more convenient way. The arguments instantiating schemes of the ASR, can be written as: $AS\_id(\mathcal{C}, \mathcal{A}, R, A, S, g)$, where AS_id is the scenario specific scheme's id and $\mathcal{C}$, $\mathcal{A}$, R, A, S and g are respectively the local context of facts, the local context of actions, the introduced set of facts, the introduced set of actions, the set of side effects and the undesirable goal. Thus $\mathcal{C}$, $\mathcal{A}$, R, A, S and g are the scheme's AS_id instantiation. For convenience, let us rewrite this instantiated arguments as $AS\_id(x_1, ..., x_n)$ where x_i ($i \in [1, n]$) are the terms that instantiate the scheme. Similarly, we denote $CQ\_id(r)$ as the instantiation of the CQ CQ_id, where r is the challenged fact. The distance between two arguments instantiating the same argument scheme is defined as follows.

Definition 9.5 *Let $A1$ and $A2$ be two arguments instantiating scheme AS. That is, $AS(x_1, ..., x_n)$ and $AS(y_1, ...y_m)$ where the x_i (i=1,..,n) and y_j (j=1,..,m) are the terms of O with which the scheme AS is instantiated. Because $A1$ and $A2$ instantiate the same scheme, $n = m$. The distance between the two arguments is defined as:*

$$\delta_{arg}(AS(x_1, ..., x_n), AS(y_1, ...y_n)) = max_{i=1}^{n}(\delta_O(x_i, y_i))$$

δ_{arg} also applies for CQs: $\delta_{arg}(CQ\_id(r1), CQ\_id(r2)) = \delta_O(r1, r2))$.

The distance between two trees of arguments, sharing the same schemes, can now be computed, and thus a notion of similarity between cases can be defined.

Definition 9.6 *Let $\mathbb{T}_1$ and $\mathbb{T}_2$ be two trees of arguments such that $\mathbb{T}_1 =_S \mathbb{T}_2$. Thus both $p_S(\mathbb{T}_1)$ and $p_S(\mathbb{T}_2)$ are equal, with their nodes being $arg_0, \ldots, arg_n$. Then, the distance between $\mathbb{T}_1$ and $\mathbb{T}_2$ is defined as:*

$$\delta(\mathbb{T}_1, \mathbb{T}_2) = max_{i=1}^{n}(\delta_{arg}(arg_i(x_1, \ldots, x_{m_i}), arg_i(y_1, \ldots, y_{m_i}))).$$

Given a real number $k > 0$ we can say that two trees of arguments, $\mathbb{T}_1$ and $\mathbb{T}_2$ are ***k-similar*** *if and only if $\mathbb{T}_1 =_S \mathbb{T}_2$ and $\delta(\mathbb{T}_1,\mathbb{T}_2) \leqslant k$.*

We say that two cases $C1$ and $C2$ are ***k-similar*** *if $tree(C1) = tree(C2)$ are k-similar.*

It is worth noting that when two arguments, say $A1$ and $A2$, are compared it is within the context of the trees of arguments they belong to. Hence the only terms that must be compared between $A1$ and $A2$ are the introduced set of facts R and actions A and the set of side effects S. In other words, the local context of facts $\mathcal{C}$ and actions $\mathcal{A}$ can be ignored as their content is compared when comparing the arguments to which $A1$ and $A2$ reply to. The undesirable goals g can also be ignored since they are embedded in the scheme the arguments instantiate. In what follows, we will write the schemes' instantiation as $AS_id(x_1, ..., x_n)$ with $x_1, ..., x_n$ the free variables of the schemes, which are the only terms that need to be compared.

Note that the k-similarity between trees of arguments is independent from the direction of the attack relations. Thus, for simplicity we will assume in this section that the trees of arguments are non directed. The direction of the attack relation becomes important at the reuse process, which we describe in the following section.

To illustrate the notion of k-similarity, let us take four cases C_1, C_2, C_3, C_4, depicted in figure 9.3, in which a kidney has been transplanted. The four cases are such that $p_S(tree(C_i))$ = $AS1_T - AS2_{T\_inf1} - AS5_{T\_inf1}$ for $i = 1, .., 4$ and so share the same reasoning line (see §7):

*–An organ **O** was intended for transplantation. The donor had some condition **P1** which would bring about a severe infection **P2** in the recipient. Treatment **T** for the recipient was proposed to prevent this infection–*

In C_1 the donor had a *streptococcus viridans endocarditis* and *penicillin* was used to prevent the *streptococcus viridans infection* on the recipient. Case C_2 is identical with the exception that the infecting bacteria was a *streptococcus bovis* instead of a *viridans*. In the third case C_3 the donor had a more resistant bacteria, *enterococcus faecalis*, which required a more intensive antibiotic, which is *teicoplanin*. In the final case C_4 the donor had *aspergillosis*, a fungal infection caused by *aspergillus*. The proposed treatment in this case was *voriconazole* [226].

With the distances illustrated in figure 9.3 we can derive that C_1 and C_2 are 3-similar, since $\delta(tree(C_1), tree(C_2)) = 3$. C_1 and C_3 are 6-similar, while C_1 and C_4 are only 55-similar. Thus, if we take for instance a threshold 8, then while C_1, C_2 and C_3 would be deemed similar while $C4$ would not.

The use of existing ontologies for arguments comparison is a good starting point. Terms that are neighbors in an ontology are likely to share important properties. However, in future work we intend to address this assumption with more care.

Besides the notion of k-similarity which will allow to retrieve those cases which share *similar* arguments, we have to introduce further notation and definitions for the retrieval

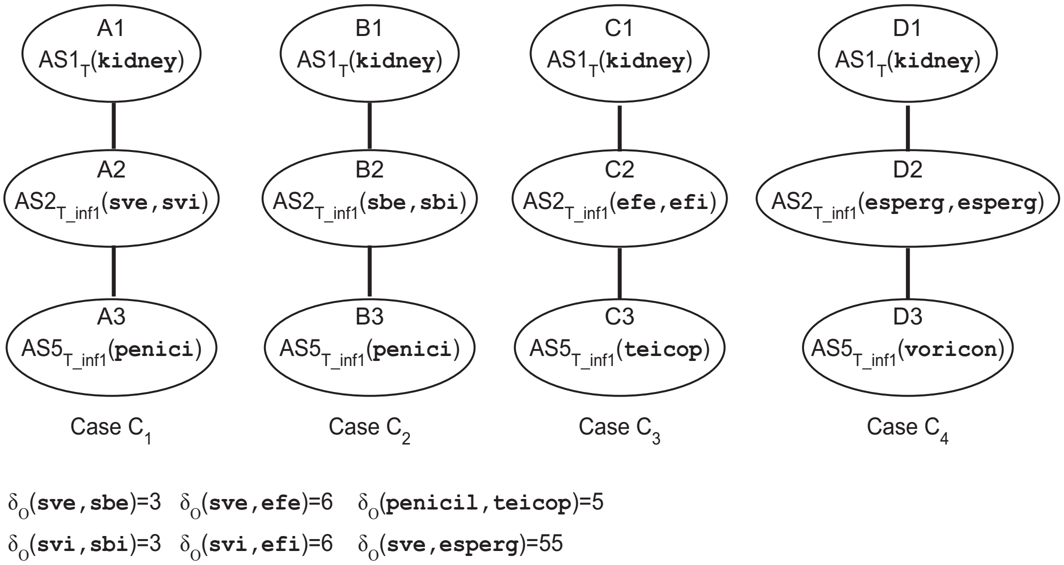


Figure 9.3: Four cases sharing the same reasoning line, however with different instantiations.

of cases that may contain additional arguments, potentially relevant for the target problem. Suppose for example that the associated tree of argument of a target case C_a depicted in figure 9.4 is such that it contain only the two arguments $AS1_T$(kidney) and $AS2_{T\_inf1}$(sve,svi). Despite the fact that cases C_1, C_2 and C_3 are relevant for solving the target case C_a only with the notion of case similarity introduced so far these cases would not be retrieved. This is because $p_S(tree(C_a)) \neq p_S(tree(C_i))$ for $i = 1, .., 3$

Definition 9.7 *Let $\mathbb{T}_1$ and $\mathbb{T}_2$ be two trees of arguments such that $\mathbb{T}_1 \preceq_S \mathbb{T}_2$. Let, $\mathbb{T}_{2sub}$ the subtree of $\mathbb{T}_2$ such that $\mathbb{T}_1 =_S \mathbb{T}_{2sub}$, then:*

$$\mathbb{T}_1 \preceq_{D_k} \mathbb{T}_2 \quad \textit{if} \ \ \mathbb{T}_1 \ \textit{and} \ \mathbb{T}_{2sub} \ \textit{are} \ k - similar.$$

Figure 9.4 depicts thee trees of arguments associated to the cases C_a, C_b and C_c such that $tree(C_a) \preceq_{D_k} tree(C_b) \preceq_{D_k} tree(C_c)$, $k \geq 0$. Let us suppose C_a is the target problem and C_b and C_c are two retrieved cases. It is clear that C_b is relevant as it highlights the same facts as relevant. However, the retrieved case C_c is only relevant in so far the donor at the target problem has Hepatitis C. If indeed the donor has Hepatitis C, then argument $C4$ should be reused (*applied*) on the target problem. Otherwise, C_c should be discarded[3]. To capture this notion let us introduce the notion of applicability.

Definition 9.8 *Let $\mathbb{T}$ be a tree of arguments, and let $facts(\mathbb{T})$ be the set of all facts introduced in $\mathbb{T}$. Then, for $K \geq 0$, we say that $\mathbb{T}$* k-applies *to a case C = $<\mathbb{T}_0$,$\mathbb{C}_F$, $F>$ if and*

[3] We discuss in §9.7 whether C_c could still be reused with some adaptations.

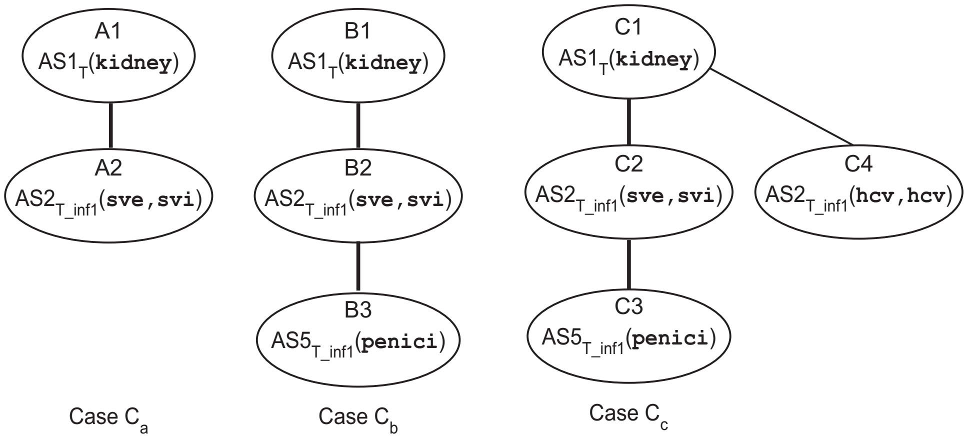


Figure 9.4: Three trees of arguments with $tree(C_a) \preceq_{D_k} tree(C_b) \preceq_{D_k} tree(C_c)$

only if $\forall r1 \in facts(\mathbb{T})$ $\exists r2 \in \mathbb{C}_F$ *such that,* $\delta_O(r1, r2) \leqslant k$*, and we write it as* $\mathbb{T}$ ***apply***$_k$ *C.*

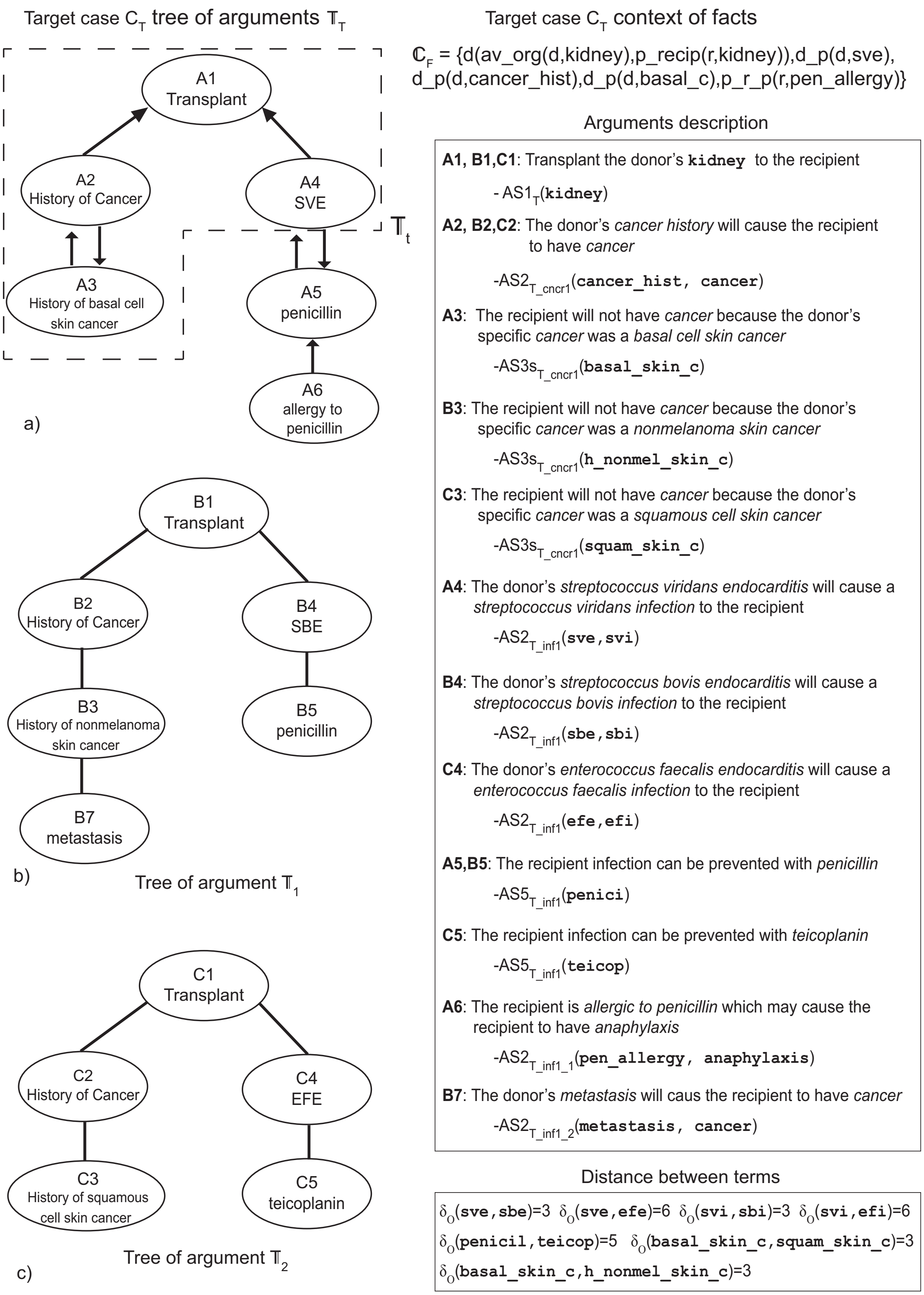
As discussed above $tree(C_b)$ is k-applicable on C_a, for $k \geq 0$. On the other hand $tree(C_c)$ is k-applicable on C_a only if the context of facts associated to C_a there is a fact similar to d_p(d,hcv). Another example is depicted in figure 9.5. In this example we can see that, while $\mathbb{T}_2$ $apply_k$ C_T, for $k \geq 6$; $\mathbb{T}_1$ does not apply to C_T because there is no fact in $\mathbb{C}_F$ similar to d_p(d,metastasis).

With the above introduced notation and definition we can now proceed to describe retrieval process. Let us suppose the target case is C_T = $<\mathbb{T}_T, \mathbb{C}_{F_T}, 0>$. The retrieval process mostly takes into account the target case' tree of arguments $\mathbb{T}_T$. In particular, it takes into account a subtree of $\mathbb{T}_T$ in which all the chain of schemes starting from an argument introducing a complementary action proposal are removed. That is, arguments instantiating scheme $AS5$ and their children are removed. The reason for this is that the core aspect that makes two cases similar is that they share the same relevant facts. The complementary courses of actions may be ignored at an early stage of case comparison. Figure 9.5, illustrates the tree of a target case $\mathbb{T}_T$ and its trimmed version $\mathbb{T}_t$.

Let CB be the set of all cases, C_T = $<\mathbb{T}_T, \mathbb{C}_{F_T}, 0>$ the target case and $\mathbb{T}_t$ the subtree of $\mathbb{T}_T$ to which the branches starting with arguments instantiating scheme $AS5$ have been removed. Then the retrieval process involves these three steps:

1. Retrieve a set $R1$ containing all cases that include the chains of reasoning used in the target deliberation up to the introduction of complementary action proposals. It should be noted that this first process is a basic query to the Case Base memory:

$$R1 = \{c \mid c \in CB,\ \mathbb{T}_t \preceq_S tree(c)\}$$


Target case C_T tree of arguments $\mathbb{T}_T$
A1
Transplant
A2
History of Cancer
A4
SVE
$\mathbb{T}_t$
A3
History of basal cell skin cancer
A5
penicillin
A6
allergy to penicillin
a)
B1
Transplant
B2
History of Cancer
B4
SBE
B3
History of nonmelanoma skin cancer
B5
penicillin
B7
metastasis
b)
Tree of argument $\mathbb{T}_1$
C1
Transplant
C2
History of Cancer
C4
EFE
C3
History of squamous cell skin cancer
C5
teicoplanin
c)
Tree of argument $\mathbb{T}_2$

Target case C_T context of facts

$\mathbb{C}_F$ = {d(av_org(d,kidney),p_recip(r,kidney)),d_p(d,sve), d_p(d,cancer_hist),d_p(d,basal_c),p_r_p(r,pen_allergy)}

Arguments description

A1, B1,C1: Transplant the donor's `kidney` to the recipient

- AS1$_T$(`kidney`)

A2, B2,C2: The donor's *cancer history* will cause the recipient to have *cancer*

-AS2$_{T\_cncr1}$(`cancer_hist, cancer`)

A3: The recipient will not have *cancer* because the donor's specific *cancer* was a *basal cell skin cancer*

-AS3s$_{T\_cncr1}$(`basal_skin_c`)

B3: The recipient will not have *cancer* because the donor's specific *cancer* was a *nonmelanoma skin cancer*

-AS3s$_{T\_cncr1}$(`h_nonmel_skin_c`)

C3: The recipient will not have *cancer* because the donor's specific *cancer* was a *squamous cell skin cancer*

-AS3s$_{T\_cncr1}$(`squam_skin_c`)

A4: The donor's *streptococcus viridans endocarditis* will cause a *streptococcus viridans infection* to the recipient

-AS2$_{T\_inf1}$(`sve,svi`)

B4: The donor's *streptococcus bovis endocarditis* will cause a *streptococcus bovis infection* to the recipient

-AS2$_{T\_inf1}$(`sbe,sbi`)

C4: The donor's *enterococcus faecalis endocarditis* will cause a *enterococcus faecalis infection* to the recipient

-AS2$_{T\_inf1}$(`efe,efi`)

A5,B5: The recipient infection can be prevented with *penicillin*

-AS5$_{T\_inf1}$(`penici`)

C5: The recipient infection can be prevented with *teicoplanin*

-AS5$_{T\_inf1}$(`teicop`)

A6: The recipient is *allergic to penicillin* which may cause the recipient to have *anaphylaxis*

-AS2$_{T\_inf1\_1}$(`pen_allergy, anaphylaxis`)

B7: The donor's *metastasis* will caus the recipient to have *cancer*

-AS2$_{T\_inf1\_2}$(`metastasis, cancer`)

Distance between terms

δ_O(`sve,sbe`)=3 δ_O(`sve,efe`)=6 δ_O(`svi,sbi`)=3 δ_O(`svi,efi`)=6
δ_O(`penicil,teicop`)=5 δ_O(`basal_skin_c,squam_skin_c`)=3
δ_O(`basal_skin_c,h_nonmel_skin_c`)=3

Figure 9.5: An example of a target case C_T and two potentially relevant trees of arguments $\mathbb{T}_1$ and $\mathbb{T}_2$. Note that while $\mathbb{T}_T \not\preceq_{D_k} \mathbb{T}_i$, for $i = 1, 2$, for any k, $\mathbb{T}_t \preceq_{D_k} \mathbb{T}_i$, for $i = 1, 2$, for $k \geq 6$. On the other hand, while $\mathbb{T}_2$ $apply_k$ C_T, $\mathbb{T}_1$ does not apply to C_T because `d_p(d,metastasis)` is not similar to any fact in $\mathbb{C}_F$

2. Given a threshold $k \geq 0$, retrieve from $R1$ the subset $R2$ which contain only those cases which have highlighted k-similar facts to those in $\mathbb{T}_t$:

   $$R2 = \{c \mid c \in R1,\ \mathbb{T}_t \preceq_{Dk} tree(c)\}$$

3. From $R2$, retrieve only those cases which associated trees of arguments apply to the target case:

   $$R3 = \{c \mid c \in R2,\ tree(c)\ apply_k\ C_T\}$$

The set $R3$, contains all the cases of CB relevant for resolving the target case C_T. All cases in $R3$ contain the reasoning lines used in $\mathbb{T}_t$. All the facts deemed as relevant in $\mathbb{T}_t$ have also bee deemed relevant in the cases of $R3$, or facts which are similar enough. Furthermore, all the arguments used by the cases in $R3$ are all *applicable* in the target situation.

9.5.2 Reuse

The reasoning cycle continues with reuse process, in which a solution is proposed based on the retrieved cases. Each case of $R3$ encodes a possible solution to the target problem. In this section we show how these cases can be used to propose a single solution proposal. In particular, how additional arguments may be proposed by the CBRc based on past similar cases, and equally how a preference relation between mutually attacking arguments is derived from the retrieved cases.

In the above section we have introduced the notion of the *applicability* of a tree of argument on a case. Let us now *apply* the retrieved trees of arguments on the target case.

Definition 9.9 *Let $\mathbb{T}$ be a tree of arguments and C = $<\mathbb{T}_0, \mathbb{C}_F, F>$, such that $\mathbb{T}$* k-applies *to a case C for a given $k > 0$. The* application *of $\mathbb{T}$ over C_T, denoted as $apply(\mathbb{T}, C_T)$, results in a tree of arguments equal to $\mathbb{T}$ but which instantiating facts are replaced by the facts in $\mathbb{C}_F$. That is, each fact $r_1 \in facts(\mathbb{T})$ is replaced by the corresponding fact $r_2 \in \mathbb{C}_F$, with $\delta_O(r1, r2) \leqslant k$[4]. Where $facts(\mathbb{T})$ is the set of all facts introduced in $\mathbb{T}$.*

By applying the trees of arguments of $R3$ to the target case, the CBRc adapts those retrieved cases to accommodate to the target situation. Namely, each of the trees of arguments in $\{apply(tree(c), C_T) \mid c \in R3\}$ encode a possible outcome of the target deliberation. The idea now is to merge all these proposals into a single tree of arguments. But, we want to do this taking into account the evidential support of each proposed solution. Broadly speaking, trees of arguments shared by many cases would have a stronger evidential support than

[4] We are assuming that for the given k, there do not exist $r_a, r_b \in facts(\mathbb{T})$ such that $\delta_O(r_a, r_b) \leqslant k$, nor $r_c, r_d \in \mathbb{C}_F$ such that $\delta_O(r_c, r_d) \leqslant k$

those trees shared by only a few cases. It would seem natural that this distinction should have an effect on how these trees of arguments are put together.

Let us consider the target case C_T and let us denote as $\mathcal{T}_s$ the set of all potential solutions derived from $R3$. That is $\mathcal{T}_s = \{apply(tree(c), C_T) \ | \ c \in R3\}$. Let us also consider $\mathcal{F}$ to be the set of all defined phases (e.g. $\mathcal{F} = \{0, 1, 2\}$). And finally let $f(\mathbb{T}, F)$ = $\{<\mathbb{T}_0$, $\mathbb{C}_F$, $F >$ $\ | \ $ $<\mathbb{T}_0$, $\mathbb{C}_F$, $F > \in R3, apply(\mathbb{T}_0, C_T)$ $=\mathbb{T}\}$, for $\mathbb{T}\in \mathcal{T}_s$. Then we can group all cases into a set of provisional solution proposals:

$$SP_0 = \{<\mathbb{T}, S, F > \ | \ \mathbb{T}\in \mathcal{T}_s, F \in \mathcal{F}, S = f(\mathbb{T}, F)\}.$$

Therefore, an element of SP_0 is a tuple $<\mathbb{T}_0, S_0, F_0>$ where S_0 is a set of cases that share a similar tree of argument $\mathbb{T}_0$, and all cases in S_0 have been resolved in the same phase F_0. In particular, not only all cases in S_0 are k-similar, but they all had the same outcome. The attack relations of their associated trees of arguments are all equal. Note that we can obtain a notion of evidential support for a given $\mathbb{T}_0$ by taking into account the number of cases in S_0 and the associated F_0.

We will assume that there is an application specific function which, given the tuple $<\mathbb{T}, S, F>$, it gives the **evidential support** associated to the solution $\mathbb{T}$. We will denote this function as $EV(\mathbb{T}, S, F)$ and we wil assume as well that $EV(\mathbb{T}, S, F) \in [0, 1)$. Broadly speaking, the idea is that the larger $|S|$ and F are, the closer $EV(\mathbb{T}, S, F)$ would get to 1, indicating a greater evidential support.

While the values of $|S|$ and F provide some notion of evidence, there are also other aspects that should also be taken into account when computing the evidential support. For example, the actual distance between the cases in S and the target case. If all cases in S are identical to the target case, the evidential support should be higher than if all cases are exactly k-simialr but not $(k-1)$-similar, for any $k > 1$. Another aspect to take into account is the actual outcome of the proposed action, if in fact performed. For an action to be deemed safe the successful outcomes should outnumber the failures, particularly when dealing with safety critical actions.

Once we assume a function $EV(\mathbb{T}, S, F) \in [0, 1)$ we can consider a threshold value $Suf_{ev} \in [0, 1)$ such that if $ES(\mathbb{T}, N, F) > Suf_{ev}$ then we say that $\mathbb{T}$ has **sufficient evidential support**.

With the notion of *sufficient evidential support* we can filter out from the set SP_0 all the $\mathbb{T}$ which have not gathered sufficient evidential support (*e.g.* if $F = 0$, or $|S| < 5$). Namely, if we take SP_0, $ES(\mathbb{T}, S, F)$ and Suf_{ev} as described above, we can then consider the set of solution proposals as:

$$SP = \{<\mathbb{T}, S, F > \ | \ <\mathbb{T}, S, F > \in SP_0, ES(\mathbb{T}, S, F) > Suf_{ev}\}$$

Each element of the set SP is not only a solution proposal but has sufficient evidential support. Figure 9.6 illustrates the set SP of target case. In this example, we can see that the attack direction of each proposed solution may be in conflict, for instance, argument $A4$ is preferred to $A5$ in $\mathbb{T}1$ but not in $\mathbb{T}2$. There may be arguments in SP not accounted for in the target case (*e.g.* arguments $A6$ and $A7$ in $\mathbb{T}3$) but there may also be the case that arguments in the target case are not in SP, such as argument $A5$ of the target case.

In order to ensure that all the arguments of the target case $tree(C_T)$ will be at the proposed solution, we define SP^* = $SP \cup < tree(C_T), \{C_T\}, 0 >$. Each tree of arguments in SP^* is by definition an argument framework $< Arg, Attack >$ with Arg being the arguments and $Attack$ the attack relation. The solution proposal will be defined as an argument framework Sol_{AF} = $< Arg_S, Attack_S >$ together with a preference function between arguments. Let us now define the solution's argument framework Sol_{AF} = $< Arg_S, Attack_S >$

Let $\{< Arg_0, Attack_0 >, ..., < Arg_n, Attack_n >\}$ be the set of all trees of arguments of SP^*. Then Sol_{AF} = $< \bigcup_{i \in [0,...,n]} Arg_i, \ \bigcup_{i \in [0,...,n]} Attack_i >$

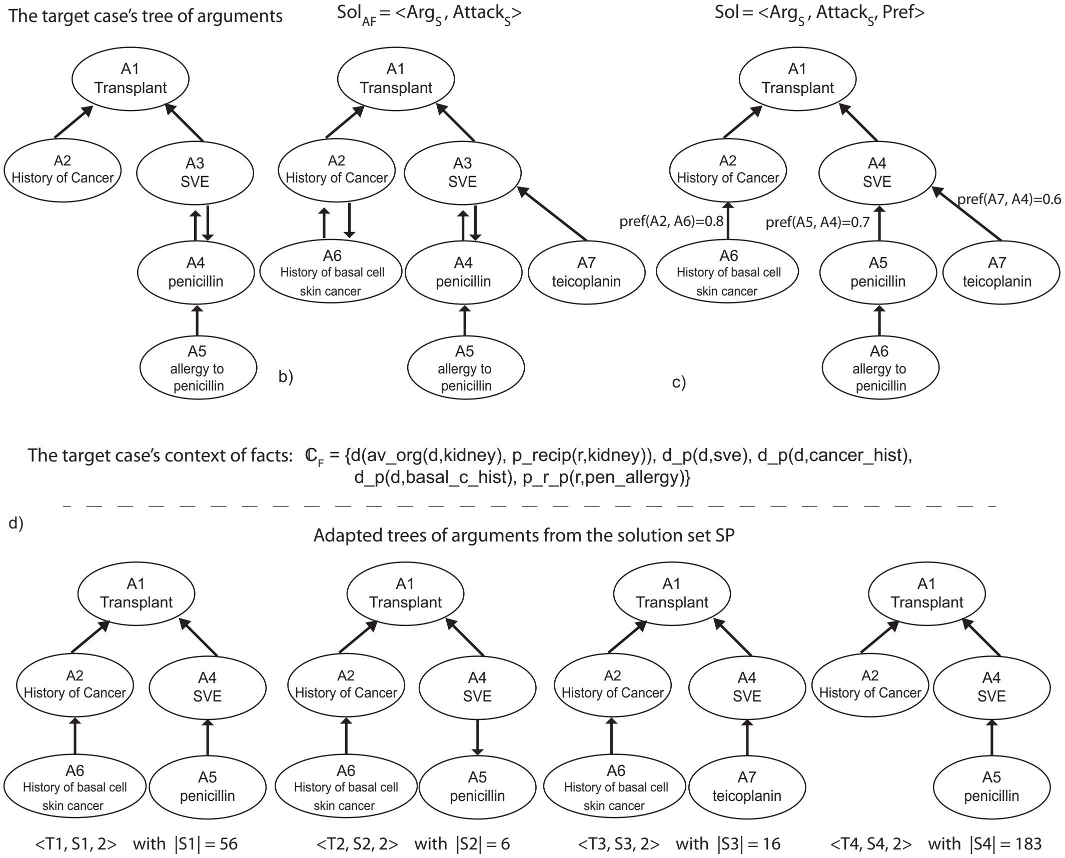

The target case's tree of arguments
A1 Transplant
A2 History of Cancer
A3 SVE
A4 penicillin
A5 allergy to penicillin
b)
Sol_AF = <Arg_S, Attack_S>
A1 Transplant
A2 History of Cancer
A3 SVE
A6 History of basal cell skin cancer
A4 penicillin
A7 teicoplanin
A5 allergy to penicillin
Sol = <Arg_S, Attack_S, Pref>
A1 Transplant
A2 History of Cancer
A4 SVE
pref(A2, A6)=0.8
pref(A5, A4)=0.7
pref(A7, A4)=0.6
A6 History of basal cell skin cancer
A5 penicillin
A7 teicoplanin
A6 allergy to penicillin
c)
The target case's context of facts: ℂ_F = {d(av_org(d,kidney), p_recip(r,kidney)), d_p(d,sve), d_p(d,cancer_hist), d_p(d,basal_c_hist), p_r_p(r,pen_allergy)}
d)
Adapted trees of arguments from the solution set SP
A1 Transplant
A2 History of Cancer
A4 SVE
A6 History of basal cell skin cancer
A5 penicillin
<T1, S1, 2> with |S1| = 56
A1 Transplant
A2 History of Cancer
A4 SVE
A6 History of basal cell skin cancer
A5 penicillin
<T2, S2, 2> with |S2| = 6
A1 Transplant
A2 History of Cancer
A4 SVE
A6 History of basal cell skin cancer
A7 teicoplanin
<T3, S3, 2> with |S3| = 16
A1 Transplant
A2 History of Cancer
A4 SVE
A5 penicillin
<T4, S4, 2> with |S4| = 183

Figure 9.6: Construction of the proposed solution for a target problem.

Note that all arguments in the target case $tree(C_T)$ are in Arg_S and potentially Arg_S may contain additional arguments. Figure 9.6 depicts the framework Sol_{AF} for the target case in the transplant scenario. In the depicted example the target case refers to a kidney transplant of a donor with a streptococcus viridans endocarditis and a history of basal cell skin cancer. The PAs have concluded the deliberation deeming the organ as non viable for transplantation believing on the one hand that the history of cancer was a contraindication and that the infection caused by the bacteria could not be mitigated because the recipient is allergic to the proposed treatment. From the retrieved cases depicted in figure 9.6d as grouped in SP, two additional arguments are proposed. The additional argument are contained in Sol_{AF} (see figure 9.6b). These are argument $A6$ indicating that for this particular case of cancer the disease should not be transmitted, and argument $A7$, proposing an alternative prophylactic treatment to prevent the bacterial infection. The final step is to assign a preference relation between the mutually attacking arguments:

Definition 9.10 *Let* $<\mathbb{T}, S, F >\in SP$ *then, for any two arguments* $A1$ *and* $A2$ *of* Sol_{AF} *we can define the function:*

$$ES_{arg}(\mathbb{T}, A1, A2) = \begin{cases} ES(\mathbb{T}, S, F) & \textit{if } A1 \textit{ asymmetrically attacks } A2\textit{, in } \mathbb{T} \\ 0 & \textit{Otherwise} \end{cases}$$

Note that, in particular, if either $A1$ or $A2$ are not in $\mathbb{T}$, then $ES_{arg}(\mathbb{T}, A1, A2)$ = 0.

Definition 9.11 *Let* Sol_{AF} = $< Args_S, Attacks_S >$, $(A1, A2) \in Attacks_S$ *and* $\{\mathbb{T}_1, ..., \mathbb{T}_n\}$ *the set of all trees of arguments of* SP*, then we define the preference function between arguments as:*

$$pref(A1, A2) = \Sigma_{i\in[1,..,n]} ES_arg(\mathbb{T}_i, A1, A2) - \Sigma_{i\in[1,..,n]} ES_arg(\mathbb{T}_i, A1, A2)$$

Where Σ *is an application specific aggregation function from* $[0, 1]^n$ *into* $[0, 1]$*. (e.g.* Max*) Hence,* $pref(A1, A2) \in [-1, 1]$

The closer $pref(A1, A2)$ is to 1 (respectively to -1), the more evidence there is that $A1$ should be deemed preferred to $A2$ (resp. $A2$ preferred $A1$). Whereas, if $pref(A1, A2)$ is close to 0 there is not enough evidence so as to deem one argument as preferred to the other.

To conclude, a solution to the target case C_T is a tree of arguments associated with the preference function between the attacking arguments, that is:

$$Sol = < Args_S, Attacks_S, \{< A1, A2, pref(A1, A2) > \ \mid \ (A1, A2) \in Attacks_S\} >$$

Figure 9.6c illustrates the solution for the target case including the additional arguments and the preference between arguments. This tuple is returned by the CBRc as the solution to the evaluation of the target case C_T, and it is integrated into the argument evaluation as described in §8.

Typically, to a case description there is attached a notion of success or failure of the applied solution [19]. This enables to select those successful solutions and prevent those which have failed. In our case, the success or failure of a proposed action is embedded in the cases' associated tree of arguments. In particular, the success and failure of a cases proposed solution is given by the dialectical status of the argument representing the decision. If a proposed solution turned out to be unsuccessful, shifting the direction of the attack relation may be enough to capture the actual outcome. This we discuss in the following subsection.

9.5.3 Revise

Once a solution is proposed, it must be tested in the real world. Firstly, the solution Sol is integrated into the ongoing deliberation, both in the form of a preference relation between arguments and possibly with the submission of additional arguments. The sole deliberation process my further alter the case description (*i.e.* the tree of arguments and the context of facts). Eventually, the deliberation concludes with a decision to either perform the proposed action or not. If the decision is not to perform the action, then the tree of arguments $\mathbb{T}$ will suffer no more changes and will be retained in the Case Base with $F = 0$. Otherwise, if the action is eventually performed, the PAs responsible for the action enactment will have to update $\mathbb{T}$ to account for the actual outcome. This is done, following the same rules of the deliberation dialogue defined in §5, however in this occasion the PAs' endorsements are conclusive. This is because the PAs are explaining what the outcome really was. Depending on the stage of the action performance that the $\mathbb{T}$ is revised the F will take the appropriate value, and eventually the case will be included in the case base memory.

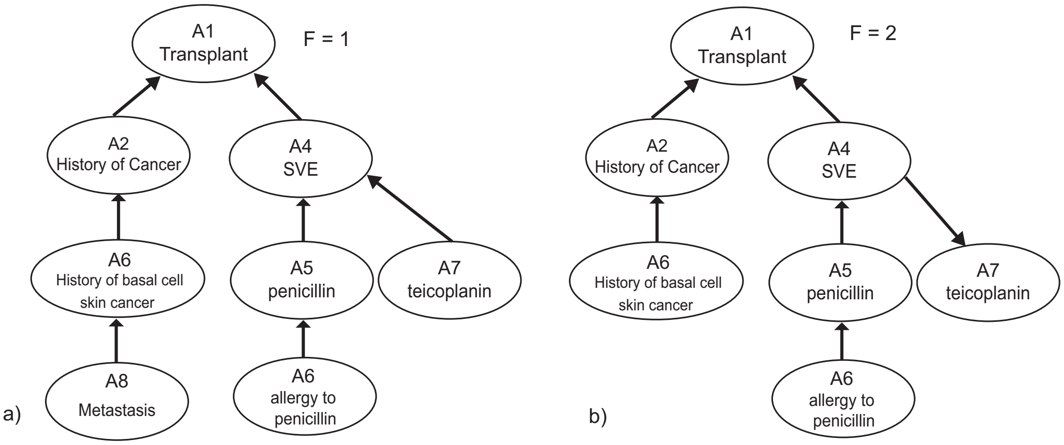


Figure 9.7: Two possible outcomes of the intended transplant. a) Metastasis was found on the donor at phase F=1, b) Teicoplanin was not effective in preventing the bacterial infection on the recipient.

If we assume the solution proposed in figure 9.6c was accepted by the PA, and so the transplant was deemed safe, the transplant unit responsible for the potential recipient would eventually proceed to extract the kidneys from the donor. Figure 9.7 depicts two negative outcomes of the transplant process. In the first case, depicted in 9.7a the transplant unit discovered during the extraction phase that the donor had metastasis. This substantially increases the risk of cancer transmission to the recipient and thus, the transplant is aborted.

In the second case, depicted in 9.7b the transplant unit has implanted the kidney into the recipient but was unable to prevent the bacterial infection to the recipient. Another possibility is, of course, that the transplant had a positive outcome and was successful, requiring no edition of the tree of arguments. In any of the three cases, the revised target case would then be retained by the CBRc. In the first case with $F = 1$ whereas in the two other cases the phase would be set to $F = 2$.

9.5.4 Retain

The goal of this process is to store the resolved case in the memory so that it can be reused to solve future similar problems. Let us suppose now that the target case C_T is has been solved and revised. The *retain* process is rather simple, if the tuple $< p_S(tree(C_T)), S >$ is in the case base memory, C_T is added to S. Otherwise, the tuple $< p_S(tree(C_T)), \{C_T\} >$ is added to the memory.

While the reuse of previous cases to resolve target deliberations is the main functionality intended for the CBRc, in addition, it is also to help decide, on an evidential basis, whether a submitted argument properly instantiates the argument schema it uses. This we describe in the following section.

9.6 Argument Validation

The argument validation process defined by *ProCLAIM* is intended to prevent the inclusion of spurious arguments into the deliberation. To perform this validation process the MA references the three knowledge resources: DCK, AEM and CBRc. A submitted argument $A1$ is accepted if it complies with the guidelines and regulations. That is, it has to be validated by the DCK. Exceptionally, if the argument is not validated by the DCK it may still be accepted if the PA that submitted the argument is trustworthy (validated by the AEM) and/or if similar arguments have been used in the past, that is, validated by the CBRc.

Definition 9.12 *Let $\mathbb{T}$ be the deliberation's tree of arguments. Let $A1$ the submitted argument. Let $\mathbb{T}_A1$ the path of arguments connecting argument $A1$ with the root argument of $\mathbb{T}$. Now, given two threshold $K \geq 0$ and $V > 0$ then:*

Argument $A1$ is validated by the CBRc if and only if $|Val| > v$ for $Val = \{c \mid c \in CB, \mathbb{T}_{A1} \preceq_{D_k} tree(c)\}$.

That is, if arguments similar to $A1$ have been used sufficiently many times in the past in similar context, then the CBRc will validate the argument.

9.7 Discussion

In this chapter we have presented *ProCLAIM*'s Case-Based Reasoning component. In so doing, we have shown how past stored deliberation can be reused in order to resolve similar target problems. A fundamental aspect for the definition of the CBRc is the fact that the trees of arguments produced in a *ProCLAIM* deliberation encode most of what is relevant for the decision making. When a deliberation concludes, the produced tree of arguments $\mathbb{T}$ contains all the factors deemed relevant by domain experts, guidelines and regulations. When $\mathbb{T}$ is evaluated by the CBRc it will also account for those factors deemed relevant in previous similar cases. Furthermore, $\mathbb{T}$ may embeds the actual outcome, whether successful or not, of the proposed action, if eventually performed. In particular, this allows to define the case description as the tuple $<\mathbb{T}, \mathbb{C}_F, F>$.

When introducing the case-base memory in §9.4 as well as the case-based reasoning cycle in §9.5 we have highlighted two important features of *ProCLAIM*'s specialised schemes that are also central for the CBRc: these are *1)* the specificity of the schemes in the ASR and *2)* that *relevant* facts and actions are introduced in a structured fashion, each singled out and introduced step by step. The former allows to efficiently identify potentially similar cases. The idea is that cases in which the same specialised schemes (reasoning patterns) were used, may be similar. Thus, by organising the case-base in terms of the argument schemes, a set of broadly similar cases can effectively be retrieved. The latter aspect of the schemes facilitates a more detailed comparison between cases on the basis of the similarity between the cases' introduced *relevant* facts.

We should also highlight the convenience of Dung's [80] argument graphs for the definition of the CBRc. The attack relation between arguments provides a natural way to encode the final decisions of the deliberation and the outcomes of the actions, if eventually performed. Thus, the solution is naturally embedded in $\mathbb{T}$, requiring no additional feature to represent the solution in the case representation. As shown in §9.5.2 this aspect is particularly useful for grouping the retrieved cases by their outcome in order to latter merge these groups into a single tree of arguments.

In §10.4 we present a prototype application intended as a proof of concept of the CBRc's formalisation, where the main purpose was to show how the entire reasoning cycle can be automated. As we discus in §10.4, the results obtained with this prototype were very positive, both in terms of the performance and acceptance by potential end users. In general terms the behaviour of the prototype was as expected, thus supporting the concepts developed in this chapter. One interesting outcome of this work is the realisation that the CBRc may be used not only for the evaluation of a given tree of arguments, but to actually generate a full tree of arguments from a given set of facts $\mathbb{C}_F$. This we intend to further explore in future work.

To the best of our knowledge, there is no other approach that allows for the reuse of past deliberation to help solve a target deliberation where: *1)* the retrieval of similar cases relevant for the solving the target case is a fully automated process; *2)* the adaptation of the retrieved cases and final solution proposal is a fully automated process; and *3)* storing resolved cases for future reuse, is also a fully automated process.

While we have made important progress both in the formalisation and implementation of the CBRc, there are a number of limitations that need to be addressed in future work. Three important aspects that need to be addressed are *1)* provide further indications for how to compute the evidential support associated to the solution proposal; *2)* better understand the relation between the distance between terms in an ontology and the similarity between arguments; and *3)* analyse the computational complexity of the proposed reasoning processes and study the CBRc's scalability.

In §6.2.4 we have introduced the notion of hypothetical arguments as arguments which instantiating facts are not in $\mathbb{C}_F$, nor their negation. In this chapter we have seen however that any additional argument submitted by the CBRc must be factual (not hypothetical). This is because, in the last step of the retrieval process the CBRc requires for all extracted cases to be *applicable* to the target case (see §9.5.1). In other words, any retrieved argument must be such that its instantiating facts are in $\mathbb{C}_F$ (or in distance k from another fact in $\mathbb{C}_F$, for $k > 0$). While hypothetical arguments are undoubtedly valuable, if we liberalise this last retrieval filter and allow for cases which are not applicable to the target case to be retrieved we might overpopulate the solution tree of arguments with hypothetical arguments. In future work we intend to find a compromise between these two options.

If we observe the above examples, we can note that each reasoning line is somewhat independent from the other. For example, the resolution of the contraindication given by the history of cancer and that of the streptococcus endocarditis are treated independently (see figure 9.6). However, on the case retrieval, the CBRc requires cases to share both reasoning lines. In future work we want to investigate to what extent each reasoning line can be treated as independent. This has an important impact in the retrieval process, as the number of relevant cases should increase substantially making a better use of the stored cases.

Chapter 10

Software Implementations

In this chapter we discuss four pieces of software that were used as a proof of concept to motivate the *ProCLAIM* model, as well as a way to better understand the model's limitations and aspects that need to be improved. Some of these aspects to improve are already addressed in this research. The first and more important result, regarding *ProCLAIM*'s implementation, is the large scale demonstrator for the medical transplant scenario which illustrates the model at work. This demonstrator was developed within the FP6-European Project Argumentation Service Platform with Integrated Components (ASPIC)[1] and presented in [11]. In §10.2 we discuss a web-based application intended to help developers construct an Argument Scheme Repository. In §10.3 we present a simple web application that allow end users (experts in the domain) navigate across the ASR in order to validate its content. In §10.4 we discuss the implementation of *ProCLAIM*'s Case-Based Reasoning component. Finally, in §10.5 we provide our conclusions on the basis of our experience based mainly, but not only, on these three implementations.

10.1 The Medical Large Scale Demonstrator

Back in 2004, in a context of emerging theoretical and practical works in the field of Argumentation Theory in AI, the EU Framework Programme 6 founded a collaboration project (ASPIC[2]) set out to first organise the diverse theoretical and practical works in the field, propose common theoretical grounds to then: *1)* advance the theory of Argumentation to a more principled and mature field upon which to build practical applications. *2)* Provide a set of standard software components based on the developed theory so as to demonstrate that argumentation is ready to extend from an abstract and formal level to a level where it has immediate and wide-ranging application potential. And, *3)* in order to test these components develop two large scale demonstrators. The main large scale demonstrator for this project implements the transplant scenario, using the *ProCLAIM* model.

The ASPIC's software components embody four standards for the argumentation-based technology: *inference*, *decision-making*, *dialogue* and *learning*. The two developed demon-

[1] http://www.argumentation.org

[2] Argumentation Services Platform with Integrated Components, FP6-IST-002307

strators were set to test different aspect of these components. The medical demonstrator was set to test the use of the *inference* and *dialogue* components in a complex and rich scenario as the human organ transplantation, so to evaluate the components' adaptability and scope of application. The second, business demonstrator, was set to test the use of the *inference*, *decision making* and *learning* components in a somewhat simpler scenario but using larger data sets. Broadly speaking, this demonstrator consisted of a central business that had to decide whether clients' requests to increase their credit should be accepted, where the rules for the credit acceptance or rejection were defeasible. In this scenario the argumentation processes themselves were simpler, however, the components were tested against large data sets (a large database of clients), which allowed testing performance characteristics of the components.

The medical large scale component was conceived as the extension of the CARREL institution introduced in §3.1.1. For this reason it was referred to as CARREL$^+$. This software application focuses only on the agents' argumentation. That is, in a mediated deliberation between a DA that offers an organ for transplantation and a RA that has to decide whether the organ is viable for a potential recipient she represents, as described in §3.1.3. While a detailed description of the four ASPIC components can be found in [31][3], we now introduce those components used in the medical demonstrator. In §10.1.2 we describe CARREL$^+$.

It is important to note that a number of changes were made to *ProCLAIM* formalisation, from the time this demonstrator was developed. The main improvement is in the formalisation of the argument schemes of the ASR. At the moment of implementing CARREL$^+$, schemes were developed in a somewhat *ad hoc* fashion. In particular, the exchanged arguments in CARREL$^+$ had to accommodate to the ASPIC components' formalisations which we outline in the following section. Another important difference is the dialogue protocol defined by the dialogue *component*. This dialogue game is based on Prakken's persuasion dialogue game defined in [176] for this reason we had to accommodate the transplant scenario into a persuasion setting, as opposed to a deliberation dialogue, as we have discussed in §2.3 and §5. An additional minor change is that the DCK (Domain Consented Knowledge) was called Guidelines Knowledge (GK). Later, in the discussion section, §10.1.3 we comment on these changes.

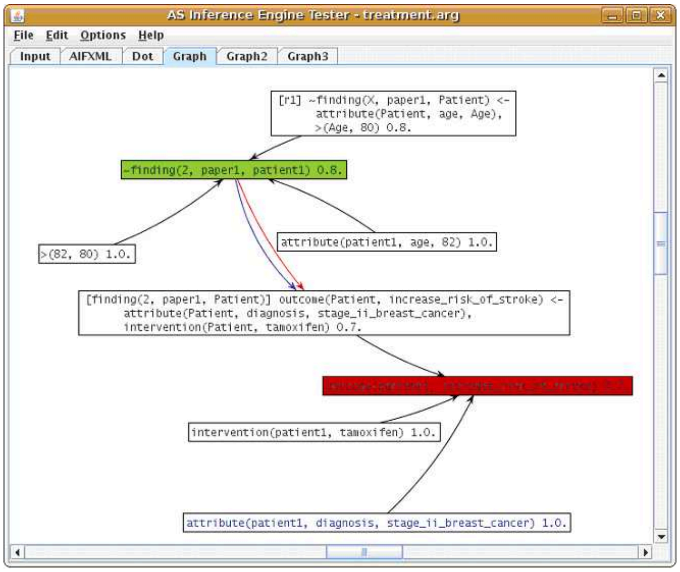
10.1.1 The ASPIC Argumentation Engine and Dialogue Manager

Inference using the Argumentation Engine

Inference is the core element of argumentation which in turn can support other important computational capabilities such as decision-making and dialogue. The ASPIC inference engine constructs and evaluates the status (justified or defeated) of arguments from a defeasible knowledge base for any claim that matches an input query. The ASPIC argumentation framework uses a defeasible model of argument-based inference, consisting of 4 steps:

1. *Argument Construction*: For any claim, arguments are organised into a tree-structure

[3] Also see http://aspic.cossac.org/components.html



```
[r1] ~finding(X, paper1, Patient) <-
     attribute(Patient, age, Age),
     >(Age, 80) 0.8.

~finding(2, paper1, patient1) 0.8.

>(82, 80) 1.0.

attribute(patient1, age, 82) 1.0.

[finding(2, paper1, Patient)] outcome(Patient, increase_risk_of_stroke) <-
     attribute(Patient, diagnosis, stage_ii_breast_cancer),
     intervention(Patient, tamoxifen) 0.7.

outcome(patient1, increase_risk_of_stroke) 0.7.

intervention(patient1, tamoxifen) 1.0.

attribute(patient1, diagnosis, stage_ii_breast_cancer) 1.0.
```

Figure 10.1: A proof network associated with the query, `outcome(patient,increase_risk_of_stroke)`

based on a knowledge base K of facts, a set S of strict rules of the form $\alpha_1, ..., \alpha_n \rightarrow \beta$, and a set R of defeasible rules of the form $\alpha_1, ..., \alpha_n \Rightarrow \beta$. The facts are expressed in a language consisting of first order literals and their negations. The ASPIC argumentation framework uses strict and defeasible modus ponens.

2. *Argument Valuation*: Arguments can be assigned a weight. No commitment is made to any particular valuation because the choice of the principle to be used will depend on the application domain.

3. *Argument Interaction*: Once arguments are constructed, binary conflict relations of attack and defeat are defined on this set of arguments. The definition of interactions between arguments depends on the specific logic that is being applied.

4. *Argument Status Evaluation*: Based on the graph of interacting arguments, Dung's *calculus of opposition* [80] is used to determine the status of arguments, specifically those that are justified.

At the end of this process, the Argumentation Engine allows the user to view a graphical visualisation of the proof argument network associated with the claim and examine the status, justified or defeated, for each argument. Though the visualisation was not as intuitive as it could be, it served as a proof of concept (see Figure 10.1). The engine also provides a machine readable version of the proof and results via AIFXML, an XML implementation of the Argument Interchange Format's abstract model [66].

The inferences derivable from a knowledge base can be characterised in terms of the claims of the justified arguments. The key contribution of the ASPIC model is that, in contrast with other approaches to argument-based inference, the model has been demonstrated to satisfy a number of quality postulates [60] which represent a set of minimal requirements that one would require to be satisfied by any rational model of argument based inference. In the ASPIC model, arguments have at least a claim and numeric support (a real number in the range (0,1]). The support is used to resolve attacks. An atomic argument can be developed from every atomic fact with the fact as the claim and the fact's Degree of Belief (DOB) as the argument's support. Further arguments can be developed through the application of rules. These tree arguments can be valuated with a choice of strategies: weakest link or last link. Weakest link valuation assigns the support for the main argument as the minimum support over all of its sub-arguments. Last link valuation assigns the degree of belief of the highest defeasible rule in the argument tree to the support of the main argument. If there are multiple highest level defeasible rules at the same level in the tree, then it assigns the support of the argument to be the minimum DOB of those rules. As in the underlying knowledge, arguments can be separated into strict and defeasible arguments where a strict argument has a support of 1.0 and a defeasible argument does not.

To define the acceptability of an argument we use defeat relations between all available arguments, and to do that we must define the conflict based attack relation between arguments. Three different types of attack are defined: rebutting, restricted rebutting and undercutting. Literals ∼`a 0.3.` and `a 0.5.` are both valid and their associated arguments rebut each other, where ∼ denotes the weak negation. Similarly, an argument formed from the fact `a.` and the rule `b<-a 0.9.` rebuts (and is rebutted) by an argument formed from the fact ∼`b 0.4.`. Strict arguments cannot be rebutted. Under restrictive rebutting, an argument whose top rule is strict cannot be rebutted by an argument whose top rule is defeasible.

Every rule in the inference engine knowledge base is automatically associated with a fact, which is the rule's name. The name forms a hidden premise for the rule. A knowledge engineer can explicitly provide that name when the rule is written and then undercut the rule by writing a fact or rule whose head is the contradiction of that name. If argument A undercuts argument B, then A claims that some rule in B is not applicable.

A = $((\sim rule\_name) \sim rule\_name)$; B = $((a, [rule\_name]a \Rightarrow b)b)$

Where $rule\_name$ is the name of the rule $a \Rightarrow b$. Note that argument A does not claim b is false, rather, that it cannot be derived from a.

Figure 10.1 shows a proof network associated with the query `outcome(patient, increase_risk_of_stroke)`. The diagram shows one argument (whose main claim is filled in red, defeated) that is developed for the query's claim but then undercut by another argument (whose main claim is filled in green, justified). The black arrows in the graph show how sub-arguments are linked together to form a single argument tree. The blue and red arrows in the graph indicate undercut and defeat relations between the two argument trees.

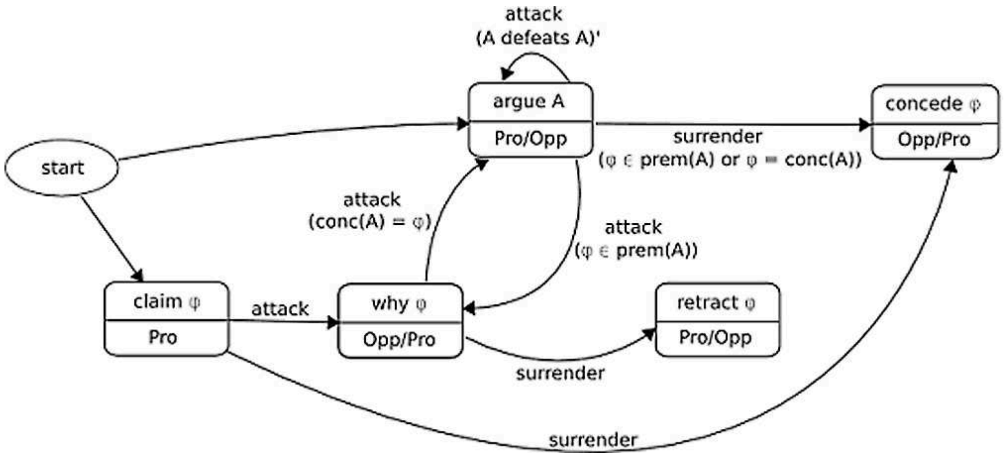


Figure 10.2: Persuasion protocol implemented in the Dialog Component, based on Prakken's persuasion dialogue game [176]

Dialogue Using the Dialogue Manager

The ASPIC Dialogue Manager provides a common API for interrogating the state and progress of an argumentation based dialogue. An argumentation based dialogue is characterised by moves whose content consists of claims or arguments that provide an explanation for a particular claim. The API is defined as a series of interfaces that must be implemented by a dialogue component implementation. The ASPIC implementation consists of two parts – a protocol that controls the enactment of the dialogue and a container that acts as an adapter between the protocol and the API. The role of the protocol is to control the initial conditions and the effect of a particular move on the state of a dialogue, *e.g.* the legal moves, the commitments and the status of the main claim. While it is envisaged that many protocols can be implemented in this framework at the time of CARREL$^+$'s implementation the available dialogue protocol was based Prakken's persuasion dialogue game [176], depicted in Figure 10.2.

The dialogue component expects moves that are constructed with the following attributes:

- agent
- move number
- locution
  - speech act (claim/why/argue/concede/retract)
  - content – a literal or an argument
- target move

In some argumentation based deliberation dialogues the target move is unneeded. In this case it can remain null.

The API consists of interfaces that enable consuming software to establish:

- the dialogue protocol
- the dialogue participants
- the dialogue topic
- the dialogue status (*initialising*, *in progress*, *terminated* or *abandoned*)
- the moves that have been previously made
- the commitments of each agent
- the legal moves
- the illegal moves and
- the status of the main claim (*undefeated* or *defeated*)

Each dialogue is represented by an instance of a dialogue object. When it is first created it has status *initialising*. In this status, no moves can be made and the protocol, participants and topic must be set. The protocol has built in constraints on the number and role of participants and the topic content. If the protocol/participants and topic are set, and the protocol validates these properties then the dialogue status can be progressed to *in progress*. After the dialogue has moved to *in progress*, the protocol, participants and topic cannot be changed and the dialogue will proceed as a series of moves (validated by the protocol, and rejected if they are invalid) until either there are no legal moves left to make or another condition, such as a participant leaving the dialogue, is met.

ASPIC has implemented three persuasion protocols. These protocols expect two participants, a proponent and an opponent, to build a dialogue about the acceptability of a particular claim (the topic). The claim must be represented as an ASPIC inference engine literal. The protocol defines a set of five speech acts (*claim*, *argue*, *why*, *retract* and *concede*). The relationship between locutions with these speech acts is shown in Figure 10.2.

It is anticipated that in these persuasion dialogues that the claim of the first move is the same as the dialogue topic and that the dialogue terminates when there are no moves left or when one of the agents leaves the dialogue.

The dialogue model of CARREL$^+$ extends the ASPIC dialogue in the following ways:

- It uses a scheme repository to further restrict and elaborate the possible attacks on an argument and thus the legal moves.

- It evaluates the defeat relations between argument's using the three knowledge resources: DCK, AEM. The evaluation made by the CBRc will be discussed in the final version on this Thesis.

In architectural terms, the CARREL Mediator agent exposes the same interface as the ASPIC dialogue component but must expose interfaces for managing the scheme repository and the evaluation component. The CARREL$^+$ evaluation components are seen as a separate entity to the interaction evaluation module within the inference engine that is consumed by the dialogue component.

10.1.2 CARREL$^+$

Figure 10.3, depicts CARREL$^+$ implementing the *ProCLAIM* model making use of the ASPIC components. The Argumentation Engine is use by the PAs for their reasoning (argument construction and evaluation) and the Dialogue Manager is used by the MA as a support for the dialogue mediation, in particular, to keep track of the exchange of arguments and evaluate the tree of arguments, once the relative strength of the arguments are given by the DCK and the AEM.

While the main purpose of CARREL$^+$ was to test the ASPIC components, another central aspect to evaluate was the *ProCLAIM* model, focusing on the mediated interaction between the PAs guided by the ASR. More in particular, we wanted to test whether in fact, both artificial and human agents can effectively participate in the argumentation. For that reason, in the following sections we outline the implementation of the three agents: MA, DA and RA, where the DA is a fully autonomous agent while RA is a human user assisted by a DSS (Decision Support System). This DSS is in fact an artificial agent that acts as a proxy between the user and the MA. Both DA and RA use the Argumentation Engine, the former to reason and take decisions, and the later to make suggestions to the end user. The MA has integrated the dialogue component to guide the deliberation, but it also references the ASR in order to narrows the scope of argumentation to what is relevant for the decision making. The MA also references the DCK and the AEM, both being shallow implementations. The former has listed the preferences between a set of possible arguments and the later assign to each participating agent a reputation degree (between 0 and 1). Later in this subsection we describe the agents' argument construction and we comment on the implementation of the ASR, DCK and AEM. In §10.1.3 we provide our conclusions regarding CARREL$^+$ implementation. Before we continue, just to note that all CARREL$^+$ agents are implemented in JADE[4] and thus interact in a JADE platform (see Figure 10.4).

The Mediator Agent

The MA is implemented as a semi-autonomous agent where only few tasks, that we now describe, are delegated to a human user via MA's GUI (Graphical User Interface).

Figure 10.5 shows the MA's GUI, where a user can see at each time the exchanged messages (top panel), the available legal moves (mid panel) and the argument-based dialogue moves (below panel). Note that moves in blue are justified arguments, moves in red are defeated arguments and moves in black are moves that are not arguments (*why*, *concede* or *retract* moves).

We can also see that the user can load a Guideline Knowledge (what we now call the DCK), an ASR and a knowledge base of agents' reputation. For consistency in the dialogue, this can only be done before the dialogue starts. Finally the user can terminate a running dialogue at anytime.

The MA has two main tasks: *1)* inform the participants of their available moves at each stage of the dialogue; and *2)* check whether the participants submitted arguments should be accepted.

[4] http://jade.tilab.com/

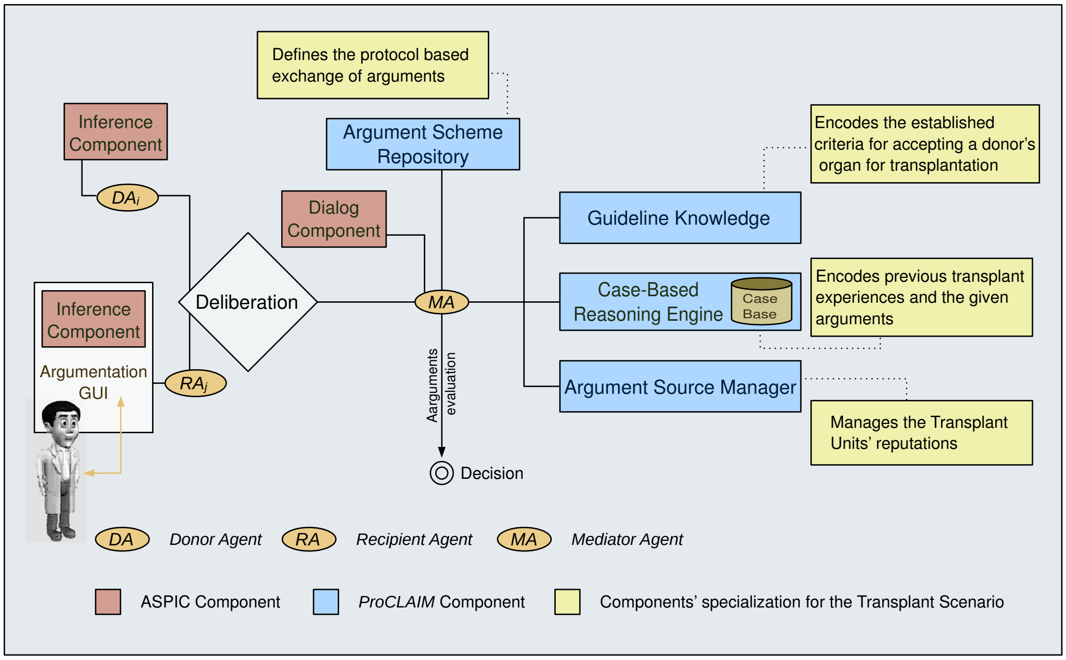

Defines the protocol based exchange of arguments
Inference Component
Argument Scheme Repository
Encodes the established criteria for accepting a donor's organ for transplantation
DA_i
Dialog Component
Guideline Knowledge
Inference Component
Deliberation
MA
Case-Based Reasoning Engine
Case Base
Encodes previous transplant experiences and the given arguments
Argumentation GUI
RA_j
Aarguments evaluation
Argument Source Manager
Manages the Transplant Units' reputations
Decision
DA
Donor Agent
RA
Recipient Agent
MA
Mediator Agent
ASPIC Component
ProCLAIM Component
Components' specialization for the Transplant Scenario

Figure 10.3: Argument-Based Framework for Deliberating Over the Viability of a Human Organ using the ASPIC components and the *ProCLAIM* model.

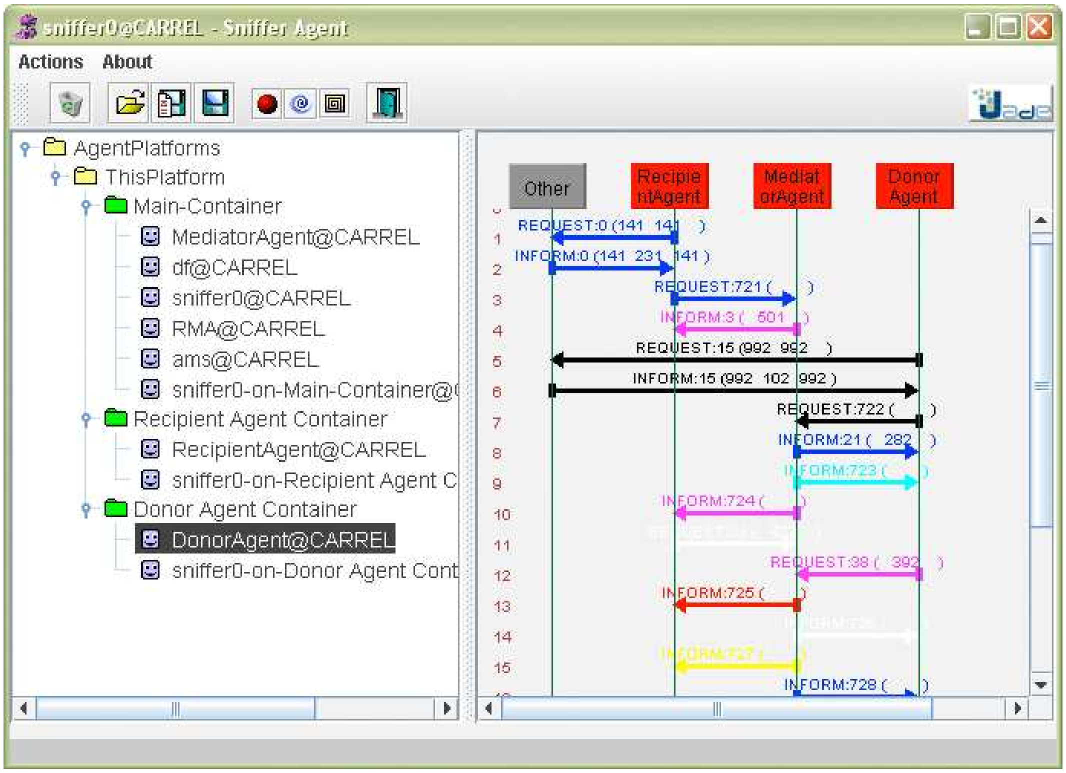
The first task involves querying the dialogue manager for the legal moves with respect to the persuasion protocol introduced in §10.1.1 and then reference the ASR to filter these moves to only those that are relevant for arguing over the organ viability. For example, while the dialogue component allows the proponent agent to start with any claim or argue locution (the legal moves are represented as `claim(X)` and `argue(since(X,Y))` being `X` and `Y` ungrounded variables representing the claim and support of the argument respectively) the ASR will filter these moves to only arguments that instantiate the argument scheme for the organ viability:

Claim $viable(Donor, Organ, Recipient)$

Support $[vs(Donor, Organ, Recipient)]\ viable(Donor, Organ, Recipient)$
$\Leftarrow available\_organ(Donor, Organ), potential\_recipient(Recipient, Organ).$
$available\_organ(Donor, Organ).\ potential\_recipient(Recipient, Organ).$

Where $vs(Donor, Organ, Recipient)$ is the name of the defeasible rule
$viable(Donor, Organ, Recipient) \Leftarrow available\_organ(Donor, Organ),$
$potential\_recipient(Recipient, Organ).$

That is, if there is an available organ for a potential recipient it is presumably viable. Note that argumentation in CARREL$^+$ is about organ viability. In the current *Pro-*



AgentPlatforms
ThisPlatform
Main-Container
MediatorAgent@CARREL
df@CARREL
sniffer0@CARREL
RMA@CARREL
ams@CARREL
sniffer0-on-Main-Container@
Recipient Agent Container
RecipientAgent@CARREL
sniffer0-on-Recipient Agent C
Donor Agent Container
DonorAgent@CARREL
sniffer0-on-Donor Agent Cont

1 REQUEST:0 (141 141)
2 INFORM:0 (141 231 141)
3 REQUEST:721 ( )
4 INFORM:3 (501)
5 REQUEST:15 (992 992)
6 INFORM:15 (992 102 992)
7 REQUEST:722 ( )
8 INFORM:21 (282)
9 INFORM:723 ( )
10 INFORM:724 ( )
11
12 REQUEST:38 (392)
13 INFORM:725 ( )
14
15 INFORM:727 ( )
INFORM:728 ( )

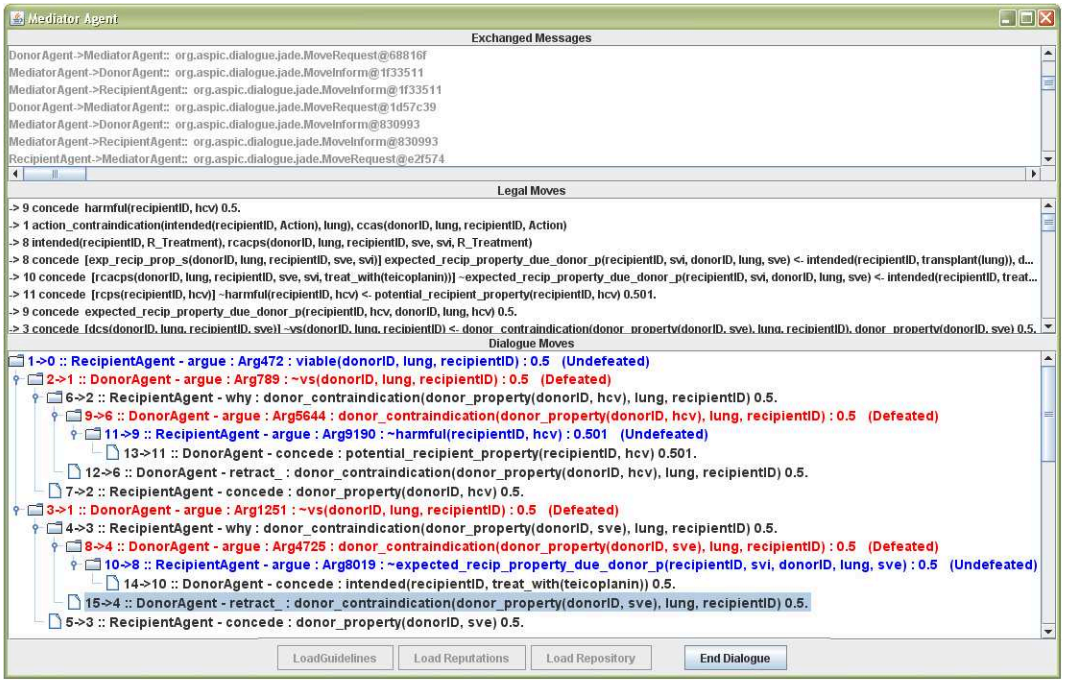
Figure 10.4: JADE's Remote Agent Management GUI showing the agent's messages interaction

CLAIM formalisation we have rephrased this into arguing over the safety of an organ transplantation.

The second task, checking whether the participants submitted arguments should be accepted, involves first checking that the submitted move is legal with respect to the dialogue's protocol and the ASR. If it is accepted and the move is not an argument, it is added to the dialogue graph. If the move is an argument further checking is required. The MA has to check that the argument is compliant with the DCK. If it is, the argument is accepted (added to the dialogue graph). If it is not accepted by the DCK, but the submitter of the argument has sufficiently good reputation, the argument may still be accepted, provided the user validates this decision, as illustrated in figure 10.6. In fact, the MA delegates most decisions to *ProCLAIM*'s components, which we describe in §10.1.2

The Donor Agent

The DA is conceived as an autonomous agent able to reason about the incoming dialogue moves, as well as construct and submit new moves in a logic programming language for which it uses the ASPIC inference component. The DA's KB is formed with a set of rules and facts expressed in a form compliant with the *inference* component.



```
DonorAgent->MediatorAgent::  org.aspic.dialogue.jade.MoveRequest@68816f
MediatorAgent->DonorAgent::  org.aspic.dialogue.jade.MoveInform@1f33511
MediatorAgent->RecipientAgent::  org.aspic.dialogue.jade.MoveInform@1f33511
DonorAgent->MediatorAgent::  org.aspic.dialogue.jade.MoveRequest@1d57c39
MediatorAgent->DonorAgent::  org.aspic.dialogue.jade.MoveInform@830993
MediatorAgent->RecipientAgent::  org.aspic.dialogue.jade.MoveInform@830993
RecipientAgent->MediatorAgent::  org.aspic.dialogue.jade.MoveRequest@e2f574
```

```
-> 9 concede  harmful(recipientID, hcv) 0.5.
-> 1 action_contraindication(intended(recipientID, Action), lung), ccas(donorID, lung, recipientID, Action)
-> 8 intended(recipientID, R_Treatment), rcacps(donorID, lung, recipientID, sve, svi, R_Treatment)
-> 8 concede  [exp_recip_prop_s(donorID, lung, recipientID, sve, svi)] expected_recip_property_due_donor_p(recipientID, svi, donorID, lung, sve) <- intended(recipientID, transplant(lung)), d...
-> 10 concede  [rcacps(donorID, lung, recipientID, sve, svi, treat_with(teicoplanin))] ~expected_recip_property_due_donor_p(recipientID, svi, donorID, lung, sve) <- intended(recipientID, treat...
-> 11 concede  [rcps(recipientID, hcv)] ~harmful(recipientID, hcv) <- potential_recipient_property(recipientID, hcv) 0.501.
-> 9 concede  expected_recip_property_due_donor_p(recipientID, hcv, donorID, lung, hcv) 0.5.
-> 3 concede  [dcs(donorID, lung, recipientID, sve)] ~vs(donorID, lung, recipientID) <- donor_contraindication(donor_property(donorID, sve), lung, recipientID), donor_property(donorID, sve) 0.5.
```

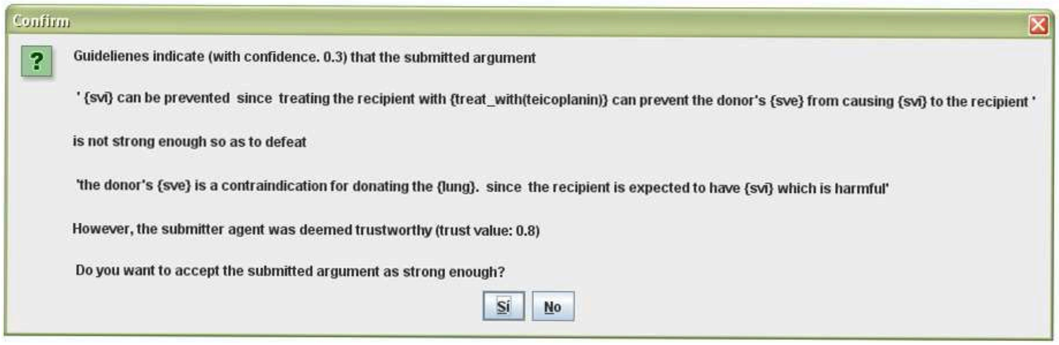
```
1->0 :: RecipientAgent - argue : Arg472 : viable(donorID, lung, recipientID) : 0.5   (Undefeated)
  2->1 :: DonorAgent - argue : Arg789 : ~vs(donorID, lung, recipientID) : 0.5   (Defeated)
    6->2 :: RecipientAgent - why : donor_contraindication(donor_property(donorID, hcv), lung, recipientID) 0.5.
      9->6 :: DonorAgent - argue : Arg5644 : donor_contraindication(donor_property(donorID, hcv), lung, recipientID) : 0.5   (Defeated)
        11->9 :: RecipientAgent - argue : Arg9190 : ~harmful(recipientID, hcv) : 0.501   (Undefeated)
          13->11 :: DonorAgent - concede : potential_recipient_property(recipientID, hcv) 0.501.
      12->6 :: DonorAgent - retract_ : donor_contraindication(donor_property(donorID, hcv), lung, recipientID) 0.5.
    7->2 :: RecipientAgent - concede : donor_property(donorID, hcv) 0.5.
  3->1 :: DonorAgent - argue : Arg1251 : ~vs(donorID, lung, recipientID) : 0.5   (Defeated)
    4->3 :: RecipientAgent - why : donor_contraindication(donor_property(donorID, sve), lung, recipientID) 0.5.
      8->4 :: DonorAgent - argue : Arg4725 : donor_contraindication(donor_property(donorID, sve), lung, recipientID) : 0.5   (Defeated)
        10->8 :: RecipientAgent - argue : Arg8019 : ~expected_recip_property_due_donor_p(recipientID, svi, donorID, lung, sve) : 0.5   (Undefeated)
          14->10 :: DonorAgent - concede : intended(recipientID, treat_with(teicoplanin)) 0.5.
      15->4 :: DonorAgent - retract_ : donor_contraindication(donor_property(donorID, sve), lung, recipientID) 0.5.
    5->3 :: RecipientAgent - concede : donor_property(donorID, sve) 0.5.
```

Figure 10.5: Mediator Agent's GUI

The DA has a GUI (see figure 10.7) where the user can view the agent's exchanged messages, the argument-based dialogue moves, the DA's intended moves, the DA's moves that where sent, accepted and rejected by the MA. The DA's strategy to propose any such moves is very simple. As depicted in Figure 10.14, when receiving an argument from the MA, the DA will test each element of the argument against her own KB using the inference component. If all elements are deemed justified, the DA concedes to the submitted argument. Otherwise she requests the MA for the schemes that attack this argument. Ones she receives these schemes she try to instantiates each element (fact or rule) of the scheme by matching the variables with her knowledge. If she succeeds and all elements of the scheme are deemed justified, the instantiated scheme is submitted. If no match is found, the DA will challenge the terms of the argument she does not agree with (by submitting a `why` locution).

The agent's GUI also allows the user to load at anytime an alternative knowledge base, or to load new knowledge (fact or rule) as depicted in figure 10.8. As we will see in the example introduced in section §10.1.2 new knowledge can change the agent's beliefs so as to, for example, *retract* from previously made dialogue moves.

The DA does not implement a method for deciding when to withdraw from dialogue. Hence this is done by the user (if no other agent has terminated the dialogue previously). With the sole purpose of controlling the demo's timing, it is the DA's user that decides when each of the agent's intended moves is to be submitted. This does not affect any relevant aspect of the DA's reasoning.



Guidelienes indicate (with confidence. 0.3) that the submitted argument

' {svi} can be prevented since treating the recipient with {treat_with(teicoplanin)} can prevent the donor's {sve} from causing {svi} to the recipient '

is not strong enough so as to defeat

'the donor's {sve} is a contraindication for donating the {lung}. since the recipient is expected to have {svi} which is harmful'

However, the submitter agent was deemed trustworthy (trust value: 0.8)

Do you want to accept the submitted argument as strong enough?

Figure 10.6: Dialogue window that the MA's user has to confirm for exceptionally accepting an argument not validated by the guidelines but which submitter has good reputation

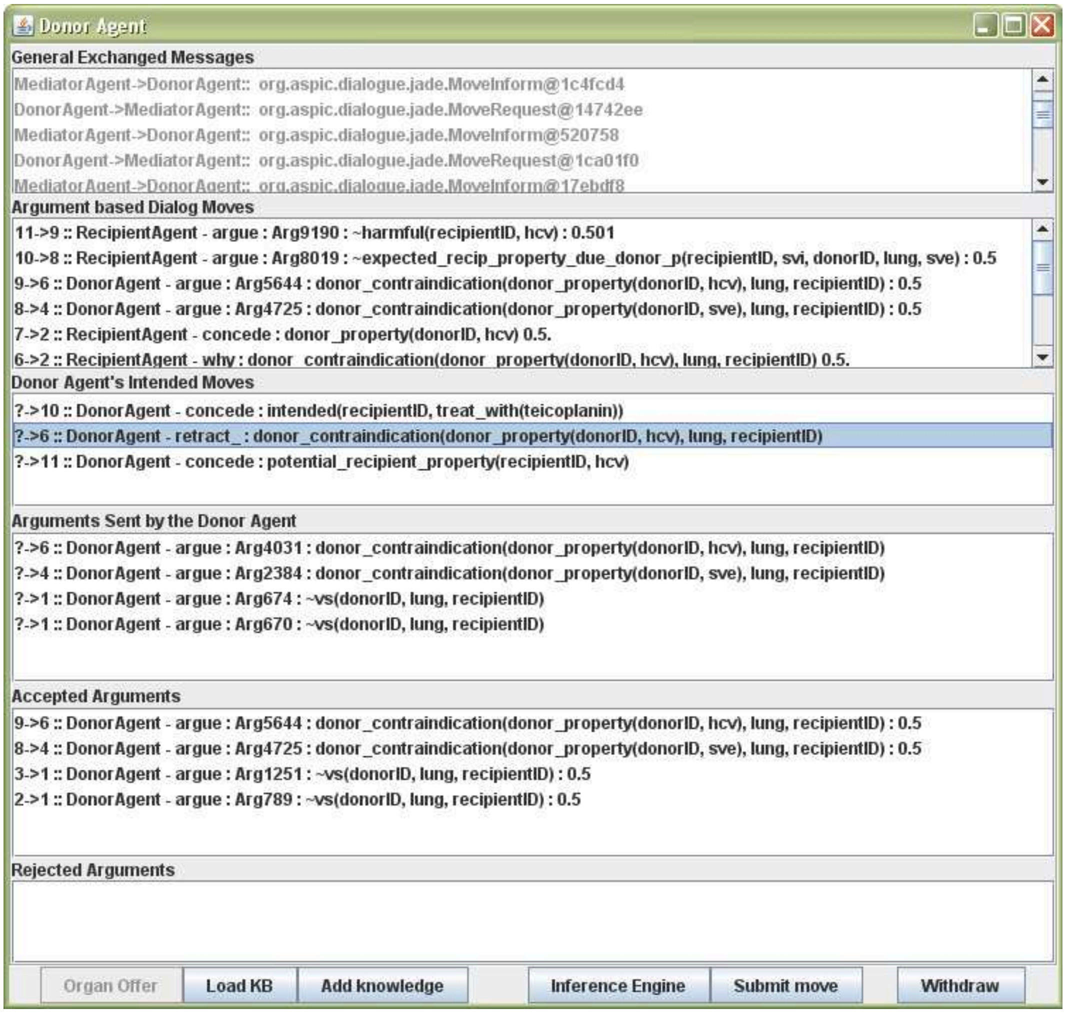
Finally, the DA's GUI allows offering a new organ introducing the donor and organ characteristics as displayed in Figure 10.9.

The Recipient Agent

As illustrated in figure 10.3, the RA is conceived as a user (medical doctor) interacting with a decision support system (DSS) that beneath has a proxy agent that communicates with the MA. The DSS assists the user in retrieving the submitted moves allowing the user to make only moves that are legal from the viewpoint of the protocol and the ASR (i.e., the legal moves facilitated by the MA). The DSS is integrated with the ASPIC inference component that enables the DSS recommend dialogue moves with which to reply to previously submitted move. While the DA construct arguments in logic programming language, the user does so in pseudo-natural language. In particular, as shown in figure 10.11, arguments are constructed by filling in the blanks in a template and the DSS can suggest possible instantiations compliant with the DSS's knowledge base.

For every selected argument-based move or suggested instantiation the user can call the inference component to display the argument tree (see figure 10.12) which allows the user to see the rational behind each DSS's recommendation. Such is also the case when a match between organ and potential recipient is found and the RA has to inform CARREL$^+$ on whether the he believes the offered organ is viable or not (see fig 10.13). The DSS recommends an assessment on the organ viability based on its knowledge base and the user can view the argument graph for such recommendation.

As in the case with the DA, the user can at anytime load an alternative knowledge base or add new knowledge. Finally the DSS allows the user to update information of potential recipients as shown in figure 10.9. Of course, any such changes (either in DA or in RA's DSS) may affect the course of the argumentation, as we illustrate in the running example later in this section.



General Exchanged Messages

```
MediatorAgent->DonorAgent::  org.aspic.dialogue.jade.MoveInform@1c4fcd4
DonorAgent->MediatorAgent::  org.aspic.dialogue.jade.MoveRequest@14742ee
MediatorAgent->DonorAgent::  org.aspic.dialogue.jade.MoveInform@520758
DonorAgent->MediatorAgent::  org.aspic.dialogue.jade.MoveRequest@1ca01f0
MediatorAgent->DonorAgent::  org.aspic.dialogue.jade.MoveInform@17ebdf8
```

Argument based Dialog Moves

```
11->9 :: RecipientAgent - argue : Arg9190 : ~harmful(recipientID, hcv) : 0.501
10->8 :: RecipientAgent - argue : Arg8019 : ~expected_recip_property_due_donor_p(recipientID, svi, donorID, lung, sve) : 0.5
9->6 :: DonorAgent - argue : Arg5644 : donor_contraindication(donor_property(donorID, hcv), lung, recipientID) : 0.5
8->4 :: DonorAgent - argue : Arg4725 : donor_contraindication(donor_property(donorID, sve), lung, recipientID) : 0.5
7->2 :: RecipientAgent - concede : donor_property(donorID, hcv) 0.5.
6->2 :: RecipientAgent - why : donor_contraindication(donor_property(donorID, hcv), lung, recipientID) 0.5.
```

Donor Agent's Intended Moves

```
?->10 :: DonorAgent - concede : intended(recipientID, treat_with(teicoplanin))
?->6 :: DonorAgent - retract_ : donor_contraindication(donor_property(donorID, hcv), lung, recipientID)
?->11 :: DonorAgent - concede : potential_recipient_property(recipientID, hcv)
```

Arguments Sent by the Donor Agent

```
?->6 :: DonorAgent - argue : Arg4031 : donor_contraindication(donor_property(donorID, hcv), lung, recipientID)
?->4 :: DonorAgent - argue : Arg2384 : donor_contraindication(donor_property(donorID, sve), lung, recipientID)
?->1 :: DonorAgent - argue : Arg674 : ~vs(donorID, lung, recipientID)
?->1 :: DonorAgent - argue : Arg670 : ~vs(donorID, lung, recipientID)
```

Accepted Arguments

```
9->6 :: DonorAgent - argue : Arg5644 : donor_contraindication(donor_property(donorID, hcv), lung, recipientID) : 0.5
8->4 :: DonorAgent - argue : Arg4725 : donor_contraindication(donor_property(donorID, sve), lung, recipientID) : 0.5
3->1 :: DonorAgent - argue : Arg1251 : ~vs(donorID, lung, recipientID) : 0.5
2->1 :: DonorAgent - argue : Arg789 : ~vs(donorID, lung, recipientID) : 0.5
```

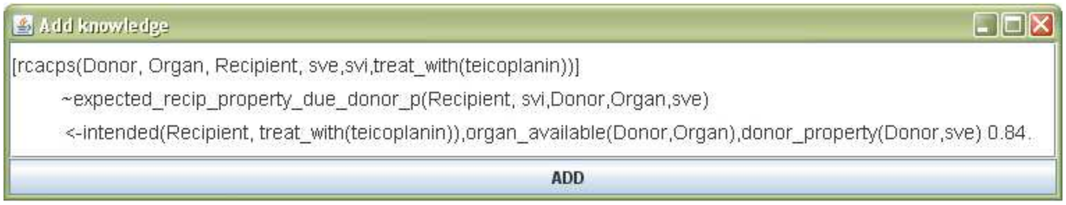
Rejected Arguments

Figure 10.7: Donor Agent's GUI

Argument Construction

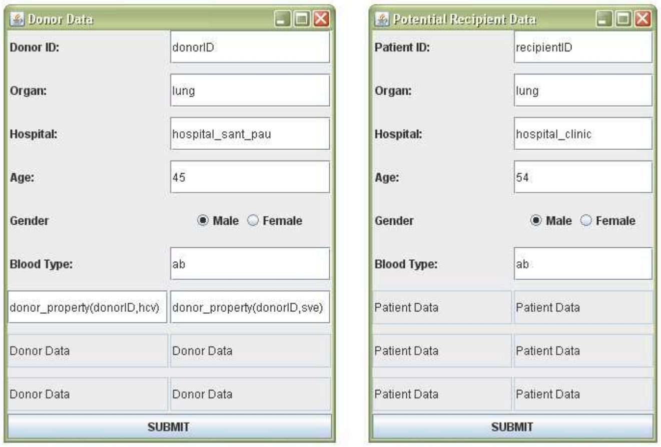
The agents reason using the ASPIC inference component via queries on their knowledge bases, where the result of a query is a list of arguments with their status of acceptability: defeated or justified.

The dialogue strategy of the PAs consist of attacking that which they disagree with and concede that with which they agree. The attack may take the form of an `argue` move if the PA is able to produce a legal argument that is accepted by the MA or a challenge, that is a `why` locution, as depicted in Figure 10.14. The difference in this process between the artificial and human agent is that while the artificial agent consults its knowledge base coded in PROLOG for the argument construction, the human agent, though assisted by a DSS, uses her own reasoning for that matter. As depicted in Figure 10.10 the GUI allows



```
[rcacps(Donor, Organ, Recipient, sve,svi,treat_with(teicoplanin))]
        ~expected_recip_property_due_donor_p(Recipient, svi,Donor,Organ,sve)
        <-intended(Recipient, treat_with(teicoplanin)),organ_available(Donor,Organ),donor_property(Donor,sve) 0.84.
```

Figure 10.8: Dialogue window to add new knowledge to a DA or a RA



Donor ID:	donorID
Organ:	lung
Hospital:	hospital_sant_pau
Age:	45
Gender	● Male ○ Female
Blood Type:	ab
donor_property(donorID,hcv)	donor_property(donorID,sve)
Donor Data	Donor Data
Donor Data	Donor Data

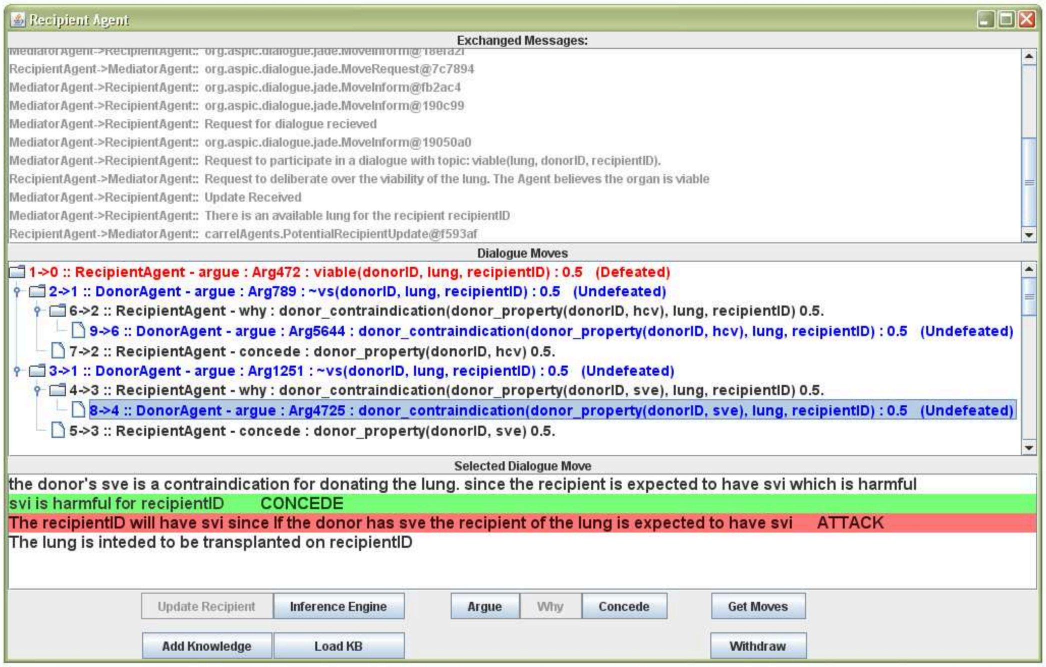
Patient ID:	recipientID
Organ:	lung
Hospital:	hospital_clinic
Age:	54
Gender	● Male ○ Female
Blood Type:	ab
Patient Data	Patient Data
Patient Data	Patient Data
Patient Data	Patient Data

Figure 10.9: Dialogue windows to make an organ offer (left) and to update CARREL$^+$ of a new potential recipient (right)

the user perform all the legal moves defined by the dialogue protocol and the DSS may suggest which moves to take.

To describe the argument construction process (both for the DA and for the RA's DSS) let us suppose that an argument is submitted by the interlocutor's agent. Our agent will then check whether it agrees or not with its content, *i.e.* its premises and claim. For each premise $\mathtt{p}_i$ for which the agent can construct a justified argument, the agent will attempt a $\mathtt{concede(p}_i\mathtt{)}$ move, if legal. However, if the agent disagrees with $\mathtt{p}_i$, namely, the result of querying $\mathtt{p}_i$ is a defeated argument, the agent will attempt to instantiate one of the legal moves that are arguments that attack $\mathtt{p}_i$ (with claim the $\sim\mathtt{p}_i$). For that purpose the agent will use the list of schemes facilitated by the MA. Suppose $\mathtt{L}$ is a legal argument move with claim $\sim\mathtt{p}_i$. If $\mathtt{L}$ is fully grounded and the agent agrees with all its premises and claim, then $\mathtt{L}$ is ready to be submitted as an argument. Otherwise, the agent must first instantiate the ungrounded variables. Suppose now that $\mathtt{pl}_1$,...,$\mathtt{pl}_{n-1}$ are the premises and rule names of $\mathtt{L}$. Let us denote $\sim\mathtt{p}_i$ as $\mathtt{pl}_n$.

Now, by querying $\mathtt{pl}_1\wedge...\wedge\mathtt{pl}_n$, the Argumentation Engine will instantiate the ungrounded variables which are binded across the premisses to produce justified or defeated arguments. If $\mathtt{pl}_1\wedge...\wedge\mathtt{pl}_n$ is grounded into a justified argument, the premises are rear-



```
MediatorAgent->RecipientAgent::  org.aspic.dialogue.jade.MoveInform@18efa2f
RecipientAgent->MediatorAgent::  org.aspic.dialogue.jade.MoveRequest@7c7894
MediatorAgent->RecipientAgent::  org.aspic.dialogue.jade.MoveInform@fb2ac4
MediatorAgent->RecipientAgent::  org.aspic.dialogue.jade.MoveInform@190c99
MediatorAgent->RecipientAgent::  Request for dialogue recieved
MediatorAgent->RecipientAgent::  org.aspic.dialogue.jade.MoveInform@19050a0
MediatorAgent->RecipientAgent::  Request to participate in a dialogue with topic: viable(lung, donorID, recipientID).
RecipientAgent->MediatorAgent::  Request to deliberate over the viability of the lung. The Agent believes the organ is viable
MediatorAgent->RecipientAgent::  Update Received
MediatorAgent->RecipientAgent::  There is an available lung for the recipient recipientID
RecipientAgent->MediatorAgent::  carrelAgents.PotentialRecipientUpdate@f593af
```

```
1->0 :: RecipientAgent - argue : Arg472 : viable(donorID, lung, recipientID) : 0.5   (Defeated)
  2->1 :: DonorAgent - argue : Arg789 : ~vs(donorID, lung, recipientID) : 0.5   (Undefeated)
    6->2 :: RecipientAgent - why : donor_contraindication(donor_property(donorID, hcv), lung, recipientID) 0.5.
      9->6 :: DonorAgent - argue : Arg5644 : donor_contraindication(donor_property(donorID, hcv), lung, recipientID) : 0.5   (Undefeated)
    7->2 :: RecipientAgent - concede : donor_property(donorID, hcv) 0.5.
  3->1 :: DonorAgent - argue : Arg1251 : ~vs(donorID, lung, recipientID) : 0.5   (Undefeated)
    4->3 :: RecipientAgent - why : donor_contraindication(donor_property(donorID, sve), lung, recipientID) 0.5.
      8->4 :: DonorAgent - argue : Arg4725 : donor_contraindication(donor_property(donorID, sve), lung, recipientID) : 0.5   (Undefeated)
    5->3 :: RecipientAgent - concede : donor_property(donorID, sve) 0.5.
```

```
the donor's sve is a contraindication for donating the lung. since the recipient is expected to have svi which is harmful
svi is harmful for recipientID       CONCEDE
The recipientID will have svi since If the donor has sve the recipient of the lung is expected to have svi     ATTACK
The lung is inteded to be transplanted on recipientID
```

Figure 10.10: Recipient Agent's GUI

ranged back into `L` and can be submitted by the agent via an `argue` move. As depicted in 10.14, if no legal move can effectively be instantiated or those instantiated are rejected by the MA, the agent will try to challenge the $\mathtt{p}_i$ via a `why` locution.

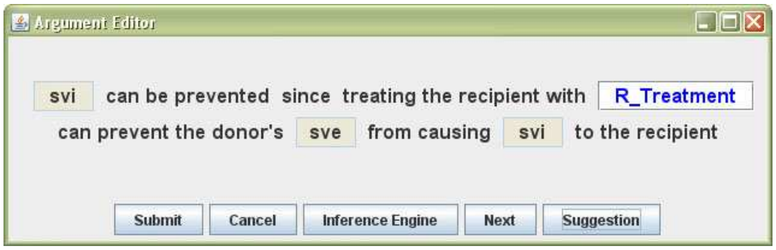
Later in this section we illustrate this process with a running example.

The ASR, DCK and AEM

The three *ProCLAIM* knowledge resources implemented in CARREL$^+$ are the ASR, the DCK and the AEM, the implementation of the CBRc will be discussed in the final version of this Thesis. The DCK and the AEM were coded in simple PROLOG script. The DCK has two main predicates: `is_valid(Argument)` and `is_strong_enough(Attacker ,Victim)`. Where `Argument`, `Attacker` and `Victim` are the rule name of the arguments' top rule. Simple examples of these are:

```
is_valid(dcs(Donor,Organ,Recipient,hcv)) 0.7.

is_strong_enough(rcps(Recipient, hcv),
ddts(Donor, Organ, Recipient, hcv, hcv)) 0.6.
```

So when a new argument with the top rule name's `rulen` is submitted, the MA will issue the query `is_valid(rulen)` to the Argumentation Engine and if returned a justified argument with DOB greater or equal to 0.5, the argument is accepted. A similar idea would



svi can be prevented since treating the recipient with R_Treatment can prevent the donor's sve from causing svi to the recipient

Figure 10.11: The RA constructs argument in pseudo-natural language. The user can either request the DSS for a suggestion on how to instantiated the variable `R_Treatment` or instantiate it himself.

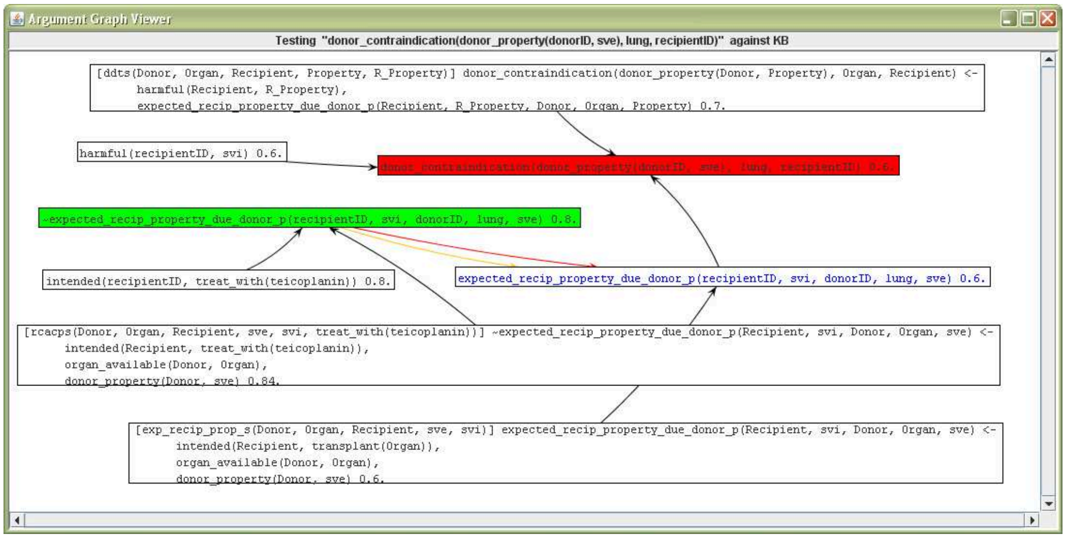
apply for deciding the preference relation between to attacking arguments.

A similar idea was followed for the AEM, where the special predicate is `trust(TransplantTeam)`. We also played with the predicate `trust(TransplantTeam, Argument)` so to incorporate the notion of trust with respect to a particular domain area. It is important to note that because the dialogue protocol used by the Dialogue Component was a persuasion protocol, PAs could not endorse any argument, *e.g.* those of the opponent. Namely, the trust in the agents regarding the problem at hand was limited to exceptional accepting the arguments they submitted (as we will illustrate in the running example, later in this section).

Finally, the ASR was developed as an independent component, implemented in java and PROLOG. The ASR stored about 20 schemes, coded in PROLOG code. As depicted in Figure 10.15 an argument scheme AS in the ASR has associated two lists of schemes: Attackers and Defenders. So that if AS is instantiated into an argument, schemes in Attackers could be used to attack that argument and schemes in Defenders can be used to defend that argument from a challenge, i.e. from a `why` locution. In particular, a premiss `p` of an argument instantiating AS can only be questioned if Defenders has a scheme that can be instantiated with claim `p`. Similarly, a that premiss can only be attacked, if Attackers has a scheme that can be instantiated with claim $\neg$`p`. For this reason the ASR interface included four main methods.

1. `getDefenders(top_rule)`
2. `getDefenders(top_rule,claim)`
3. `getAttackers(top_rule)`
4. `getAttackers(top_rule,claim)`

With any of these methods the first thing the ASR will try to do is, based on the top rule's name `top_rule`, identify the argument scheme it instantiates. If no match is found, the methods will inform of this mismatch (return `null`). Once the scheme is identified, the



Testing "donor_contraindication(donor_property(donorID, sve), lung, recipientID)" against KB

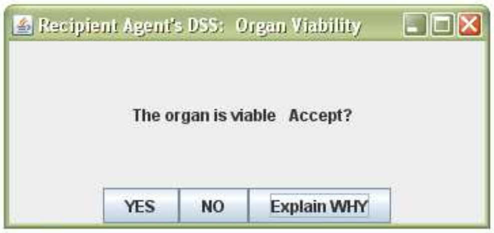
```
[ddts(Donor, Organ, Recipient, Property, R_Property)] donor_contraindication(donor_property(Donor, Property), Organ, Recipient) <-
      harmful(Recipient, R_Property),
      expected_recip_property_due_donor_p(Recipient, R_Property, Donor, Organ, Property) 0.7.

harmful(recipientID, svi) 0.6.

donor_contraindication(donor_property(donorID, sve), lung, recipientID) 0.6.

~expected_recip_property_due_donor_p(recipientID, svi, donorID, lung, sve) 0.8.

intended(recipientID, treat_with(teicoplanin)) 0.8.

expected_recip_property_due_donor_p(recipientID, svi, donorID, lung, sve) 0.6.

[rcacps(Donor, Organ, Recipient, sve, svi, treat_with(teicoplanin))] ~expected_recip_property_due_donor_p(Recipient, svi, Donor, Organ, sve) <-
      intended(Recipient, treat_with(teicoplanin)),
      organ_available(Donor, Organ),
      donor_property(Donor, sve) 0.84.

[exp_recip_prop_s(Donor, Organ, Recipient, sve, svi)] expected_recip_property_due_donor_p(Recipient, svi, Donor, Organ, sve) <-
      intended(Recipient, transplant(Organ)),
      organ_available(Donor, Organ),
      donor_property(Donor, sve) 0.6.
```

Figure 10.12: A proof network associated with the query, `donor_contraindication(donor_property(donorID,sve),lung, recipientID)`. The claim is defeated, namely, `sve` is not deemed as a contraindication because the infection on the recipient can be prevented.

`getDefenders` method will return a list of argument schemes that defend the argument with top rule name `top_rule`. These are the possible replies to a challenge move. If a `claim` is included in the method call, the ASR will only return the argument schemes with the claim matching `claim`. In both cases, the returned list of legal moves may be empty. An important aspect to be noted is that because `top_rule` is fully instantiated, *i.e.* all variables are grounded, through variable unification the returned argument schemes are partially, and sometimes completely, instantiated. We will see examples of this in the running example we now present. To conclude, the method `getAttackers` works in a similar way as `getDefenders` except that it returns the attackers rather than the defenders.

Running the Demonstrator

The demo starts with a DA and a RA informing the CARREL$^+$ of an available organ and of a potential recipient respectively (see Figure 10.9). As soon as there is a match between an offered organ and a patient in the waiting list both the appropriate DA and RA are informed of this match together with the patient's and donor's clinical data. On the basis of this data the agents inform CARREL$^+$ of whether they believe the organ is viable or not. The DA is an autonomous software agent, and thus it creates and submits its response automatically. The RA's DSS provides the user with an argument why it should deem the organ as viable or not. The user may accept or reject such suggestion. If both agents agree on the organ viability no argumentation takes place and the organ is deemed viable or non viable by the MA in accordance with the DA and RA assessment. If they disagree, the



Recipient Agent's DSS: Organ Viability

The organ is viable Accept?

Figure 10.13: Dialogue window asking for confirmation from the RA's user on the assessment of the offered organ's viability

MA uses the dialogue component to initiate a new dialogue instance where the agent that believes the organ to be viable undertakes the proponent's role and the other the opponent's. The dialogue protocol is set to be the persuasion protocol introduced in section 10.1.1 and the topic of the dialogue is set to: `viable(donorID,organ, recipientID)`, with `donorID` being the donor identification, `organ` is the offered organ and `recipientID` is the potential recipient's identification.

The first move in the dialogue is the argument for viability. This argument is submitted by the MA on behalf of the proponent agent. Subsequent moves will attack or defend this argument.

In the example shown in Figure 10.9, the DA offers a lung of a donor, `donorID`, whose cause of death was a streptococcus viridans endocarditis (`donor_property(donorID, sve)`) and had hepatitis C (`donor_property(donorID,hcv)`). Let as supposes as well that the offer arrives to a RA responsible for the patient `recipientID` (figure 10.9) that although not reported to CARREL$^+$, has also hepatitis C. Let us suppose as well that the DA believes the lung is not viable for `recipientID` because if the organ is transplanted to this patient he will result in having: *1)* an infection caused by the streptococcus viridans bacteria; and *2)* hepatitis C. Both being severe infections, bacterial and viral respectively. On the other hand, the RA's DSS suggests deeming the organ as viable because there are no known contraindications. The bacterial infection can be prevented by administrating teicomplanine to the recipient and patient `recipientID` already has hepatitis C, hence it cannot be deemed as a harmful consequence of the transplant.

Supposing the DSS persuades the user to deem the organ as viable and the appropriate message is sent to CARREL$^+$, a dialogue is initiate by the MA with RA being the proponent, DA the opponent and `viable(donorID, lung, recipientID)` the topic. The argument for viability of the lung (argument A1) is submitted by the MA on behalf of the RA and broadcasted to the participants.

Claim $viable(donorID, lung, recipientID)$

Support $[vs(Donor, Organ, Recipient)]viable(Donor, Organ, Recipient) \Leftarrow$
$available\_organ(Donor, Organ), potential\_recipient(Recipient, Organ).$
$available\_organ(donorID, lung).potential\_recipient(recipientID, lung).$

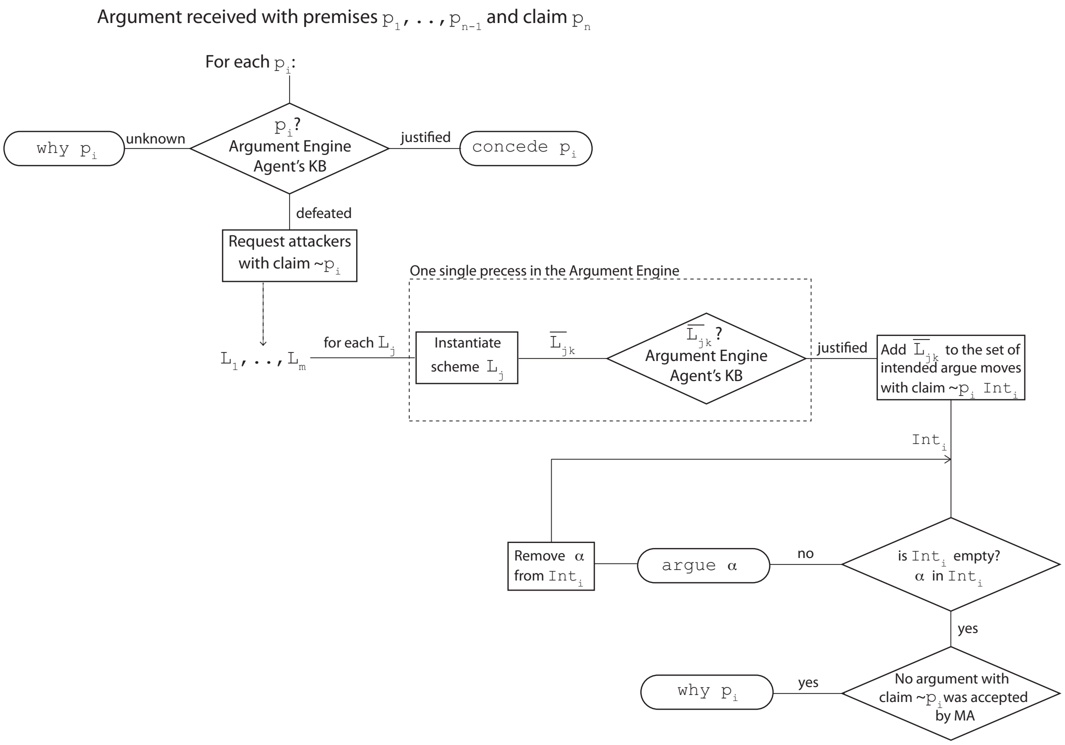
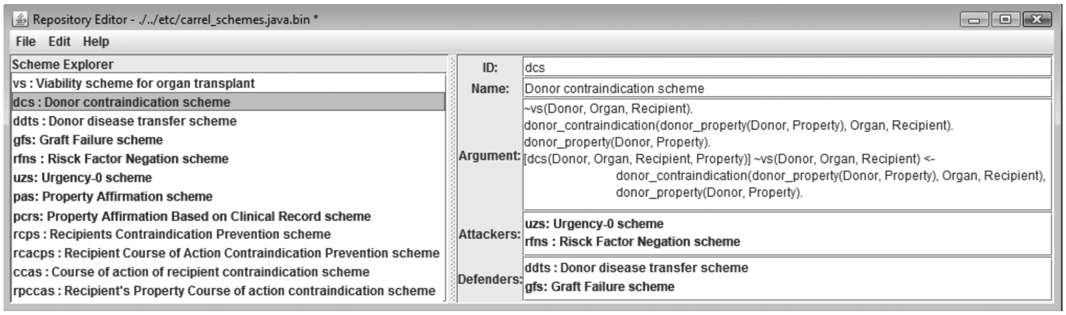

Argument received with premises $p_1, .., p_{n-1}$ and claim p_n
For each p_i:
why p_i
unknown
p_i? Argument Engine Agent's KB
justified
concede p_i
defeated
Request attackers with claim ~p_i
One single precess in the Argument Engine
$L_1, .., L_m$
for each L_j
Instantiate scheme L_j
$\overline{L_{jk}}$
$\overline{L_{jk}}$? Argument Engine Agent's KB
justified
Add $\overline{L_{jk}}$ to the set of intended argue moves with claim ~p_i Int_i
Int_i
Remove α from Int_i
argue α
no
is Int_i empty? α in Int_i
yes
why p_i
yes
No argument with claim ~p_i was accepted by MA

Figure 10.14: Process for the argument construction

Together with the submitted move the MA inform the participants of their available legal moves at this stage of the dialogue. From the view point of the dialogue protocol, each premise in the argument's support can be conceded, challenged with a why locution or attacked via an argument with claim the negation of one of the premises (i.e., $\sim vs(Donor, Organ, Recipient)$, $\sim available\_organ(Donor, Organ)$ or $\sim potential\_recipient(Recipient, Organ)$). Note that the content of the argument's support is not constraint. To focus the dialogue on the relevant issues to be addressed, rather than all the logically possible, the MA references the ASR. Thus, for example, the legal moves to reply to the argument for viability are reduced to only arguments that claim $\neg vs(Donor, Organ, Recipient)$ on the basis of, for example, a donor's contraindication, an organ dysfunction or a logistical contraindication. Also the opponent may concede to $vs(Donor, Organ, Recipient)$ in which the dialogue ends. Note that the opponent cannot attack the premise $available\_organ(Donor, Organ)$ or $potential\_recipient(Recipient, Organ)$ nor it can challenge any of the premises of the argument for viability. Any of these moves would be deemed illegal. In this way the dialogue is initially focused on whether or not there are any contraindications for transplanting the available organ.

Amongst the legal moves the MA sends to the opponent agent, in this case the DA, is the Donor Contraindication Scheme, represented in CARREL$^+$ as:



Scheme Explorer

vs : Viability scheme for organ transplant
dcs : Donor contraindication scheme
ddts : Donor disease transfer scheme
gfs: Graft Failure scheme
rfns : Risck Factor Negation scheme
uzs: Urgency-0 scheme
pas: Property Affirmation scheme
pcrs: Property Affirmation Based on Clinical Record scheme
rcps : Recipients Contraindication Prevention scheme
rcacps : Recipient Course of Action Contraindication Prevention scheme
ccas : Course of action of recipient contraindication scheme
rpccas : Recipient's Property Course of action contraindication scheme

ID: dcs

Name: Donor contraindication scheme

Argument:
```
~vs(Donor, Organ, Recipient).
donor_contraindication(donor_property(Donor, Property), Organ, Recipient).
donor_property(Donor, Property).
[dcs(Donor, Organ, Recipient, Property)] ~vs(Donor, Organ, Recipient) <-
                    donor_contraindication(donor_property(Donor, Property), Organ, Recipient),
                    donor_property(Donor, Property).
```

Attackers:
uzs: Urgency-0 scheme
rfns : Risck Factor Negation scheme

Defenders:
ddts : Donor disease transfer scheme
gfs: Graft Failure scheme

Figure 10.15: The GUI of the ASR component.

Claim $\neg vs(donorID, lung, recipientID)$

Support $[dcs(donorID, lung, recipientID, DonorProperty)]$
$\sim vs(donorID, lung, recipientID) \Leftarrow$
$donor\_contraindication(donorID, lung, recipientID, DonorProperty),$
$donor\_property(donorID, DonorProperty).$
$donor\_contraindication(donorID, lung, recipientID, DonorProperty).$
$donor\_property(donorID, DonorProperty).$

Note that the donor the recipient and the organ are know by the context (instantiated by the ASR) and what reminds to be instantiated by the DA is $DonorProperty$. Namely, identify a property on the donor that the DA believes to be a contraindication. In this case, the DA constructs and submits two arguments, A2 and A3, identifying hepatitis C (*hcv*) and streptococcus viridans endocarditis (*sve*) as contraindications for transplanting the lung.

The submitted arguments are then evaluated by the MA to check that they are legal with respect to the dialogue component protocol, the ASR and the Guidelines Knowledge. The latter allows MA to check that the argument instantiation is legal, in this case, that *hcv* and *sve* are in fact contraindications.

Supposing these two arguments are accepted by the MA and thus added to the dialogue graph, the MA will broadcasts the accepted moves together with the legal moves to the participants. At this stage the argument for viability is defeated and so if the dialogue terminates at this point the lung would be deemed non-viable. Hence, to defend the organ's viability the RA must defeat both arguments A2 and A3.

The RA may request for some evidence on the facts that the donor had *hcv* and *sve* by challenging premises $donor\_property(donorID, hcv)$ and $donor\_property(donorID, sve)$ of arguments A2 and A3 respectively via the `why` locution. Or, concede these premises relying on DA's information[5]. However, since RA does not agree with

[5] Any information `info` provided by the interlocutor `source` are added to the agents' knowledge base via the predicate `recieved_info(info,source)` the agent then *believes* `info` to be the case only if she trusts `source` regarding information of type `info`. *E.g.* the RA will typically trust the DA on information about the donor.

$donor\_contraindication(donorID, lung, recipientID, hcv)$ nor $donor\_contraindication(donorID, lung, recipientID, sve)$ he will try to attack such premises. Legal attacks on these premises are based on *1)* the potential recipient is in a highly precarious condition (risk of death in the following 24 hours) that can only be overcome with a lung transplant, hence hcv (sve rep.) cannot be deemed as a contraindication; *2)* hcv (resp. sve) is a risk factor of some condition X known to be a contraindication, but the donor does not have X.[6] Neither is the case, so the RA's DSS is unable to construct an attacking argument on either A2 or A3. Therefore, it suggests challenging the facts that *hcv* and *sve* are contraindication, effectively shifting the burden of proof back to DA. The user can ask the DSS why the challenge locution is suggested to which the DSS will display an argument attacking $donor\_contraindication(donorID, lung, recipientID, sve)$ (rep. *hcv*) as depicted in Figure 10.12.

Note that at any time the user may ignore the DSS's suggestions and submit any other dialogue move. Nonetheless, the DSS allows the user submitting only moves that are legal from the viewpoint of the dialogue's protocol and the ASR. That is, those moves facilitated by the MA.

Supposing the RA finally concedes to the facts that the donor had *hcv* and *sve* but challenges the fact that these are contraindications, the DA will have to justify why these conditions are contraindication.

Amongst the schemes the DA can instantiate to defend A2 as well as A3 is the Donor Disease Transmit Scheme:

Claim

$donor\_contraindication(donor\_property(donorID, sve), lung, recipientID)$

Support $[ddts(donorID, lung, recipientID, sve, R_Property)]$

$donor\_contraindication(donor\_property(donorID, sve), lung, recipientID)$
$\Leftarrow harmful(recipientID, R\_Property), expected\_recip\_property\_due\_donor\_p$
$(recipientID, R\_Property, donorID, lung, sve).$
$[exp\_recip\_prop\_s(donorID, lung, recipientID, sve, R_Property)]$
$expected\_recip\_property\_due\_donor\_p(recipientID, R\_Property, donorID,$
$lung, sve) \Leftarrow intended(recipientID, transplant(lung)),$
$donor\_property(donorID, sve).\ intended(recipientID, transplant(lung)).$
$donor_property(donorID, sve).\ harmful(recipientID, R_Property).$

The DA can thus instantiate this scheme to indicate that *sve* (rep. *hcv*) is a contraindication because the recipient will result having *svi*: streptococcus viridans infection (resp. *hcv*) which is harmful.

Supposing these two arguments (A4 and A5 respectively) are submitted by DA and accepted by MA, the RA will have to defeat both A4 and A5 in order to defend the organ's viability. In this case the RA's DSS suggest to attack both arguments indicating in the

[6] An example use of this argument would be to attack the fact that smoking history is a contraindication when the donor does not have chronical obstructive pulmonary disease.

first case that given that $recipientID$ already has *hcv*, resulting in having *hcv* cannot be deemed as a harmful consequence of the transplant. In the latter case, the DSS suggests attack A5 (see figure 10.10) by arguing that the infection on the recipient can be prevented by administrating teicoplanine to the recipient (see figure 10.11).

Let us suppose that both arguments (A6 and A7 respectively) are submitted by RA and, that while A6 is validated by MA, MA derives from Guidelines that there is not enough confidence on the use of teicoplanine for the prevention of svi so as to accept argument A7. Let us also suppose that RA has good reputation and thus his assessment that the suggested antibiotic can effectively prevent the recipient's infection may be accepted (see Figure 10.6). If the MA finally accepts both arguments as legal, the status of acceptability of the initial argument would be accepted, i.e., the organ would be deemed viable for $recipientID$.

In this example, when DA is informed of the submission of A6 it updates its knowledge base adding the fact that $recipientID$ already has hcv (it trusts the RA assessment on that matters), in consequence it concedes to the fact that the recipient has *hcv* and retracts from its previous claim that hcv is a contraindication (see figure 10.7). In general, at any time new knowledge can be added to an agent's knowledge base that may result in changes in the agent's believes. The dialogue can accommodate to such changes by allowing participants to retract and in general to backtrack to reply to any previously submitted dialogue moves. Another example of this is if we add via the DA's interface new knowledge (see figure 10.8) indicating that teicoplanine is an effective treatment to prevent *svi*, the DA will also retract from its claim that *sve* is a contraindication.

At any point during the dialogue the participant agents can withdraw, or the MA can terminate the dialogue and the resolution is given by the dialectical status of acceptability of the argument for viability, in this case, the argument is accepted and thus, if the dialogue terminates the MA will send both DA and RA a message informing them that the lung was deemed viable for $recipientID$.

10.1.3 Discussion

The results of the demonstrator were very good and it received excellent comments from the reviewers deeming it "*the most sophisticated argumentation-based application of that time*". While there were a number of obvious weaknesses that we now address, we believe that CARREL$^+$'s main strength resides in the principled way in which the problem was structured. That is, how by implementing *ProCLAIM* we could build a setting in which human and artificial agents can effectively argue about the viability of an organ for transplantation, with little requirements in the development of the artificial agent (the DA) nor from the DSS that assists the human agent in the argumentation.

Having served as a very good proof of concept for *ProCLAIM*, CARREL$^+$ has a number of limitations. These include the use of a persuasion protocol as opposed to a deliberation one, a shallow use of the DCK and of the AEM, the absence of an explicit ontology, it was not integrated with the CBRc and it provided a very basic GUI. However, from our point of view, the most important issue for this kind of application was the lack of a systematic procedure to build the ASR. While at that time we had up to 40 schemes (see [7]) and a preliminary validation process (see 10.3), schemes were in effect build following intuition.

While it was clear this was a limitation it become even clearer when we had to give indications on how to build the ASR to a research team that was not familiar with Argumentation, which is what happened in the development of the environmental scenario (see §11.2).

For this reason, we placed all our efforts in proposing a systematic way to build the ASR. This led to what we believe to be a central contribution of this Thesis. That is, the proposal of a reasoning patterns tailored for deliberating over the safety of an action, that it is used as the basis to develop the schemes in to a scenario-specific argument schemes and CQ which, as just illustrated in the running example, facilitates an effective argumentation among heterogeneous agents.

In addition to propose a systematic way to develop the ASR, the new reasoning patterns provide a number of important improvements for the deliberation. The first improvement is reducing the agents' interchanged arguments and so making the argumentation more efficient. This is because, in our current approach a submitted argument not only introduces the relevant factors (facts or actions) but also must also indicate why these factors are relevant (see §6). As a result, what in the introduced example involves the exchange of four argumentation moves: *1)* the organ is viable; *2)* it is not viable because X is a contraindication; *3)* why X is a contraindication; and *4)* X is a contraindication because of Y. The same example presented in 6, was addressed in only two arguments: *1)* the action is safe; and *2)* it is not safe because X will cause an undesirable side effect Y.

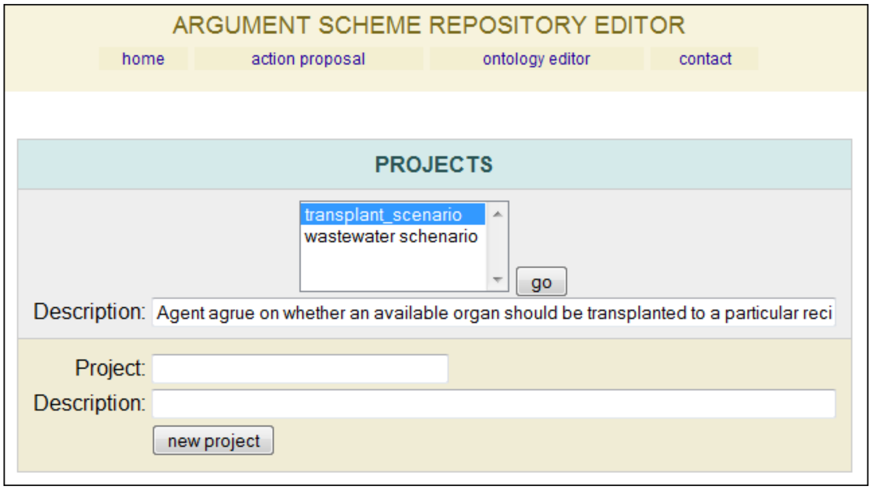
One of the objectives of the ASR is to enable heterogeneous agent to argue effectively. Where the heterogeneity may be in the form of human/artificial agents but also amongst artificial agents with different modes of reasoning. In other words, the ASR is used as a bridge to connect the diverse modes of reasoning. Of course, for this to make sense each agent must be able to translate her reasoning into the schemes in ASR. Namely, each agent must be able to effectively instantiate the schemes in the ASR and effectively understand those submitted by their interlocutors. Now, the problem with the ASR in CARREL$^+$ is that each scheme incorporates one or more rules, these rules are different from scheme to scheme and follow no predefined structure. Therefore, each agent must understand each and every rule present in the ASR. This results in a strong imposition on the artificial agents' implementation since any translation module intended to interface between the agent's internal mode of reasoning and the schemes in the ASR, will have to work on a cases by case. This not only imposes limitation on the agents' heterogeneity but it also makes non-trivial the updates on the ASR. We believe to have made important progress in addressing these issues by providing a set of twelve reasoning patterns upon which the ASR is developed. Thus, for agents to understand each other, or rather, effectively use the schemes in the ASR, would require only to have a shared ontology regarding $\mathbf{R}$, $\mathbf{A}$, $\mathbf{S}$ and $\mathbf{G}$, that is, the used facts, actions, effects and *undesirable* goals, and an understanding of a cause effect relation captured in twelve reasoning patterns introduced in §6.2.

Other improvements include the decoupling of the resolution of *what is the case* and the deliberation over the actions' safety. Firstly, this gives priority to the main question:–*Is the action safe in current circumstances?*-so that, for example, questioning the current circumstances (*i.e.* the facts in $\mathbb{C}_F$) is licensed only if this challenges the action's safety (at least in a local context). Secondly it allows one to address, in a relatively simple fashion, problems such as incomplete or uncertain information, at the time of constructing the arguments, when updating the new available information and when evaluating the arguments. Other improvements relate also to the dialogue game protocol, now defining a deliberation more in line with requirements of *ProCLAIM*.

10.2 Argument Scheme Editor

In this section we present the web application we develop for constructing ASRs, we tested it on the transplant and environmental scenario. This application was useful as a proof of concept to demonstrate how users with no prior knowledge of argumentation can be guided, step by step, in the construction of repository of schemes. We now describe the application's functionalities and user interaction and we later, in §10.2.1 we discuss its added value and limitations.

The Argument Scheme Editor is a web based application built on PHP[7] and MySQL[8] and is available at *http://www.lsi.upc.edu/∼tolchinsky/newASR/*.



PROJECTS

transplant_scenario
wastewater schenario

Description: Agent agrue on whether an available organ should be transplanted to a particular reci

Project:

Description:

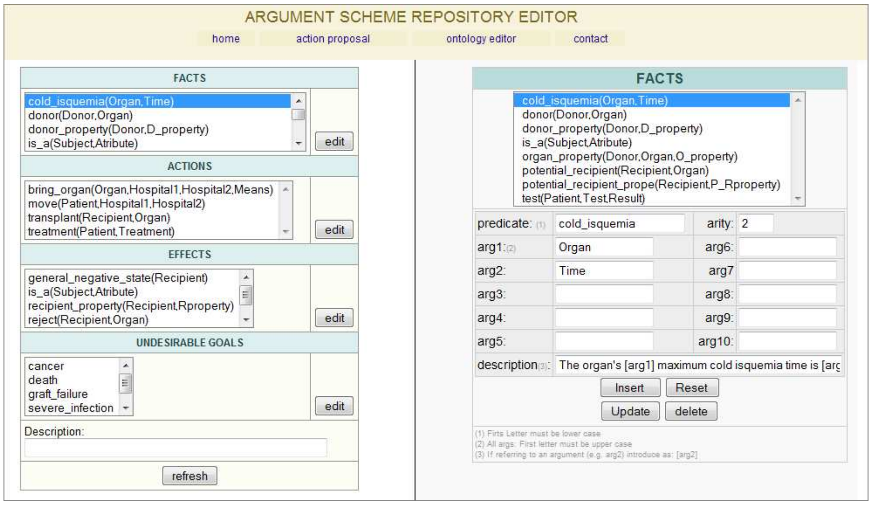
Figure 10.16: Argument Scheme Editor, entry page.

The first thing a user must to is create a project or enter into an existing one, as depicted in 10.16. If the project is new, the user should populate the projects' associated ontology, clicking on the *ontology editor* in the top menu. This is takes the user to page displaying the list of facts, actions side effects and undesirable goals associated with the selected project. In this page the user can populate the $\mathbf{R}$, $\mathbf{A}$, $\mathbf{S}$ and $\mathbf{G}$ as depicted in 10.17

Once the $\mathbf{R}$, $\mathbf{A}$, $\mathbf{S}$ and $\mathbf{G}$ are populated, the user may create a new action proposal. As depicted in Figure 10.18, the transplant project only has one action proposal (to transplant an organ).

[7] http://www.php.net/

[8] www.mysql.com/



FACTS

cold_isquemia(Organ,Time)
donor(Donor,Organ)
donor_property(Donor,D_property)
is_a(Subject,Atribute)

ACTIONS

bring_organ(Organ,Hospital1,Hospital2,Means)
move(Patient,Hospital1,Hospital2)
transplant(Recipient,Organ)
treatment(Patient,Treatment)

EFFECTS

general_negative_state(Recipient)
is_a(Subject,Atribute)
recipient_property(Recipient,Rproperty)
reject(Recipient,Organ)

UNDESIRABLE GOALS

cancer
death
graft_failure
severe_infection

Description:

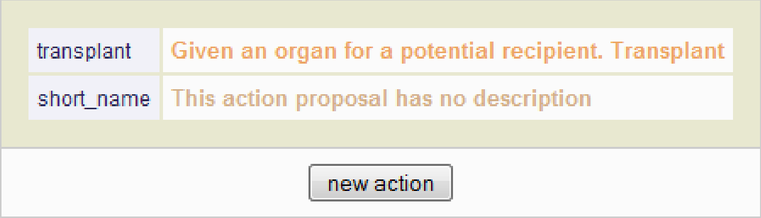
FACTS

cold_isquemia(Organ,Time)
donor(Donor,Organ)
donor_property(Donor,D_property)
is_a(Subject,Atribute)
organ_property(Donor,Organ,O_property)
potential_recipient(Recipient,Organ)
potential_recipient_prope(Recipient,P_Rproperty)
test(Patient,Test,Result)

predicate: (1)	cold_isquemia	arity:	2
arg1: (2)	Organ	arg6:	
arg2:	Time	arg7	
arg3:		arg8:	
arg4:		arg9:	
arg5:		arg10:	
description (3):	The organ's [arg1] maximum cold isquemia time is [arg		

(1) Firts Letter must be lower case
(2) All args: First letter must be upper case
(3) If referring to an argument (e.g. arg2) introduce as: [arg2]

Figure 10.17: Page to populate the **R** , **A** , **S** and **G** for the transplant scenario project



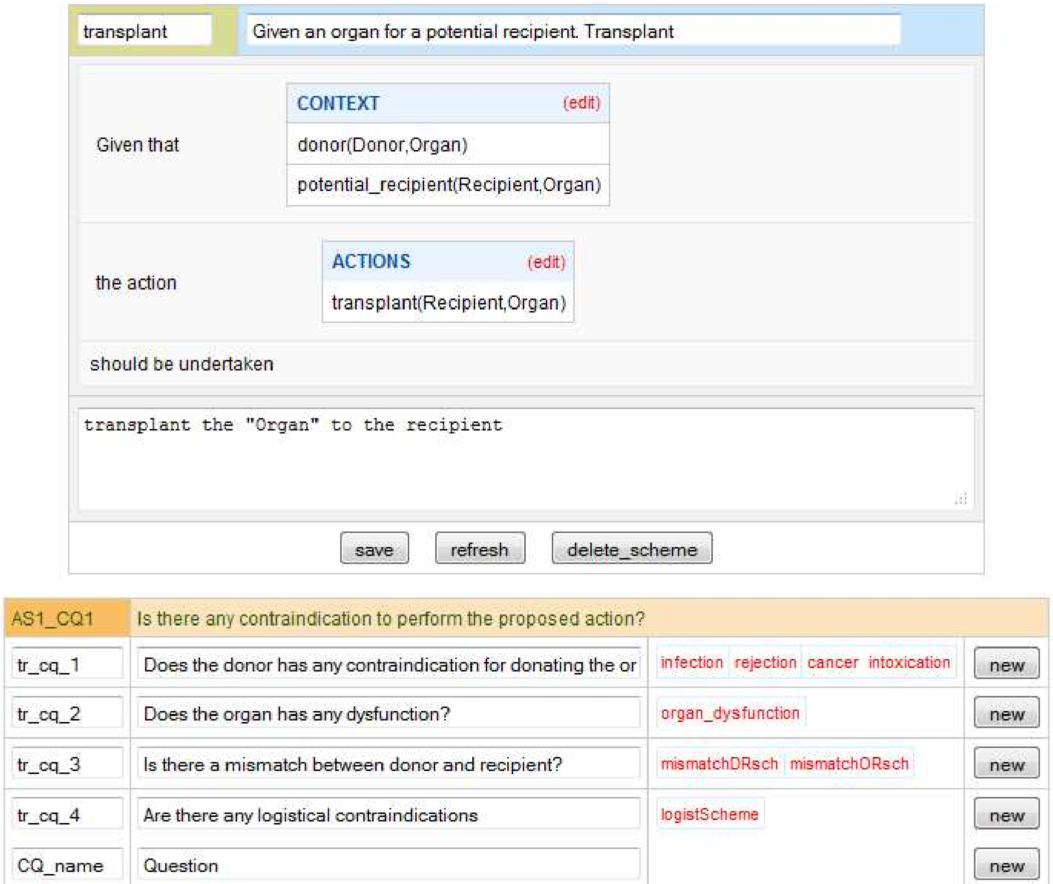
transplant	Given an organ for a potential recipient. Transplant
short_name	This action proposal has no description

Figure 10.18: Page for creating a new action proposal

Clicking on the action proposal one enters into an instantiation of a scheme AS1 (see 6.2.1). As shown in Figure 10.19, the user hast to fill in an identifier of the scheme (*e.g. transplant*) and a description. Next, the user has to fill in a list of facts and actions in order to specialise scheme AS1. By clicking on the *(edit)* link alongside the *CONTEXT* or *ACTIONS* lists the user will be given a list of facts (rep. of actions) from the project's ontology. The user must also provide a NL version of the scheme, where variables are placed inside quotes. The idea is that when constructing the ASR, the three representation are constructed in parallel: an underlying programming language (*i.e.* PROLOG), the artificial agents' communication language (as defined in the interaction protocol in §5) and finally a NL representation that will facilitate human user interaction.

Below the scheme we can see a list of specialised CQs of AS1_CQ1, the user can add as many specialised CQs as she requires, and to each specialised CQ, the user can link specialised schemes. In this case, these would be schemes specialising AS2 (see 6.2.1).

If we were at this point to add a new scheme linked to AS1_CQ1, we would be led to the page shown in Figure 10.20. In this page we can see that the context of facts and actions is already given. The user can start specialising the scheme by first selecting an undesirable



transplant

Given an organ for a potential recipient. Transplant

Given that

CONTEXT (edit)

donor(Donor,Organ)

potential_recipient(Recipient,Organ)

the action

ACTIONS (edit)

transplant(Recipient,Organ)

should be undertaken

```
transplant the "Organ" to the recipient
```

AS1_CQ1	Is there any contraindication to perform the proposed action?		
tr_cq_1	Does the donor has any contraindication for donating the or	infection rejection cancer intoxication	new
tr_cq_2	Does the organ has any dysfunction?	organ_dysfunction	new
tr_cq_3	Is there a mismatch between donor and recipient?	mismatchDRsch mismatchORsch	new
tr_cq_4	Are there any logistical contraindications	logistScheme	new
CQ_name	Question		new

Figure 10.19: Page for specialising scheme AS1 with its associated CQs.

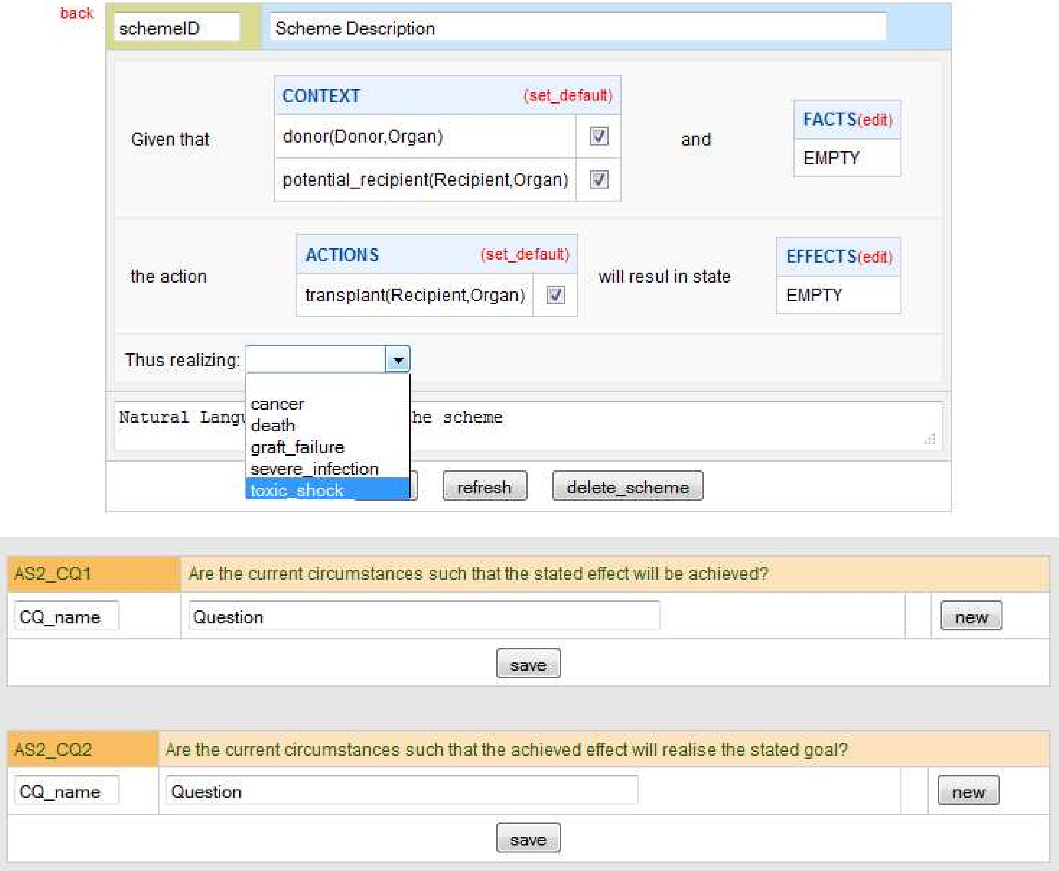
goal from the dropdown list, to then specialise the *EFFECT* list by clicking on the *(edit)* link, to then select in a similar fashion the list of facts (type of facts, since the predicates are ungrounded) because of which the undesirable side effect will be realised. Figure 10.21 shows the panel in which the user selects the tailors the list of facts, froma dropdown list.

In addition, the user will have to address each of the CQs associated to scheme which may in turn link to other schemes.

In this way, the user is guided via schemes an CQs to explore all the relevant lines of reasoning that need to be specialised for the particular application at hand.

10.2.1 Discussion

Throughout this paper, and in particular in this chapter, we have shown the benefits of having a structured set of scenario-specific schemes and CQs. However, as discussed in §10.1.3, a lack of any clear systematic procedure for developing the ASR results in a strong limitation on the applicability of the proposed model. For this reason we have developed (in §6) a



schemeID | Scheme Description

Given that

CONTEXT (set_default)	
donor(Donor,Organ)	☑
potential_recipient(Recipient,Organ)	☑

and

FACTS(edit)
EMPTY

the action

ACTIONS (set_default)	
transplant(Recipient,Organ)	☑

will resul in state

EFFECTS(edit)
EMPTY

Thus realizing:

cancer
death
graft_failure
severe_infection
toxic_shock

Natural Langu... he scheme

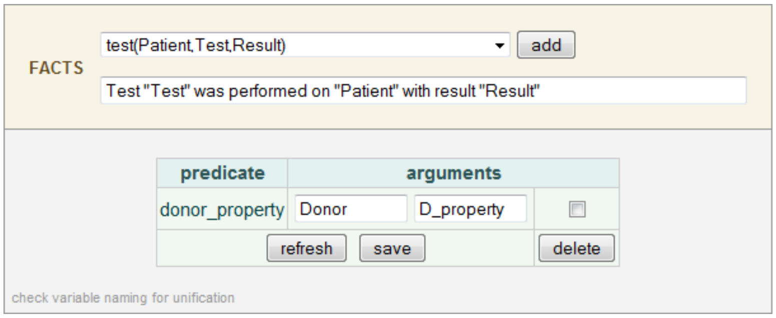
AS2_CQ1 Are the current circumstances such that the stated effect will be achieved?

CQ_name | Question

AS2_CQ2 Are the current circumstances such that the achieved effect will realise the stated goal?

CQ_name | Question

Figure 10.20: Page for specialising scheme AS2 with its associated CQs.



FACTS test(Patient,Test,Result)

Test "Test" was performed on "Patient" with result "Result"

predicate	arguments		
donor_property	Donor	D_property	□

check variable naming for unification

Figure 10.21: Page for specialising scheme AS2 with its associated CQs.

set of reasoning patters of an intermediate level of abstraction, that is, while tailored for reasoning over safety critical actions they remain scenario-independent. Along these reasoning patters, encoded in schemes and CQs, we proposed (in §7) a procedure for their further specialisation in order to build the ASR. This procedure is embodied in the web application we have just presented. This web application thus serves as a good proof of concept for the ideas we want to convey regarding the systematisation of the ASR's construction.

As shown in this section, once the sets of facts, actions, side effects and undesirable goals are filled, the construction of scenario-specific schemes is rather simple, by using the web application and following the procedure described in §7. Moreover, the construction of these schemes requires no formal knowledge of Argumentation Theory[9]. As we will discuss in §11.2, our immediate experience with this application is that not only the process of constructing the scenario-specific schemes was much clearer, but also the scope and limitation of *ProCLAIM* deliberations' were better understood by the developers of the environmental scenario. For example, that *ProCLAIM* was not intended for planning. This led them to reformulate the scenario to better match the model's intended use.

As a proof of concept, this web application has numerous aspects to improve. Firstly, we should allow connecting the ontology to existing ontologies developed for the domain of application; a visual graph should allow visualising and browsing the whole ASR, this will not only enable a fast navigation and access to different parts of the ASR but presumably, will help developers have a better understanding of the ASR structure. Users' should be assisted in the creation of ID's for the each new scheme and CQs. We should allow for plugins to convert the content of the ASR into different formats (*e.g.* PROLOG[10] code). More importantly, however, is to allow for automatic mechanisms for updating the schemes' structure and relations. While the schemes and CQs defined in §6 are in a mature state, there can be minor changes in their structure and relations as we further continue our research. Currently, this web application cannot adapt to these changes, in fact, it is currently more similar to the formulation we presented in [219] than that of §6 presented also in [16]. Another aspect to address in future work is to allow for the reuse of the specialised schemes. This will allow for repeated cycles in the reasoning patterns. Future versions of the Argument Scheme Editor should incorporate these changes.

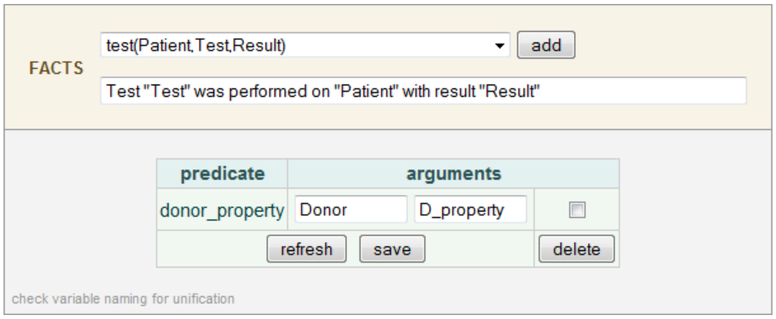
10.3 ASR Browser

As part of the software we developed to assess *ProCLAIM*'s applicability was another web application intended to allow users to navigate across the ASR. This is a very simple web page in which we placed some of the schemes of the medical ASR. This application was intended for the transplant professionals to read and interact with the proposed reasoning patterns, specific of the transplant scenario, and provide us with feedback regarding the schemes validity. In Figure 10.22 we can see a snapshot of the web application, the user is presented with a scheme, its associated CQs and links to the schemes in turn associated

[9] Of course they are required to have some basic knowledge of logics and be able to understand the used notation.

[10] http://en.wikipedia.org/wiki/prolog

with the CQs. All these, presented in Natural Language.



test(Patient,Test,Result)

Test "Test" was performed on "Patient" with result "Result"

predicate	arguments		
donor_property	Donor	D_property	□

check variable naming for unification

Figure 10.22: Argument Scheme Browser

In Figure 10.22 we can see that a CQ can be replied with a *Yes* or a *No*. This is in line with CARREL$^+$'s ASR, we presented in §10.1.2, where schemes negating the CQs represent the possible direct attacks while, schemes at the *Yes* part, represent can be used as defenders from a challenge. Note in particular, that when no scheme is in the *Yes* part, the CQs cannot be used as a challenge.

This web application has a few very simple functionalities: when hovering with the mouse over a link to a scheme, a description of that scheme (reasoning pattern) is presented to the user. There are a number of blank spaces in the scheme, the user can type in the value she desires and the blanks containing the same variable will change accordingly. Conversely, the user can also click on the *example* button to get different possible instantiations.

Prior to this application we devoted a few session to convey our ideas to the transplant professionals[11] and we gave them a few notions regarding basic Argumentation Theory. During these initial sessions we found an important shift in involvement, engagement and understanding of our proposed ideas once we began describing the specialised schemes and CQs. The proposals and corrections by the medical doctors were much more valuable and meaningful. The Argument Scheme Browser was used within this elicitation process and it helped showing that *reasoning patterns* or *argument schemes* were not necessarily an abstract logical construct to which end users should adapt, but rather than these constructs could accommodate to their needs and language. Also, it helped showing that the links between one scheme and another were not conceptual but explicit. In the sense that by clicking on a CQ on the browser it displayed a scheme which encoded a meaningful reasoning patten for the transplant professionals.

In a more advance stage of our research, we explored the use of Atkinson's scheme for action proposal discussed in §6.1, in order to guide users in their deliberation, both in the transplant and the environmental scenario . However, our experience were two folds, some of the experts lost interest in our study[12] while others start to guess possible instan-

[11] Most of the medical research was made in collaboration with a Transplant Coordinator with his team of the Transplant Unit of Sant Pau Hospital in Barcelona, Spain.

[12] We speculated that one important factor for their distraction was due to the absence of keywords related to

tiations. That is, even though they were familiar with the actual reasoning patterns, there were confusions regarding the instantiations of the sets R, A, S G and V. But more importantly, they were guessing rather than asserting. However, when again presented with the specialised schemes, the engagement increased and there were no more hesitations at the time of instantiating the schemes.

10.4 The CBR component

In order to test the developed formalisation of *ProCLAIM*'s CBRc we implemented a prototype application for the transplant scenario. One of the main purposes of this prototype was to show that the CBRc's reasoning cycle, as described in §9.5, can be fully automated. Let us recall that the purpose of the CBRc is to reuse past resolved deliberations in order to solve a target case. This means: *1)* resolve symmetrically attacking arguments into asymmetric ones and *2)* submit additional arguments deemed relevant in previous similar cases.

As described in §9.3, and depicted in figure 10.23 cases in the CBRc prototype are represented by a tree of arguments, a set of facts (a pair of type and value) and the phase in which the case was resolved. As shown in figure 10.23 all these features of the case can be edited by the user before triggering the reasoning cycle. The user may also define the maximum distance between terms instantiating the schemes by which cases are deemed similar. This is, the value of k in the retrieval process as defined in §9.5.1. In figure 10.23, cases will be deemed similar if they are at least 2-similar.

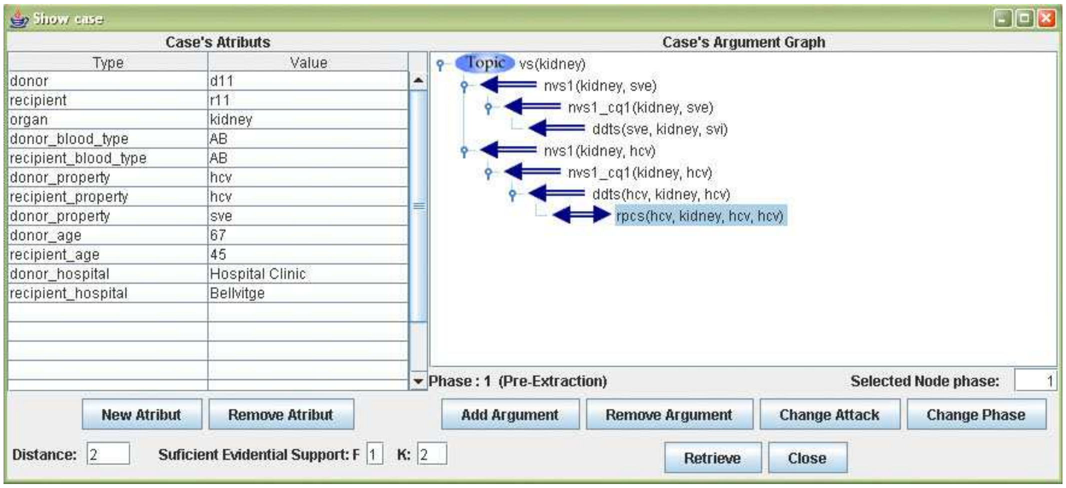
When initiating the reasoning cycle, the CBRc prototype will retrieve those cases which are similar and applicable to the target case, just as defined in §9.5.1. In figure 10.24, the retrieved cases are already organised in the solution proposal set SP, as described in §9.5.2. Let us recall that an element of SP (called *Accrued Argument Graphs* in figure 10.24) is a tuple $<\mathbb{T}, S, F>$, where S is a set of cases that share the same tree of argument $\mathbb{T}$ and the same resolution phase F. When clicking on an element of the list of *Accrued Argument Graphs* the users can see the cases in S. By clicking on the cases' ids, users can browse through all the retrieved cases.

Associated to each *Accrued Argument Graph* there is an evidential support, which in this prototype application is given by the number of cases in S, and the associated phase F. As we can see in figure 10.23, users can set the minimum threshold value for an *Accrued Argument Graph* to have *sufficient evidential support*. In particular, the threshold given in figure 10.23 is $F = 1$ and $K = 2$. This means that *Accrued Argument Graph* which are of phase $F < 1$ or have less than two associated cases in S will not have *sufficient evidential support* and thus will not be accounted for in the reuse phase.

Figure 10.25 depicts the proposed solution tree of arguments. At this stage the user may proceed to revise the solution, where she may change the attack relations, add new arguments or delete existing ones, edit the set of facts and edit the case's associated resolution phase. Once done, the case is retained in the case base as described in 9.5.4.

The cases are stored in XML files organised as described in §9.4, that is, grouped together based on the shared reasoning lines and with a hierarchical relations based on the

their domain expertise.



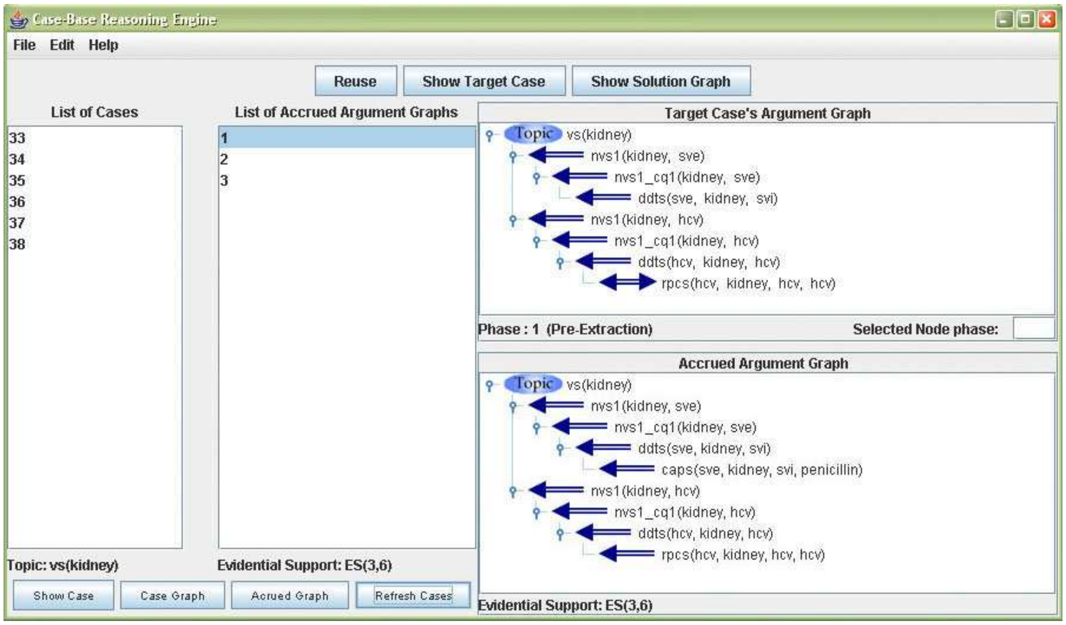
Type	Value
donor	d11
recipient	r11
organ	kidney
donor_blood_type	AB
recipient_blood_type	AB
donor_property	hcv
recipient_property	hcv
donor_property	sve
donor_age	67
recipient_age	45
donor_hospital	Hospital Clinic
recipient_hospital	Bellvitge

```
Topic vs(kidney)
  nvs1(kidney, sve)
    nvs1_cq1(kidney, sve)
      ddts(sve, kidney, svi)
  nvs1(kidney, hcv)
    nvs1_cq1(kidney, hcv)
      ddts(hcv, kidney, hcv)
        rpcs(hcv, kidney, hcv, hcv)
```

Phase : 1 (Pre-Extraction) Selected Node phase: 1

Distance: 2 Suficient Evidential Support: F 1 K: 2

Figure 10.23: A case representation in the CBRc prototype. Through this dialoge window, users can edit the tree of arguments, the set of facts and the case's phase. User can also set the maximum distance value by which cases are deemed similar (the value of k in §9.5.1), and also set the minimum values for F and K for the *sufficient evidential support*. By clicking on the *Retrieve* button, the reasoning cycle begins.



List of Cases: 33, 34, 35, 36, 37, 38

List of Accrued Argument Graphs: 1, 2, 3

```
Topic vs(kidney)
  nvs1(kidney, sve)
    nvs1_cq1(kidney, sve)
      ddts(sve, kidney, svi)
  nvs1(kidney, hcv)
    nvs1_cq1(kidney, hcv)
      ddts(hcv, kidney, hcv)
        rpcs(hcv, kidney, hcv, hcv)
```

Phase : 1 (Pre-Extraction) Selected Node phase:

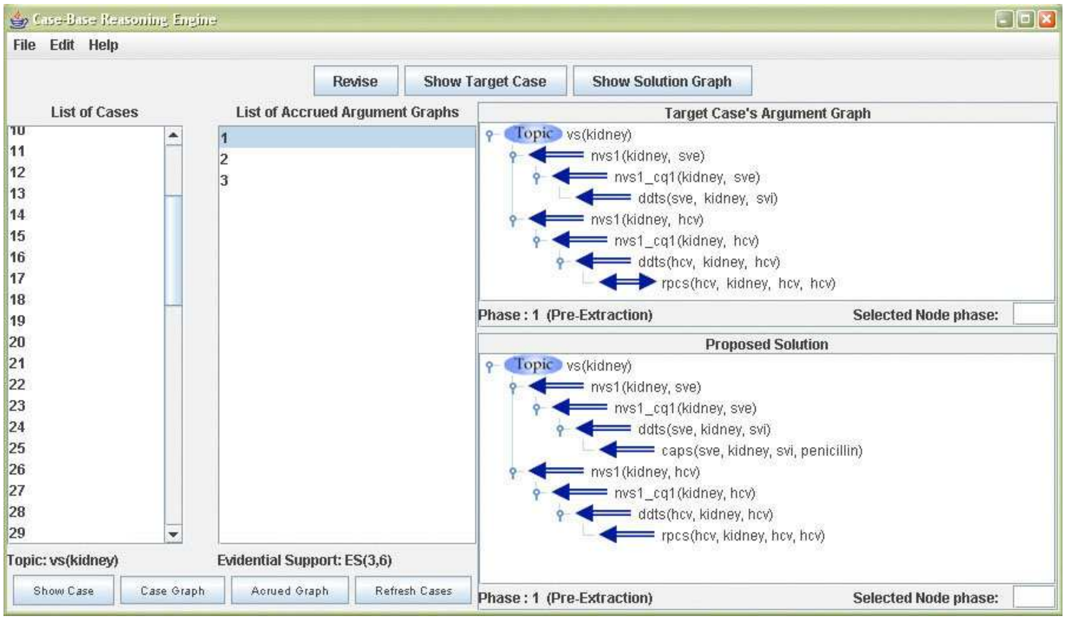
```
Topic vs(kidney)
  nvs1(kidney, sve)
    nvs1_cq1(kidney, sve)
      ddts(sve, kidney, svi)
        caps(sve, kidney, svi, penicillin)
  nvs1(kidney, hcv)
    nvs1_cq1(kidney, hcv)
      ddts(hcv, kidney, hcv)
        rpcs(hcv, kidney, hcv, hcv)
```

Topic: vs(kidney)

Evidential Support: ES(3,6)

Evidential Support: ES(3,6)

Figure 10.24: Retrieval phase in the CBRc prototype, where cases are already grouped in *Accrued Argument Graphs*



```
Topic vs(kidney)
  nvs1(kidney, sve)
    nvs1_cq1(kidney, sve)
      ddts(sve, kidney, svi)
  nvs1(kidney, hcv)
    nvs1_cq1(kidney, hcv)
      ddts(hcv, kidney, hcv)
        rpcs(hcv, kidney, hcv, hcv)
Phase : 1 (Pre-Extraction)          Selected Node phase:
```

Topic: vs(kidney)

Evidential Support: ES(3,6)

```
Topic vs(kidney)
  nvs1(kidney, sve)
    nvs1_cq1(kidney, sve)
      ddts(sve, kidney, svi)
        caps(sve, kidney, svi, penicillin)
  nvs1(kidney, hcv)
    nvs1_cq1(kidney, hcv)
      ddts(hcv, kidney, hcv)
        rpcs(hcv, kidney, hcv, hcv)
Phase : 1 (Pre-Extraction)          Selected Node phase:
```

Figure 10.25: Reuse phase in the CBRc prototype, where a solution proposal is given.

inclusion relation with respect to the structure of the tree of arguments. This facilitates the retrieval and retain processes. All together we have loaded over 100 cases trying to focus on families of three broadly similar situations so that the case base would not be too sparse. Cases can only be added to the case base by following the full CBRc reasoning cycle.

As can be seen in figure 10.23, the argument scheme formalisation used for the prototype application follows the early formalisation presented in [17]. From the perspective of the CBRc implementation the only difference between the two formalisations is that in the current formalisation the factors that need to be compared are already singled out in the scheme instantiation. This makes the implementation slightly simpler. When implementing the CBRc we isolated those terms that needed to be compared in an *ad hoc* fashion. Other than this feature, for all other aspects the two formalisations share the same properties from the perspective of the CBRc implementation.

The overall performance of the CBRc prototype, fully implemented in JAVA[13], was very good. The results we obtained were those expected and were provided with no noticeable delay. The prototype has been shown to domain experts (transplant professionals) with very positive feedback, considering the stage of development of the prototype. The results given in each trial by the CBRc matched their expectation and more importantly, the rational for the whole reasoning cycle was also evident to them. We thus believe that this prototype was successful as a proof of concept of the CBRc's formalisation.

Through the interaction with the CBRc we learned of a possibility we had not anticipated. The CBRc can construct a full tree of arguments only from a list of facts. This

[13] http://www.java.com/

is because, the empty tree of argument was set to be a subtree of any trees of arguments. Thus, for a target case $< \{\}, \mathbb{C}_{F_T}, F_T >$ the CBRc will retrieve initially all the cases from the case base, as described in §9.5.1. On a second filtering process the CBRc will consider only those cases whose associated trees of arguments *apply* to the target case. From then on, the reasoning cycle proceeds as usual. This feature of the CBRc suggests the possibility of changing the decision making workflow. The CBRc can be used to build the first provisional solution as soon as a set of potentially relevant facts are made available. In consequence, the PAs may require to actually participate in the deliberation only if they have anything to add to the solution proposed by the CBRc.

Another more important difference with the formalisation presented in §9 is that in the prototype, the CBRc does not make use of an ontology. The distance between facts and actions is hardcoded and used only for illustrative purposes. We will, in future work, include an ontology based notion of term similarity. Another important limitation of this prototype is that the principles by which we propose how evidence can be used and obtained are relatively basic. In future work we would like to provide a more well-founded notion of evidence, necessary for more realistic scenarios.

10.5 Discussion

In this chapter we have reviewed four software applications we developed[14] as part of our research. In this chapter we believe to have shown, by means of these proof-of-concept applications, the practical realisation of *ProCLAIM*, not only in its final implementation, as shown with CARREL$^+$ and its CBRc, but also in the process of constructing and validating the ASR as illustrated in §10.2 and §10.3 which is central to its applicability.

At the same time, the developed application helped highlight several weaknesses of *ProCLAIM*. Two important limitations of *ProCLAIM* found in the development of CARREL$^+$ are now being addressed at length in this Thesis. The first one being the inadequacy of the dialogue game and the second and more important limitation was the lack of systematic way to construct the ASR. In §5 we propose *ProCLAIM*'s dialogue game that accounts for the particularities of the model's deliberations and while in §6 we proposed a circuit of reasoning patterns for deliberating over safety critical actions, in §7.1 we describe how to build, step by step, the ASR for a given scenario. Furthermore, as discussed in §10.2, we implemented a web application intended to help developers to construct and edit the ASR.

Each of the proposed prototype application helps show the potential of *ProCLAIM* as a principled way to structure and implement the family of applications to which it is intended. At the same time, these applications highlights weakness and point to different lines for future work. Most weaknesses were already discussed in the previous sections.

[14] All presented applications were developed and implemented by the author of this Thesis.

Chapter 11

Two Case Studies

In this chapter we discuss *ProCLAIM* in light of our experience in defining and developing the two case studies, *i.e.* the transplant and the environmental scenarios, introduced in §3. These two case studies had very distinct development processes, which provide a broad perspective on *ProCLAIM*'s applicability, contributions and limitations. The problems addressed in the transplant scenario have initially inspired the conceptualisation of *ProCLAIM* and has guided us throughout its development. The addressed problem and context of this scenario are relatively well understood, and all together we believe to have successfully tackled the issues we aimed to resolve. In the environmental scenario, on the other hand, *ProCLAIM* was used as the initial inspiration for the definition of new scenario within the context of wastewater management. The development of this scenario was undertaken by researchers from a Laboratory of Chemical and Environmental Engineering[1] and thus by researchers who are not familiar with Argumentation and with limited experience in Computer Science in general. This scenario was very useful for identifying the scope of applicability and for the definitions of procedures to facilitate the construction of the Argument Schemes Repository which constitutes the backbone of our proposed model.

In the following section we focus on the transplant scenario describing the different insights we gain through the development of this scenario and interaction with the transplant professionals. In a similar fashion, in §11.2 we describe the development process of the environmental scenario and provide our conclusions from our experience and feedback obtained from its implementation. In §11.2.1 we provide a running example within the environment scenario. In §11.3 we conclude with broader discussion regarding *ProCLAIM*'s applicability, contributions and limitations.

11.1 The Transplant Scenario

Our research in the transplant scenario should be understood as a continuation of the study of the transplant domain from the perspective of CARREL [224], an electronic institution intended to facilitate the offer and allocation of human organs for transplantation. By formalising most of the logistics of the transplant process, accounting for high sensitive of

[1] http://lequia.udg.es/eng/index.htm, from the University of Girona

environment, CARREL provides the opportunities to address and formalise more complex problems assuming a multi-agent system. Our aim was to make use of this opportunity. Central to the transplant domain is the problem of shortage of organ availability of transplantation, where big efforts in the medical community are devoted to reduce this gap between demand and supply [77, 175, 236]. Hence, the immediate choice was to envision ways in which CARREL could be extended so as to address this problem. Broadly speaking, there are four complementary strategies intended to fight this problem. These are: *1)* Education, with the aim of raising awareness of the value of organ donation in order to increasing actual organ donors[2]; *2)* the study and implementation of alternative medical treatments, such as the use of artificial organs [105] or xenotransplantation[3] [216]; *3)* the us of living donors, in particular for kidney and liver [172, 162] and finally; *4)* the expansion of the organ acceptability criteria so as to minimise the discard of human organs [23], in which we could include efforts to make use of organs from non-heartbeating donors [77].

No doubt all these strategies are of great value. However, in analysing all four options we found the last one to be particularly interesting due to its intrinsic value and challenging nature. The first two approaches involving education programs and alternative treatments respectively, fall somewhat outside the scope of CARREL. The inclusion of living donors in the formalisation of CARREL may be resolved, in its simplest fashion, by defining an additional source of organ donation into the workflow. Our challenge was thus to propose how a multi-agent system can provide support to the transplant professionals' efforts in optimising the use of available organs by continuously challenging established guidelines.

To address this problem, we followed a standard path of conceptualisation, formalisation and later implementation. Firstly and always under supervision of transplant professionals of the Sant Pau Hospital, we conceptualised the problem at hand in terms suitable for CARREL [14, 13]. This has resulted in the proposal of the alternative human organ selection process, presented in §3.1.3, in which, broadly speaking, through justifying their assessments, transplant professionals are given the opportunity to make use of human organs that under current policy would be discarded. This proposal has been broadly discussed with transplant professionals and has been presented in national and international transplant conferences [9, 10, 15] where we obtained positive and valuable feedback. The design and formalisation of this process was later published as a journal paper in [12] and ultimately, as discussed in §10.1, it was implemented and presented as the main Large Scale Demonstrator of the FP6-EU Project ASPIC, [11], where it was deemed by one of the project reviewer as the most sophisticated Argumentation-based application.

The development of the transplant scenario included regular meetings with transplant professionals of the Sant Pau Hospital where we presented our proposals. While the central subject at the early meetings were the definition of the alternative human organ selection process, in later meetings we focused on the development of motivational examples and the formulation of the argument schemes and CQs specialised for the transplant scenario. These meetings helped us validate our proposals from the perspective of the medical domain as well as from the perspective of the end users. It was throughout these meetings that

[2] http://www.organdonor.gov/grantProgramPublicEdu.asp

[3] Xenotransplantation is the transplantation of living cells, tissues or organs from one species to another.

we deemed as necessary the use of specialised argument schemes. In our discussions we noticed a clear shift in engagement as soon as argument schemes were strip out from any formal notation and presented using only medical jargon. Patterns such as Walton's [233] or Atkinson's *et al.* [34] argument schemes were initially used to illustrate the general principles, but while the principles were understood, we found that the leap between these schemes and the arguments needed to be produced was too big to be performed in a realtime discussion and in a consistent manner. At first this process involved certain overhead for the participants and soon after, participants were disengaged with the task at hand.

At that time we developed an early formalisation of the argument schemes and CQs presented in [17] with near to fifty specialised schemes [7]. For this process we made use of the ASR browser, presented in 10.3. While the initial intention of this online ASR browser was for the transplant professionals to validate our proposed schemes on their own, not during the meetings, do to their tight schedules we learned this to be unrealistic. Nonetheless, the interaction with the ASR browser during the meetings was very useful to overcome certain scepticism towards the somewhat abstract notion of *reasoning patters* or *argument schemes*. The ASR browser thus helped convey the idea that specialised schemes can be instantiated with no overhead and that the connection between schemes via CQs is not only a conceptual construct but can translate into a simple user-computer interaction that helps users navigate among the schemes implemented in this application as hyperlinks.

Discussions of the medical scenario were extended to many informal meetings with transplant professionals besides those of the Sant Pau Hospital. Among these discussions we would like to mention the presentation we made to the board of directors of the Argentinean Transplant Organisation INCUCAI[4]. In this presentation we discussed both the alternative organ transplantation and the argument schemes were discussed. The feedback was highly positive and it included a proposal for collaborating in their ICT project SINTRA[5] for the development of a computer system for the management and oversight of procurement activities and transplantation of organs, tissues and cells at the national level. This proposal is included in our plans for future works.

In summary, we believe to have successfully addressed our initial aim. That is, to extend CARREL so as to support mechanism to help combat the shortage of organ availability for transplantation. More in particular, we believe to have successfully illustrated how a multiagent system may support an alternative organ selection process that while accounts for the criticality of the context it also supports the transplant professionals' effort to make the best use of the available organs, where this may involve challenging consented criteria for organ acceptability and may possibly involve disagreement between transplant professionals themselves. The proposal, design and final implementation of the alternative organ selection process constitute one of the main contribution of the medical scenario. While there is a large number of works relating Artificial Intelligence techniques and the medical domain, and in particular involving organ or tissue transplantation, to the best of our knowledge, no other work addresses similar problematics to those explored in the medical

[4] http://www.incucai.gov.ar

[5] http://sintra.incucai.gov.ar/intro.html

scenario (see §3.1). Another important contribution achieved within this scenario includes the implementation of an argument-based deliberation among heterogeneous agents[6]. In this implementation we have covered not only *1)* the actual exchange of arguments, but also *2)* illustrated how these arguments can be constructed (see §10.1.2) given the provision of specialised argument schemes and *3)* we have develop procedures for the construction of these specialised schemes(see §7). To the best of our knowledge no other work has encompassed these three aspects which we believe are necessary for the actual implementation of argument-based dialogues for real life applications. We continue this discussion in §11.3 after having analysed the environmental scenario.

Our main concern, thus far, during the implementation of the model has been to shed light on the question: how agents can construct arguments that *1)* embody the agents' positions; *2)* include only that which is relevant from the deliberation's perspective, without making strong assumptions about the agents' reasoning capabilities; and *3)* ensure that the exchanged arguments are understood by the interlocutors. While we believe to have made important progress in addressing these three issues, and in general we believe to have shown that *ProCLAIM* is a suitable approach for addressing the transplant scenario, other aspects of the model require further development. In particular, more work needs to be done regarding argument evaluation. Substantial efforts has been devoted in laying out the basic principles for the argument evaluation in §8. However, future work should attempt to instantiate this framework with more realistic cases in terms of the three dimensions by which arguments are evaluated (*i.e.* evidence, consented knowledge and argument endorsement).

Other minor task for future work relate to the construction and use of the ASR. As the ASR increases to include more argument schemes, agents may be exposed to wider set of options for reply certain arguments. While we do not believe this to be a problem for artificial agents, our concern is that human agents might be overwhelmed if presented with more than a handful set of schemes. This, of course, may hinder the deliberation process. We believe this problem can be addressed by allowing users to indicate which are the factors they believe to be relevant for the deliberation and on this basis construct the most likely arguments. The users may then edit those arguments if required. This involves developing the appropriate graphical user interface that would facilitate this interaction. At the same time, this may also require developing reasoning mechanisms to interpret the user's intention.

Having explored a great variety of examples in the transplant scenario and shown a running prototype to the potential end users of the application, the next step in this line of work is to develop more rigorous evaluation of the proposed application, possibly in the form of a simulation exercise with the potential end users. While positive results were obtained when presenting our developed prototype to both to the transplant professionals at the Sant Pau Hospital and to those at the Argentinean Transplant Organisation, any move forward in this scenario should be preceded by such rigorous analysis. In fact, this was our intended plan of work prior to explore the environmental scenario. However, as we got involved in this new scenario, we realised that developing the specialised schemes for a new application

[6] Whereby heterogeneous agent we do not imply an open system, rather, that agents may be human or artificial, and those artificial agent may have diverse implementation.

was not trivial, particularly for developers who are not familiar to Argumentation Theory. Hence, we devoted our efforts to facilitate procedures for the production of these schemes. In order to do this, we developed the argument schemes and CQs of intermediate level of abstractions, that is the circuit of schemes and CQ specialised for reasoning over safety critical actions presented in §6 and published in [16]. We now discuss our experience of implementing the *ProCLAIM* model in the environment scenario.

11.2 The Environmental Scenario

The development of the environment scenario was lead by members of researchers from a Laboratory of Chemical and Environmental Engineering[7]. The starting point of this scenario was the proposal to apply the *ProCLAIM* model in the context of wastewater management, as discussed in §3.2. This proposal gave us a great opportunity to test *ProCLAIM*'s applicability in other scenarios. Furthermore, while the transplant scenario provide us feedback from the view point of the end users, the environmental scenario gave us feedback from the perspective of the developers of a new application, where these developers were neither familiar with the *ProCLAIM* model nor Argumentation Theory and in general, their experience in Computer Science was rather limited.

The environmental scenario grew into an ambitious proposal for an integrated management of wastewater in a river basin [39]. The proposal was centred in the formalisation of a multi-agent systems intended to optimise infrastructure operations while minimising the environmental impact of industrial discharges in emergency situation. The idea is that in normal situations the multi-agent system would manage the information exchange, among the relevant actors, for planning *normal* (expected) industrial discharges, and thus addressing scheduling problems while ensuring guidelines and legislation enactment. Then, in emergency situations, where a more dynamic approach is necessary to resolve *ad-hoc* situations, the system would help identify safe solutions through the deliberation among all the relevant actors and thus, making the best use of the agents' expertise. The *ProCLAIM* model was thus used to support this collaborative decision making process with the purpose of reducing the environmental impact of industrial discharges.

The addressed problem is particularly challenging for the variety of actors involved in the process (industries, sewer systems, wastewater treatment plants, storing tanks...) with diverse interests and knowledge about problem. Also challenging because of the variety of external factors relevant for the decision making (*e.g.* meteorological context, mechanical failure in an infrastructure, concurrent wastewater produced by other sources, industrial and non-industrial, etc...). One important part of the work in developing this scenario was devoted to the identification of all these actors, their interrelations, roles and tasks. This was an important difference with respect to the transplant scenario, where the underlying system of agents, their relations and workflows was already well defined and formalised. In the environmental scenario however, *ProCLAIM* was applied before having a well understanding of the problem at hand. Another important difference between the two case studies was the availability of literature debating about what constitute a safe action (*i.e.* an industrial

[7] http://lequia.udg.es/eng/index.htm, from the University of Girona

spill or an organ transplant) beyond the definition of standard regulations. In the transplant domain there is plenty of literature discussing cases of organ transplantation and the circumstances that presumably lead to the success or failure of the organ transplant, where the purpose of this analysis is to improve existing guidelines and donor and organ acceptability criteria. In fact, these discussions were the precursors that motivated the development of the transplant scenario. However, substantially fewer discussions of this nature were found in the environmental engineering domain addressing the safety of an industrial spill. This circumstance, made even more challenging the development of the environmental scenario, because, without the availability of case analysis it was hard to produce realistic examples. Furthermore, without a clear account of the content of the deliberations or the agents involved in it, it was very difficult to produce specialised argument schemes for this scenario. This was particularly difficult at an early stage of development of *ProCLAIM*, when schemes were produced in an *ad-hoc* fashion.

Through the development of the environmental scenario we learned of the importance of providing support in the construction of the specialised argument schemes and CQs. Our initial approach was to introduce the environmental engineers (from now on would be referred to as the *developers*) to the principles of argumentations, in particular to argument schemes and CQs and then, through joint work propose a set of specialised schemes in order to capture an initial set of examples, with the assumption that the scheme construction would be tedious but not as difficult as it turned to be. While developers found the concepts of Argumentation intuitive, particularly with the support of visual representations such as Dung's graphs [80], we found numerous mistakes and misconceptions in the schemes they produced. This was in part due to a weak understanding of the problem at hand, but also due to the missing support in the scheme construction. Hence, while in the transplant scenario lead us to recognise the need for specialised schemes for real-time deliberation among heterogeneous agents, the main conclusions drown from the environmental scenario was the requirement to deliver guidance for the construction of these specialised schemes.

In order to respond to this requirement, we have developed the circuit of schemes and CQs intended for deliberating over safety critical actions presented in §6, with the main purpose of providing guidance in the construction of the Argument Scheme Repository as presented in §7. An early formulation of this circuit of schemes and CQs was proposed in [219], on the basis of which the environmental scenario was initially proposed in [2]. This early formulation was very useful in the schema construction but also in providing a clearer definition of scope of *ProCLAIM*'s deliberations. To further support the construction of ASRs we implemented the Argument Scheme Editor (see §10.2), a prototype web applications for the construction of the ASR, based on [219]. All together allow the developers have a clearer understanding of the scope of the *ProCLAIM*, and thus to focus the examples to cases than can be covered by the model's deliberation. For example, one recurrent case prior to [219], was the proposal of examples based more on planning, a feature that currently *ProCLAIM* is missing. Another recurrent error was to justify the safety of an action based on its compliance with existing regulations. With the new formulation it was clearer to developers that arguments where made only of cause effects relations, and that aspects such as regulations are perspectives by which the given arguments are evaluated.

All together, we believe that, despite the above mentioned challenge, the results arrived in this scenario are very positive, producing among other outcomes a PhD Thesis in Environmental Engineering [39] and a journal paper [3] among other publications [6, 2, 1]. In the following section we present one of the developed scenarios introduced in [3]. The formulation of this example was initially developed using the formalisation of schemes a CQs presented in [219]. While this formalisation constitute an improvement with respect to our initial *ad-hoc* approach [17], it is too schematic with a somewhat vague underlying logical formulation, and it does not have an associated dialogue game. We have substantially improved this early formulation into the mature work presented in §6 and published in [16]. We will use this later formulation to present the environmental scenario along with the dialogue game introduced in §5.

11.2.1 Running Example

To situate ourselves, let us first overview the environmental scenario introduced in §3.2. In [39] Aulinas proposes a multi-agent system for an integrated management of wastewater in river basin. The agents of this system represent the different actors and infrastructures involved in the Urban Wastewater System (UWS). The purpose of the UWS is to cope with the domestic wastewater production as well as the wastewater produced by the industries. Domestic and industrial wastewater are connected to sewer system which collects along with the rainfall the wastewater to transport it to the Wastewater Treatment Plants (WWTPs). Once appropriately treated the WWTP spills the wastewater into the final receiving media, in this case the river.

As above mentioned, *ProCLAIM* takes a part in this system only in emergency situations, when normal procedures are either non-viable or unsafe. In [39] Aulinas identifies a number of unsafe situations, these are mostly focused on the content of industrial spill and how this content may affect environment and whether it can damage the WWTP operability, as the latter situation may lead to more acute situations. Prototypical situations that must be avoided in the WWTP are for example, bulking or foaming, both potentially caused by the growth of filamentous bacteria in the WWTP's tank. If any of these two situations occur, the WWTP will significantly reduce its effectiveness in the treatment processes and will produce poor quality effluent. The cases studied include wastewater with *toxic substances*, that contain *heavy metals*, excessive amounts of *nutrients* (principally nitrogen and phosphorous) or excessive amounts of *biodegradable organic components*. For each of these situations, as well as for other cases explored in [39], the question is whether the system can coupe with these kind of spills. That is, can these spills be exceptionally undertaken by the systems in a way that there is no direct harmful effect in the environment and no infrastructure (mainly the WWTP) is damaged in this process to a degree of substantially reducing its operability. In this section we will explore one example in which an industry has to spill substances toxic substances.

In [39] a complex multi-agents system is defined, with over twelve roles and their associated protocols of interaction, the services the infrastructures can provide and their responsibilities with the UWS. For the purpose of our example we will introduce only two of the agents' role, in their succinct version. Agents enacting these roles will participate in collaborative deliberation to decide the safety of the industrial spill. These roles are:

- **Industry Agent (InA)**: represents individual industries and/or groups of industries that need to manage their produced wastewater as a result of their production process.

- **Wastewater Treatment Agent (WTA)**: represents the manager of WWTP. Its main function is to keep track of wastewater flow arriving at WWTP as well as to supervise and control the treatment process. In [39], this role is further refine into the managers of WWTP (WTA_M) and the operators (WTA_O). For simplicity, we will consider the WTA as a single role.

For the purpose of the example, let us suppose that an industry represented by agent InA has to spill wastewater into the sewer system and into the WWTP, represented by WTA. Let us suppose as well that the spill contains substantial amount of a toxic substance, for example *Cadmium VI*. One of the main effects this may cause on the WWTP is the inhibition of Extracellular Polymeric Substances (EPS)[8], these substances are important in the flocculation process employed in the purification of wastewater and sewage treatment. The EPS inhibition prevents the formation of large flocs that are used to remove organic material. This situation is typically called *pinpoint floc* (for the sparse growth of flocs of the size of a pinpoint) and its immediate implication is the a substantial reduction in the effectiveness of the WWTP. Of course in normal circumstances, wastewater containing any toxic should be first treated by the industry and only discharged when toxic concentrations are bellow given thresholds. In emergency situations however, when these thresholds cannot be realised, the UWS should coordinate to find cases by case solutions that would minimise the environmental impact of the discharged wastewater.

To define the elements of the deliberation we need first to identify the space of facts, actions, side effects and undesirable goals, namely: $\overline{\mathbf{R}}$, $\overline{\mathbf{A}}$, $\overline{\mathbf{S}}$ and $\overline{\mathbf{G}}$. For the purpose of the example, let us suppose these sets contain the predicates and propositions defined in table 11.2.1.

Now, let us introduce a fragment of the scenario's ASR. In specialising the initial $AS1$ scheme we can identify two action proposal:

$AS1_{Ind\_w}$: `propose({ind_ww(Ind,WW_id),ind_wwtp(Ind,SS,WWtp)}, {discharge(WW_id,WWtp)})`

| *Discharge* ***`WW_id`*** *into the WWTP*

$AS1_{Ind\_r}$: `propose( {ind_ww(Ind,WW_id),ind_rive(Ind,SS,R)}, {discharge(WW_id,R)})`

| *Discharge* ***`WW_id`*** *into the river*

To each scheme there is associated the CQ questioning whether there are any contraindications for the actions proposals. For the former scheme $AS1_{Ind\_w}$:, this CQ is embodied as an attack by schemes indicating the possible undesirable side effects the discharge may

[8] Extracellular Polymeric Substances are high-molecular weight compounds secreted by microorganisms into their environment.

Set	Ungrounded Predicate	Description
$\overline{\mathbf{R}}$	ind_ww(Ind,WW_id)	industry Ind has wastewater WW_id
	ind_river(Ind,SS,R)	industry Ind is connected to the river R via the sewer system SS
	ind_wwtp(Ind,SS,WWtp)	industry Ind is connected to the WWTP WWtp via the sewer system SS
	wwtp_param(WWtp,Par,Val)	The WWtp has design parameter Par with value Val
	wwtp_cond(WWtp,Cond)	The WWtp has circunstancial condition Cond
	ww_cons(WW_id,Cons)	The discharged wastewatre WWtp has Cons
	ww_flow(WW_id,F)	The discharged wastewatre WWtp has flow F
	meteo_cond(Met)	Met are the expected Meteorological conditions
$\overline{\mathbf{A}}$	discharge(WW_id,RM)	Discharge the wastewater WW_id into the receiving media RM
	wwtp_treat(WWtp,T)	Treatment T will be performed at the WWTP WWtp
$\overline{\mathbf{S}}$	wwtp_c(WWtp,C)	The WWTP WWtp will have condition C
	discharge_p(D,P)	Discharge content D has property P
$\overline{\mathbf{G}}$	pinpoint_floc	Small floc that settle very slowly
	sludge_toxicity	Accumulation of toxic substances in the sludge
	fil_bulking	Bulking due to an overgrowth of filamentous bacteria
	viscous_bulking	Sludge becomes viscose and compact

Table 11.1: A subset of the elements of $\overline{\mathbf{R}}$, $\overline{\mathbf{A}}$, $\overline{\mathbf{S}}$ and $\overline{\mathbf{G}}$ for the given discussion

cause to the WWTP. For the later scheme $AS1_{Ind\_r}$:, the attacks indicate the possible undesirable side effects the discharge may cause to the river. Among the schemes embodying $AS1_{Ind\_w}\_CQ1$ (the CQ of $AS1_{Ind\_w}$) are:

$AS2_{Ind\_pp}$: argue({ind_ww(Ind,WW_id),ind_wwtp(Ind,SS,WWtp)},
{discharge(WW_id,WWtp)}, contra(ww_cons(WW_id,C),
wwtp_c(WWtp,eps_inhib), ***pinpoint_floc***))

| *Substace* ***C*** *will cause* $eps\ \ inhibition$ *leading to* $pinpoint\ \ floc$

$AS2_{Ind\_rs}$: argue({ind_ww(Ind,WW_id),ind_wwtp(Ind,SS,WWtp)},
{discharge(WW_id,WWtp)}, contra(ww_cons(WW_id,C),
wwtp_c(WWtp,desnitrif), ***rising***))

| *Substace* ***C*** *will increase* $desnitrification$ *in the secondary reactor settler leading to* $rising$

$AS2_{Ind\_hs}$: argue({ind_ww(Ind,WW_id),ind_wwtp(Ind,SS,WWtp)},
{discharge(WW_id,WWtp)}, contra(meteo_cond(rainfall),
wwtp_c(WWtp,overflow), ***hydraulic_shock***))

| *Due to* $rainfall$*, incoming flow will exceed the WWTP capacity causing an* $hydraulic\ \ shock$

As discussed above, the reasoning line we explore is the one captured by scheme $AS2_{Ind\_pp}$ in which the discharge of wastewater can cause *pinpoint floc*. Following the formalisation presented in §6, among the associated CQs to $AS2_{Ind\_pp}$ are: $AS2_{Ind\_pp}\_CQ1$, $AS2_{Ind\_pp}\_CQ2$ and $AS2_{Ind\_pp}\_CQ4_1$ are the specialised schemes and challenge:

$AS3_{Ind\_pp\_1}$: argue({ind_ww(Ind,WW_id),ind_wwtp(Ind,SS,WWtp),
ww_cons(WW_id,C)},{discharge(WW_id,WWtp)},no_side_effect(
{meteo_cond(Met)},{wwtp_c(WWtp,eps_inhib)}))

| *Meteorological conditions* ***Met*** *can prevent the* $eps\ inhibition$

$AS3_{Ind\_pp\_2}$: argue({ind_ww(Ind,WW_id),ind_wwtp(Ind,SS,WWtp),
ww_cons(WW_id,C)},{discharge(WW_id,WWtp)},no_side_effect(
{wwtp_cond(WWtp,Cond)},{wwtp_c(WWtp,eps_inhib)}))

| *Condition* ***Cond*** *in the WWTP prevent the* $eps\ inhibition$

$AS5_{Ind\_pp\_1}$: argue({ind_ww(Ind,WW_id),ind_wwtp(Ind,SS,WWtp),
ww_cons(WW_id,C)},{discharge(WW_id,WWtp)},preventive_action(
{wwtp_treat(WWtp,T)},{},{wwtp_c(WWtp,eps_inhib)})))

| *Performing* ***T*** *can overcome* $eps\ inhibition$ *on the WWTP*

$AS2_{Ind\_pp\_CQ4_1}$: challenge(evidence(ww_cons(WW_id,C)))

| *Provide evidence that* ***ww_cons(WW_id,C)***

Associated to the CQ $AS2_{Ind\_pp\_CQ4_1}$, thus in reply to the challenge, are schemes specialising scheme $AS2ev$, for instance:

$AS2ev_{Ind\_pp\_CQ4_1\_1}$: `argue({ind_ww(Ind,WW_id),ind_wwtp(Ind,SS,WWtp)},`
`{discharge(WW_id,WWtp)}, contra(replace_ev(ww_cons(WW_id,C),`
`{test(WW_id,Tst,Res)}),wwtp_c(WWtp,eps_inhib),pinpoint_floc))`

To conclude, let us include in the set of specialised schemes, the scheme $AS6_{Ind\_pp\_2\_1}$, embodying the CQ: *–Are current circumstances such that an undesirable side effect will occur?–* associated to $AS3_{Ind\_pp\_2}$:

$AS6_{Ind\_pp\_2\_1}$: `argue({ind_ww(Ind,WW_id),ind_wwtp(Ind,SS,WWtp),`
`ww_cons(WW_id,C), wwtp_cond(WWtp,Cond)},{discharge(WW_id,WWtp)},`
`contra( {sludge_p(WW_id,Tx)}, toxic_in_sludge))`

| *Toxic* ***Tx*** *will remain in sludge*

Having introduced subsets of the four dimensions: $\overline{\mathbf{R}}$, $\overline{\mathbf{A}}$, $\overline{\mathbf{S}}$ and $\overline{\mathbf{G}}$ and a fragment of the ASR, we can now proceed to run the deliberation following the dialogue game proposed in §5. A *ProCLAIM* deliberation begins with an argument proposing the main action via the `open_dialogue` locution at the Open Stage. In this example an industry `ind` arrives to a emergency situation in which they have to discharge wastewater containing toxic substances. In this case, the agent representing this industry, InA, with id `inA`, will initiate the proposal for a deliberation by sending a request to the MA, for which InA instantiates scheme $AS1_{Ind\_w}$. The exchanged messages for this example are depicted in Figure 11.1. The following seven messages initiate the deliberation, where `wwtp` is the WWTP connected to the industry through the sewer system `ss`, and where the basic data provided by the deliberation agents (`ind_basic_info` and `wwtp_basic_info`) are the sets of facts given in Table 11.2.1. Also, let `proposal` denote `propose({ind_ww(ind,ww_id), ind_wwtp(ind,ss,wwtp)},{discharge(ww_id,wwtp)})`, then the deliberation begins with the following exchanged messages:

`request(inA,ma,0,-1,open_dialogue( proposal))`

`inform(ma,all,conv_id,1,0,open_dialogue(proposal))`

`request(inA,ma,conv_id,2,1,`
`enter_dialogue(proposal,inA, ind_basic_info))`

`inform(ma,all,conv_id,3,2,`
`enter_dialogue(proposal, inA, ind_basic_info,` $\{$`ma`$\}$, $\mathbb{C}_{F\wedge A}$,
$\mathbb{T}$`,legal_replies))`

`request(wta,ma,conv_id,4,1,`
`enter_dialogue(proposal,wta, wwtp_basic_info))`

Agent	Submitted Facts	Description
InA	`ind_ww(ind,ww_id)`	industry `ind` has wastewater `ww_id`
	`ind_wwtp(ind,ss,wwtp)`	industry `ind` is connected to the WWTP `wwtp` via the sewer system `ss`
	ww_flow(ww_id,100m^3/d)	the flow of the discharge is of 100m^3/d
	`ww_cons(ww_id,ammonia)`	The wastewater contains ammonia
	`ww_cons(ww_id,cdVI)`	The wastewater contains Cadmium VI
WTA	`wwtp_par(wwtp,typ,act_sludge)`	is an activated sludge plant
	wwtp_par(wwtp,flow,300m^3/d)	the WWTP has a flow capacity of 300m^3/d
	`wwtp_par(wwtp,prim_clarif,2)`	Two primary clarifiers
	`wwtp_par(wwtp,aer_basin,2)`	Two aeration basins
	`wwtp_par(wwtp,aer_tp,diffuse)`	The plant with diffused aeration

Table 11.2: Information made available by the InA and the WTA when initiating the deliberation

inform(ma,all,conv_id,5,4,
  enter_dialogue(proposal, wta, wwtp_basic_info,$\{$ma,inA$\}$ $\mathbb{C}_{F\wedge A}$,
  $\mathbb{T}$, legal_replies))

At this stage the context of facts and actions $\mathbb{C}_{F\wedge A}$ contains the agents provided basic info and the proposed action. The MA has provided the deliberating agents with the list of legal replies, among which are the schemes: $AS2_{Ind\_pp}$, $AS2_{Ind\_vb}$, $AS2_{Ind\_rs}$ or $AS2_{Ind\_hs}$, presented above. In particular, the scheme for pinpoint floc is given:

argue($\mathcal{C}$,$\mathcal{A}$, contra(ww_cons(ww_id,C),wwtp_c(wwtp,eps_inhib),
pinpoint_floc))

| *Substace* ***C*** *will cause* $eps\ \ inhibition$ *leading to* $pinpoint\ floc$

Where $\mathcal{C}$ = {ind_ww(ind,ww_id),ind_wwtp(ind,ss,wwtp)} and $\mathcal{A}$ = {discharge(ww_id,wwtp)}. The WTA can instantiate this scheme by grounding `C` with `cdVI` to form argument $A2$ attacking the initial action proposal, which we can denote as argument $A1$ (see Figure 11.2a). This involves the exchange of the following two messages:

request(wta,ma,conv_id,6,5,argue(
  argue($\mathcal{C}$,$\mathcal{A}$, contra($\{$ww_cons(ww_id,cdVI)$\}$,
  $\{$wwtp_c(wwtp,eps_inhib)$\}$,pinpoint_floc)),1)).

Assuming this argument is accepted by the MA it will be added to $\mathbb{T}$ with id *2* and broadcasted to all participants:

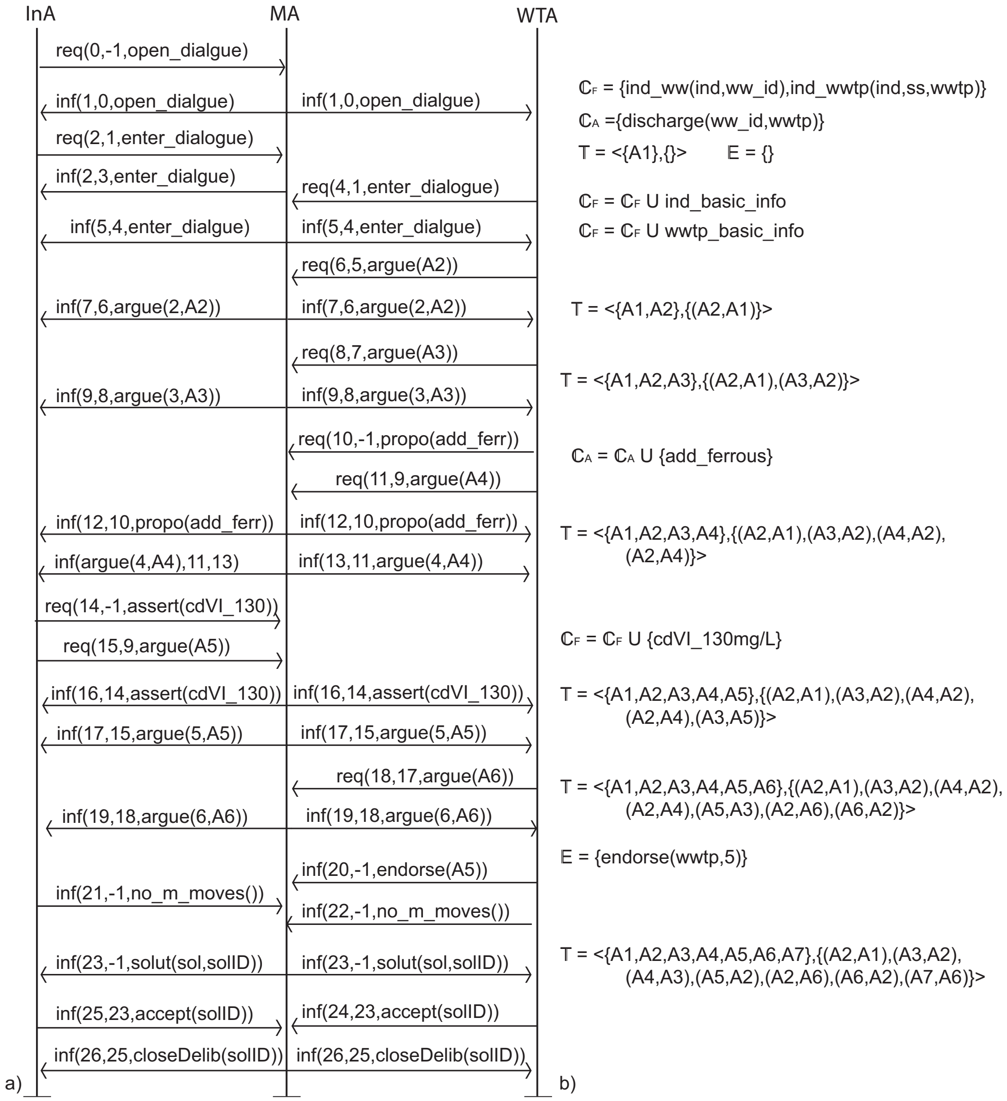


Figure 11.1: Figure illustrating the agents exchanged messages and how the sets $\mathbb{C}_F$, $\mathbb{C}_A$, $\mathbb{T}$, and $\mathbb{E}$ are accordingly updated.

```
inform(ma,all,conv_id,7,6,argue(2,
  argue(𝒞,𝒜,contra({ww_cons(ww_id,cdVI)},
  {wwtp_c(wwtp,eps_inhib)}, pinpoint_floc)),1,legal_replies))
```

Again, to this message, the MA attaches the legal replies to attack argument $A2$. These legal replies include the above introduced schemes and CQs: $AS3_{Ind\_pp\_1}$, $AS3_{Ind\_pp\_2}$, $AS5_{Ind\_pp\_1}$ and $AS2_{Ind\_pp\_CQ4_1}$. The WTA may choose to submit a challenge and two arguments. Firstly, using CQ $AS2_{Ind\_pp\_CQ4_1}$ the agent may request the submission of challenge $A3$ to which the MA will respond by adding $A3$ to $\mathbb{T}$ and broadcasting the new submitted challenge:

```
request(wta,ma,conv_id,8,7,argue(
  challenge(evidence(ww_cons(ww_id,cdVI))),2)).
```

```
inform(ma,all,conv_id,9,8,argue(3,
  challenge(evidence(ww_cons(ww_id,cdVI))),2,legal_replies)).
```

The WTA may then propose as a palliative action to prevent EPS inhibition to add coagulant. For this, the WTA first submits an actions proposal `wwtp_treat(wwtp, add_ferrous)` to the set $\mathbb{C}_A$ and instantiate scheme $AS5_{Ind\_pp\_1}$ – *Performing* ***T*** *can overcome* $eps\ \ inhibition$ *on the WWTP*– where treatment `T` is the only variable to instantiate and can be grounded with `add_ferrous` and so construct argument $A4$ (see Figure 11.2).

```
request(wta,ma,conv_id,10,-1,
  propose(wwtp_treat(wwtp,add_ferrous))).
```

```
request(wta,ma,conv_id,11,9,argue(
  argue(𝒞,𝒜, preventive_action({wwtp_treat(wwtp,add_ferrous)},
  {},{wwtp_c(wwtp,eps_inhib)})),2)).
```

Assuming the action proposal and the argument are accepted by the MA they will respectively be added to $\mathbb{C}_A$ and $\mathbb{T}$ and broadcasted all participants:

```
inform(ma,all,conv_id,12,10,
  propose(wwtp_treat(wwtp,add_ferrous))).
```

```
inform(ma,all,conv_id,13,11,argue(4,
  argue(𝒞,𝒜, preventive_action({wwtp_treat(wwtp,add_ferrous)},
  {},{wwtp_c(wwtp,eps_inhib)})),2,legal_replies)).
```

Meanwhile, the InA may wish to reply to challenge $A3$ for which the agent may first add the fact `test(www_id,aas,cdVI_135mg/L)`, indicating a that the wastewater showed a 135ml/L concentration of *Cadmium VI* using an Atomic Absorption Spectroscopy test. Then, the agents may instantiate scheme $AS2ev_{Ind\_pp\_CQ4_1\_1}$ given by the MA as a legal reply to the challenge $A3$:

```
request(indA,ma,conv_id,14,-1,
   assert(test(www_id,aas,cdVI_135mg/L))).

request(indA,ma,conv_id,15,9,argue(
   argue({ind_ww(ind,ww_id),ind_wwtp(ind,ss,wwtp)},
   {discharge(ww_id,wwtp)}, contra(replace_ev(ww_cons(ww_id,cdVI),
   {test(ww_id,aas,cdVI_135mg/L)},wwtp_c(wwtp,eps_inhib),
   pinpoint_floc))),3)).
```

Again, assuming these two messages are accepted by the MA, test(www_id,aas, cdVI_135mg/L) will be added to $\mathbb{C}_F$ and the submitted argument, say $A5$, will be added to $\mathbb{T}$. Accordingly, the MA will broadcast the submission of a new fact and of a new argument:

```
inform(ma,all,conv_id,16,14,
   assert(test(www_id,aas,cdVI_135mg/L))).

inform(ma,all,conv_id,17,15,argue(5,
   argue({ind_ww(ind,ww_id),ind_wwtp(ind,ss,wwtp)},
   {discharge(ww_id,wwtp)}, contra(replace_ev(ww_cons(ww_id,cdVI),
   {test(ww_id,aas,cdVI_135mg/L)},wwtp_c(wwtp,eps_inhib),
   pinpoint_floc))),3)).
```

To conclude let us suppose the WTA propose another reason for which the spill can safely be preform. The WTA argument is that if there happens to be fungi in the biomass, the Cadmium VI, can be reduced to Cadmium III, a lesser toxic form of metal and thus preventing the EPS inhibition. To do so, the WTA may use the scheme $AS3_{Ind\_pp\_2}$ – *Condition* ***Cond*** *in the WWTP prevent the* $eps \quad inhibition$ – given as a legal reply to argument $A2$. To instantiate scheme $AS3_{Ind\_pp\_2}$ is to instantiate variable Cond, in this case as wwtp_cond(wwtp,fungi_biomass). The WTA can thus request to submit argument $A6$, and assuming the MA accepts this argument it will be added to $\mathbb{T}$ with id 6:

```
request(wta,ma,conv_id,18,17,argue(
   argue({𝒞,𝒜, no_side_effect({wwtp_cond(wwtp,fungi_bio_mass)},
   {wwtp_c(wwtp,eps_inhib)})),2)).

inform(ma,all,conv_id,19,18,argue(6,
   argue({𝒞,𝒜, no_side_effect({wwtp_cond(wwtp,fungi_bio_mass)},
   {wwtp_c(wwtp,eps_inhib)})),2)).
```

Note that wwtp_cond(wwtp,fungi_bio_mass) is not in $\mathbb{C}_F$. That is, $A6$ is an hypothetical argument. The argument evaluation will help assess whether it will be nec-

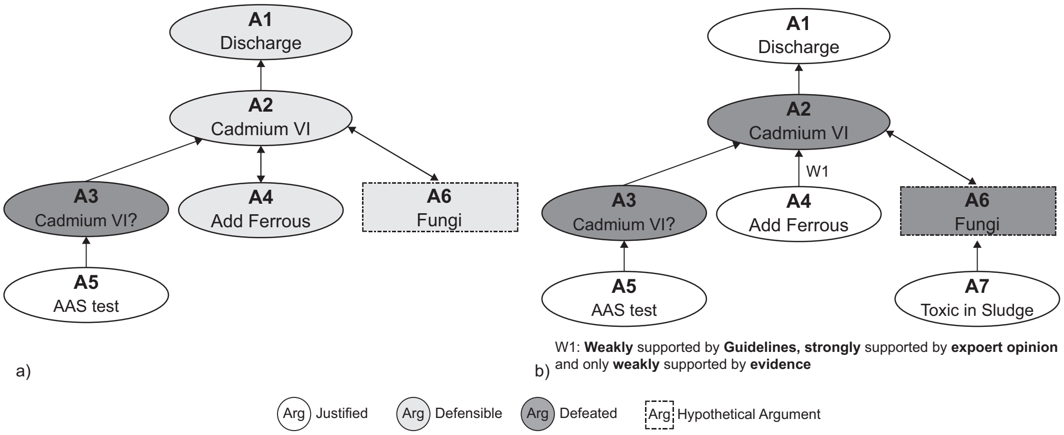


Figure 11.2: a) Arguments submitted by the participating agents. b) MA's proposed solution.

essary or not to check whether indeed `wwtp_cond(wwtp,fungi_bio_mass)` holds or not. In Figure 11.1 we can see how the deliberation continues with WTA endorsing argument $A4$ and later both participating agents informing that they have no more moves. In reply, the MA proposes a solution, depicted in Figure 11.2b, that if accepted by all parties, the deliberation concludes. According to the proposed solution the spill can safely be discharged provided the proposed complementary courses of actions are performed. Note that, on the one hand, MA has deemed argument $A4$ stronger than $A2$, though indicating that this preference is not conclusive. On the other hand, the MA submitted argument $A7$ defeating the hypothetical argument $A6$. Argument $A7$ instantiates scheme the above introduced scheme $AS6_{Ind\_pp\_2\_1}$ –*Toxic **Tx** will remain in sludge* – by grounding `Tx` with `cdIII` (Cadmium III). Namely, $A7$ indicates that even if there were fungi in the biomass the spill will still be unsafe since the resulting sludge will remain toxic because it will contain Cadmium III. Argument $A7$, that may be proposed by the CBRc, the DCK, or both knowledge resources, defeat argument $A6$ rendering unnecessary to check whether or not `wwtp_cond(wwtp,fungi_biomass)` holds. Finally, if the action is eventually performed, the outcome should be fed-back into $\mathbb{T}$ and then retained by the CBRc for later reuse.

11.2.2 Discussion

The above example illustrates the application of *ProCLAIM*in the environmental scenario, showing again, as with the transplant scenario, that with the provision of the ASR the model provides a setting for effective deliberation without making strong assumptions on the PAs' argumentation abilities. While more evolved examples and discussions regarding this scenario can be found in Aulinas' Thesis [39], we believe this section have provided insights on the model's applicability in diverse scenarios. As discussed at the beginning of this section, the development of the environmental scenario had significant impact in the definition of the *ProCLAIM* model. This scenario has motivated the need to provide a more structured account of the argument schemes that better define the scope of what can be argued about, what has become the current formalisation of the argument schemes presented in §6 and published in [16]. This scenario has further motivated the definition of a procedure for the construction of the ASR as discussed in §7.1. Hence, while in the early development of the model we focused on enabling deliberation among heterogeneous agents assuming no particular skills for argumentation, our latter efforts were devoted to enable developers who are not familiar with Argumentation to implement the *ProCLAIM* model.

While we believe to have made important progress in the elaboration of *ProCLAIM* the environmental scenario has outline some limitations that we need to address in future work. In general, the environmental scenario showed more requirements to account for coordination and planning. That is, a first step would be to incorporate time as a dimension in the deliberation over safety critical actions. This may include the extension of notion two transitions states: current state of affairs $\mathbf{R}$ and the effects of the actions $\mathbf{S}$. While this simplistic approach yield well with the transplant scenario, more complex scenarios as the environmental case study, may require concatenation of cause effect relations capturing the cascade of damaging effects an action may cause. Another aspect that should be addressed in future work is the integration of the cost value into the deliberation. Currently, the deliberation considers only safety as a measure for determining whether or not an action should be performed. However, in many scenarios, the cost involved in performing a course of action must also be accounted for as proposed solutions may be plausible in principle but economically prohibitive. To conclude, another important aspect that should be addressed in future work is the fact that in some occasions, undesirable side effects cannot be prevented, and yet the safety critical action must be performed. Such may be the case in the above example if none of the proposed solutions are satisfactory and thus, either pinpoint floc or toxic in sludge cannot be prevented. In those cases, the least of the worst cases should be chosen. Note that in this case the deliberation would aim to *minimise* the impact of an action, rather than simply classifying the action as either safe or unsafe. In that case, we would have to motivate whether in this new setting *ProCLAIM* is still the best approach.

11.3 Conclusions

In this chapter we have discussed our experience in implementing *ProCLAIM* in the two case studies, the transplant and the environmental scenarios. While the two explored scenario were described in detail in §3, motivating in each case the use of *ProCLAIM*, in this chapter we focus on the insights we gathered from the actual process of applying the proposed model in each of the two cases. Because the transplant and environmental scenario had very distinct development processes they provided us with different kind of feedback a which gave us a broad perspective on *ProCLAIM*'s applicability, its contributions and limitations.

Broadly speaking, *ProCLAIM* main aim is the proposal of a setting for agents to deliberate effectively and efficiently over safety critical actions. Initially we focused on the transplant scenario described in §3.1 where among the main aims where: *1)* partially automate the decision making over whether an available organ is viable or not for transplantation. *2)* Enable transplant professionals actively participate in the decision making without assuming they have any special skills, besides being experts in their domain area. And finally; *3)* accounting for the fact that decision makers may be in disagreement among each other and with established criteria, propose a solution for the deliberation in a way that the different view points are organised and relevant features for the decision making are highlighted. All three aims where central in our initial research and we believe to have made important progress in all three. Results of this work include: conceptualisation of the problem at hand [14, 13], design and formalisation [12], and finally implementation [11]. Indeed, one important validation of this work is the development of the ASPIC's large scale demonstrator as discussed in §10.1. We should also note that this work was always under supervision of transplant professionals and main concepts were discussed in transplant conferences [9, 10, 15].

While the main concern in this first scenario was the actual agents' deliberation, in the environmental scenario our concerned shifted towards the applicability of the *ProCLAIM* in a new scenario. This time, besides assuming no argumentation skills from the deliberating agents, our aim was to define *ProCLAIM* so that its implementation does not requires developers to have any prior knowledge of Argumentation. This effort has resulted in substantial improvement of *ProCLAIM*'s definition which better delimited the scope of application of the model and delivered procedures for the development of the Argument Scheme Repository, as illustrated in §7. On the one hand this scenario helped us develop a more mature formulation of *ProCLAIM*, particularly in the definition of the Argumentation layer as presented in §6 and published in [16], while on the other hand the development of the environmental scenario, presented both in a PhD Thesis in Environmental Engineering [39] and in [3], has its own value, principally as it shows the possibility that developers who are not familiar with Argumentation can actually implement the proposed model.

Central to the achievements of *ProCLAIM* and its contributions is the use of Argument Schemes and CQs, which enables to define and shape de agents' interaction. Over the last years a growing number of proposals appeal to the use of argument schemes for argumentation-based dialogues [182, 167, 196, 101, 35, 55]. These works generally assume the schemes proposed by Walton [231] or that proposed by Atkinson *et al.* [34]. While these schemes are undoubtedly of great value we believe they are too abstract for many real life applications, as we gathered from our experience. Indeed we used Atkinson *et al.* [34] to develop the more specialised schemes of *ProCLAIM*; and in turn, Atkinson *et al.*'s schemes are informed by Walton's proposed schemes [231], as we discuss in §2.1. However when we presented these schemes to either the potential end users or to the developers of a new scenario, it was not obvious for them how to instantiate the given schemes and thus it involved an important overhead. Clearly for some circumstances this overhead may be well justified, because the purpose may require argument schemes to cover all possible line of reasoning. This is clearly the case in some legal applications [102, 42, 239] or in many e-democracy applications [101, 62]. However, other decision-making application can benefit from narrowing down the lines of reasoning to only what is essential to the problem at hand, thus making a better use of the decision making context. The specialised schemes and CQ not only reduce the computational cost for the reasoners but they also focus the dialogue on what is essential, increasing the chances for a successful deliberation process. To the best of our knowledge we know of no other work that have proposed and explored the added value of scenario-specific schemes and CQs.

One of the main contributions of our work is in showing that the provision of the scenario specific schemes and CQs can facilitate relatively sophisticated deliberations in sensitive domains such as organ transplantation or industrial wastewater management, while reducing the complexity of argument construction to filling in simple templates (as shown both in §11.2.1 and in §7.2). While the main focus of our work has been on facilitating the agents' exchange of arguments, another contribution is *ProCLAIM* 's approach to argument validation and evaluation. The former is required to flexibly prevent spurious arguments from disrupting the deliberation. The latter is required to provide decision support as to whether the proposed action is safe or not, and is achieved by incorporating the relevant facts and actions into a tree of arguments that is evaluated on the basis of guidelines, expert opinion and past collected evidence. While, we believe to have made important progress in laying-out the basic principles for the argument evaluation and validation §8, future work should attempt to instantiate this framework with more realistic cases in terms of the three dimensions by which arguments are evaluated (*i.e.* evidence, consented knowledge and argument endorsement).

In the same line, in future work we intend to make a systematical study of real cases to nurture the ASR for both case studies. From our current experience, we already anticipate that we will have to provide better tools for the construction of the ASR so as to facilitate its maintenance when dealing with large number of schemes. More immediate tasks are the extensions of the circuit of schemes and CQs already discussed §6.2.5. One of the discussed extensions is of particular importance for the environmental scenario. That is, the possibility to question the degree of undesirability of goals, as it will allow to unlock the impasse that may be arrived to when all consequences arrive to undesirable goals. This is because, while in the transplant scenario if the transplant is deemed unsafe it is not performed, in the environmental scenario a hazardous spill has to be discharged, eventually and so decision makers should choose the course of actions believed to cause the least of the undesirable goals. The environmental scenario demand other extension to *ProCLAIM* that include the provision of a more descriptive language for actions so as to enable addressing more appropriately coordination and planning problems, currently not considered in *ProCLAIM*. Of course, prior to this, we should evaluate whether *ProCLAIM* is indeed appropriate for dealing with planning and coordination problems or an alternative approach should be undertaken.

With regard to related work, there are a number of works (*e.g.* [132], [56]) proposing deliberation, persuasion or negotiation models of argumentation for agent systems[9]. However, to the best of our knowledge, none of these works address the more practical aspects that enable actual implementation of the proposed models in scenarios more elaborated than simple illustrative examples. There are also a number of works applying multi-agent systems to safety critical domains; particularly the medical (see [116]) and the environmental

[9] See proceedings of the Argumentation in Multi-Agent Systems (ArgMAS) Workshop Series (http://www.mit.edu/∼irahwan/argmas/).

domains (see [73]). The most relevant that we are aware of is [160], in which a specific logic for argumentation is proposed for aiding medical doctors in their decision making in a multi-agent setting. However this work is primarily conceptual and does not address the agents' dialogical interaction or the roles of schemes and critical questions in guiding argument construction. Other works, such as [192] and [99] are related in the sense that a repository of Argument Schemes and CQ play a central role. The former is intended to assist users in argument diagramming, and the latter is intended to help (human) users construct a wide variety of arguments, improving their ability to protect their interests in (potential) dialogues, especially in the legal domain. A different approach is taken in the Magtalo system [195], in which a repository of fully instantiated arguments is used to help users express their position regarding a subject of public debate (in particular, whether *Identity cards are a bad idea*). Users can direct a dialogue among different artificial agent which allow them to explore the system's knowledge base following the natural flow of a dialogue. A user may agree with the different exposed arguments, may select an argument directly from the argument store and, as a last resource, type her own arguments in natural language (*i.e.* free text). This interaction is presented as a non intrusive mode for eliciting knowledge from users. This claim is based on what Walton and Krabbe call the *maieutic function* of dialogue [232]. Because users are immerse in a dialogue they do not feel they are being interrogated. We believe to have shown that *ProCLAIM* goes beyond this meiautic function by carefully defining the arguments' underlying structure (noted to be of value for this purpose [51]) and exploiting the context of application so that *1)* PAs need not be concerned about the argument construction, but only in filling in the blanks of templates presented in their domain expertise jargon; and *2)* the elicited knowledge is readily available for computational use, as opposed to embedded in a free text paragraph.

Chapter 12

Conclusions

In this Thesis we have present the *ProCLAIM* model, intended to extend existent distributed systems to support real-time deliberations among heterogeneous agents on whether or not a safety critical action can safely be performed. We illustrated the use of this model in two case studies: in a transplant scenario where the deliberation involved deciding over the safety of a human organ transplant, and an environmental scenario where the deliberation involved deciding the safety of an industrial wastewater spill. Let us now enumerate our main contributions in this Thesis. We conclude in §12.2 with a critical analysis of *ProCLAIM*'s limitations and envision possible lines of future work.

12.1 Main Original Contributions

One of the key reasons for the growing interest in argumentation theory has been the recognition that it provides an abstraction of formal logical approaches to non-monotonic reasoning, inherently capturing the dialectical *process* of argument and counter-argument in a manner that is intuitive and easily understandable by human reasoners.

Thus, one of the key added values of argumentation theory is that it has the potential to bridge formal logic-based models of reasoning and more informal human reasoning, in a manner that naturally accounts for the dialogical processes whereby knowledge is acquired and reasoned with, and thus providing for formal logic-based models of rationality to normatively guide human reasoning processes.

The joint use of schemes and CQs on the one hand and Dung framework [80] on the other, is consolidating as the main approach to embody the above value proposition. We believe that the major contribution of our work is that it provides one of the most sophisticated practical realisations of the linkage between the schemes and CQs approach (initiated by the informal logic/philosophical community) and the formal abstract argumentation approach of Dung that provides rational criteria for evaluating arguments.

In achieving this we have made the following contributions:

1. Defined dialogical framework addressing collaborative decision making regarding whether a safety-critical action can be performed, in which:

- Basic relevant factors for the decision making can automatically be elicited from the artificial agents in the form of a deliberation.
- Domain experts, can enter the deliberation at any stage in order to actively participate in the decision making in real-time.
- A solution for the deliberation is proposed accounting for domain guidelines, expert opinion and evidence.
- Proposed solutions are justifications to *why* a critical action can safely be performed or not, hence avoiding black box recommendations.
- Deliberations, along with their outcomes, are retained and later reused to solve future similar cases on an evidential basis.

2. Defined a set of reasoning patterns, represented in terms of argument schemes and critical questions, intended to automatise deliberations on whether a proposed action can safely be performed, and shown how these schemes and CQs can be specialised for applications and used to automate deliberations. This involves:

    - A circuit of schemes and CQs of an intermediate level of abstraction, that while tailored for reasoning over safety-critical actions are application-independent.
    - A procedure, along with a prototype application, intended to facilitate the further specialisation of these schemes into more specific schemes specialised for the target application.
    - Application specific schemes that enables real-time deliberations among heterogeneous (human and artificial) agents in safety-critical domains such as organ transplantation of wastewater management.

3. Developed a Case Based Reasoning component that reuses past deliberations in order to help resolve similar target cases on an evidential basis.

4. Implemented a large scale demonstrator developed for the FP6-European Project ASPIC that serves as an advanced proof of concept of *ProCLAIM*.

5. Established the generality of *ProCLAIM* by applying it to an alternative scenario, developed primarily by environmental engineers not familiar with Argumentation.

In summary, our main contribution has been realised through *ProCLAIM*'s automation of deliberation dialogues between agents (human or software) over organ transplant decisions or environmental decisions, in a manner which is structured and orderly, and which elicits all the information needed to make such decisions jointly and rationally, even when this information is possessed only by some of the participating agents. Key to realising our aim of using formal argumentation models to normatively guide human reasoning, has been the development of a dialogical model that does not require the participants to have specialised knowledge of argumentation theory. This is achieved by the framework's embedding of domain expertise (*e.g.* medical or environmental) in a natural way using scenario-specific argumentation schemes.

12.2 Limitations and Future Work

While important progress has been made both in the definition of the *ProCLAIM*model as in the development of the two case studies, there are a number of limitations and open questions that we believe are worth addressing in future work. Broadly speaking there are three areas of research that we believe are particularly worth strengthening in our work. These relate to: *1)* the argument schemes' expressivity and exhaustivity: *2)* argument evaluation and solution proposal and; *3)* human-computer interfacing features. All three aspects are relevant from a general theoretical perspective and are required to further develop the two case studies.

1. Regarding the **argument schemes' expressivity and exhaustivity** :

    - As we have discussed in §6.2.5, *ProCLAIM*'s defined circuit of schemes and critical questions needs to be further extended to account for a number of limitations that have particularly emerged when implementing the environmental scenario. However, beyond these more technical limitations, lies the inherent problem of **exhaustivity** with argument schemes. That is, how can we be certain that all fundamental questions are addressed? While we are sceptical about the possibility to prove exhaustivity, we nonetheless believe that this problem has to be addressed in a somewhat more rigorous way, not only form a theoretical perspective, but mainly from a practical viewpoint. A better understanding of this limitation may help produce more sound applications.
    - Complementary to the above task, and a necessary step forward in the development of both use cases, is to perform a more systematic analysis of reported cases both in the transplant and in the environmental scenarios. This, in turn will help us enrich the ASR in order to perform more realistic simulations with the potential end users.
    - *ProCLAIM*'s deliberations are inherently bounded by the schemes encoded in the ASR. Beyond the problem of exhaustivity, this implies that *ProCLAIM* applications' are intrinsically limited to address situations which can be anticipated. The particularities of the context and the proposed solutions may be novel, but to some extent, the overall situation must be foreseen beforehand. This of course, reduces the possibilities of using *ProCLAIM* applications in particularly singular situations where the protocols for safety are ill defined, such as in natural catastrophes which carry unexpected situations. An interesting question worth exploring is, to what extent such singular situations may benefit from the use of argument schemes or Argumentation in general. We believe that any proposal will have to start by understanding the context in which the decision making takes place, including number and role of stakeholders, their assumed skills and knowledge, the time constraints for the decision making, as well as understanding the nature of the justifications (*e.g.* scientific or political).

2. *ProCLAIM* is intended to lay out the available information and knowledge at hand in order to support the decision makers in their task. We acknowledge that the preference

assignment by the different knowledge resources is not conclusive but informative. In future work, however, we intend to augment the range of knowledge resources used in the preference assignment, in order to further increase the confidence that can be ascribed to these assignments. In particular, we could consider statistical and other numerical medical data, elicited, for example, from clinical trials and meta-reviews.

3. As we have discussed in §2.5 Argumentation has shown to yield particularly well with graphical visualisations, allowing for the presentation of complex reasoning in a clear and unambiguous fashion, promoting focused and productive discussions. Indeed Argumentation has an important advantage in that many of its formalisations are easily visualised through simple and clear graphs and diagrams. However, while this provides a privileged starting point for producing software interfaces, we believe that for more advanced applications we should separate more clearly the systems' back-end representations from their front-end representations. Regarding *ProCLAIM*, a first step is to propose workflows that rather than mimicking the underlying formalisation exploit the context and shared knowledge in order to simplify the human-computer interaction. For instance, domain experts may not always require a full graphical representation of the interacting arguments in order to understand the problem at hand. On occasion a simple list of facts and actions may suffice. Hence, at the front-end, part of the deliberation may involve only clicking on certain facts and actions displayed on the screen.

Bibliography

Author's Bibliography

[1] M. Aulinas, P. Tolchinsky, C. Turon, M. Poch, and U. Cortés. An argument-based approach to deal with wastewater discharges. In *Proceeding of the 2007 conference on Artificial Intelligence Research and Development*, pages 400–407. IOS Press, 2007.

[2] M. Aulinas, P. Tolchinsky, C. Turon, M. Poch, and U. Cortés. Is my spill environmentally safe? towards an integrated management of wastewater in a river basin using agents that can argue. In *7th International IWA Symposium on Systems Analysis and Integrated Assessment in Water Management (WATERMATEX 2007)*, 2007.

[3] M. Aulinas, P. Tolchinsky, C. Turon, M. Poch, and U. Cortés. Argumentation-based framework for industrial wastewater discharges management. *Engineering Applications of Artificial Intelligence*, 25(2):317–325, 2012. [doi: 10.1016/j.engappai.2011.09.016].

[4] S. Modgil, P. Tolchinsky, and U. Cortés. Towards formalising agent argumentation over the viability of human organs for transplantation. In *Advances in Artificial Intelligence: 4th Mexican International Conference on Artificial Intelligence (MICAI 05)*, pages 928–938, Monterrey, Mexico, November, 2005.

[5] P. Tolchinsky, K. Atkinson, P. McBurney, S. Modgil, and U. Cortés. Agents Deliberating Over Action Proposals Using the ProCLAIM Model. *Proc. Fifth Intern. Central and Eastern European Conf. on Multi-Agent Systems (CEEMAS 2007), LNAI, Berlin, Germany*, 2007.

[6] P. Tolchinsky, M. Aulinas, U. Cortés, and M. Poch. Deliberation Over the Safety of Industrial Wastewater Discharges into Wastewater Treatment Plants. In *Advanced Agent-Based Environmental Management Systems*, Whitestein Series in Software Agent Technologies and Autonomic Computing, chapter 2, pages 37–60. Birkhuser Basel. Springer, 2009.

[7] P. Tolchinsky and U. Cortés. Argument Schemes and Critical Questions for deciding upon the Viability of a Human Organ for transplantation. Technical report, Technical University Of Catalonia, 2005.

[8] P. Tolchinsky and U. Cortés. Arguing agents for fairness in the allocation of human organs for transplantation. In *4th Workshop on Agents Applied in Health Care (ECAI-06)*, 2006.

[9] P. Tolchinsky, U. Cortés, F. Caballero, and A. López-Navidad. CARREL$^+$, intelligent electronic support for distribution and allocation of organs for transplantation. In *Transplant International*, page 343, 2007.

[10] P. Tolchinsky, U. Cortés, F. Caballero, and A. López-Navidad. A novel organ selection procedure that uses artificial intelligence to increase the organ pool for transplant. In *Transplant International*, page 343, 2007.

[11] P. Tolchinsky, U. Cortés, and D. Grecu. Argumentation-Based Agents to Increase Human Organ Availability for Transplant. In *Agent Technology and e-Health*, Whitestein Series in Software Agent Technologies and Autonomic Computing, chapter 3, pages 65–93. Birkhuser Basel. Springer, 2008.

[12] P. Tolchinsky, U. Cortés, S. Modgil, F. Caballero, and A. López-Navidad. Increasing human-organ transplant availability: Argumentation-based agent deliberation. *IEEE Intelligent Systems*, 21(6):30–37, 2006. [doi:10.1109/MIS.2006.116].

[13] P. Tolchinsky, U. Cortés, J. C. Nieves, F. Caballero, and A. López-Navidad. Using arguing agents to increase the human organ pool for transplantation. In *3rd Workshop on Agents Applied in Health Care (IJCAI-05)*, 2005.

[14] P. Tolchinsky, U. Cortés, J. C. Nieves, and J. Vázquez-Salceda. Extending CARREL's Architecture for Agents to Argue over the Viability of a Human Organ. Technical report, Technical University Of Catalonia, 2005.

[15] P. Tolchinsky, A. López-Navidad, F. Caballero, and U. Cortés. CARREL$^+$, Soporte Inteligente para la Distribucion y Asignacion de Organos para trasplante. In *En XXII Reunión Nacional de Coordinadores de Trasplante, Palma de Mallorca*, 2007.

[16] P. Tolchinsky, S. Modgil, K. Atkinson, P. McBurney, and U. Cortés. Deliberation dialogues for reasoning about safety critical actions. *Autonomous Agents and Multi-Agent Systems (JAAMAS)*, pages 1–51, 2011. [doi:10.1007/s10458-011-9174-5].

[17] P. Tolchinsky, S. Modgil, and U. Cortés. Argument schemes and critical questions for heterogeneous agents to argue over the viability of a human organ. In *AAAI 2006 Spring Symposium Series; Argumentation for Consumers of Healthcare*, pages 105-111, AAAI Press, 2006.

[18] P. Tolchinsky, S. Modgil, U. Cortés, and M. Sànchez-Marrè. CBR and Argument Schemes for Collaborative Decision Making. In P. E. Dunne and T. J. M. Bench-Capon, editors, *Conference on Computational Models of Argument (COMMA 06)*, volume 144 of *Frontiers in Artificial Intelligence and Aplications*, pages 71–82. IOS Press, September 2006.

General Bibliography

[19] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Commun.*, 7(1):39–59, 1994.

[20] Adelard. ASCAD: The Adelard Safety Case Development Manual, 1998. ISBN 0 9533771 0 5.

[21] A. Aldea, B. López, A. Moreno, D. Riaño, and A. Valls. A multi-agent system for organ transplant co-ordination. *Artificial Intelligence in Medicine*, pages 413–416, 2001.

[22] V. Aleven. *Teaching case-based argumentation through a model and examples*. PhD thesis, 1997.

[23] JW Alexander and JC Zola. Expanding the donor pool: use of marginal donors for solid organ transplantation. *Clinical transplantation*, 10(1 Pt 1):1, 1996.

[24] L. Amgoud and C. Cayrol. On the acceptability of arguments in preference-based argumentation framework. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 1–7, 1998.

[25] L. Amgoud and C. Cayrol. Inferring from inconsistency in preference-based argumentation frameworks. *International Journal of Automated Reasoning*, Volume 29 (2):125–169, 2002.

[26] L. Amgoud, N. Maudet, and S. Parsons. Arguments, dialogue, and negotiation. In *Proceedings of the 14th European Conference on Artificial Intelligence*, pages 338–342, 2000.

[27] L. Amgoud, N. Maudet, and S. Parsons. Modelling dialogues using argumentation. In *Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-00)*, pages 31–38, Boston, MA, 2000.

[28] Aristotle. *Topics*. Clarendon Press, Oxford, UK, 1928.

[29] K. D. Ashley. Reasoning with cases and hypotheticals in hypo. *International Journal of Man-Machine Studies*, 34(6):753–796, 1991.

[30] K.D. Ashley. Hypothesis Formation and Testing in Legal Argument. *Invited paper. Inst. de Investig. Jurídicas 2d Intl Meet. on AI and Law, UNAM, Mexico City. April*, 2006.

[31] ASPIC. Deliverable d4.6: Detailed designs and test sets of aspic components. Technical report, ASPIC, 2007. http://www.argumentation.org/Public Deliverables.htm.

[32] K. Atkinson. *What should we do?: Computational representation of persuasive argument in practical reasoning*. PhD thesis, Ph. D. Thesis, Department of Computer Sciences, University of Liverpool, UK, 2005.

[33] K. Atkinson, T. Bench-Capon, and P. McBurney. A Dialogue Game Protocol for Multi-agent Argument over Proposals for Action. *Argumentation in Multi-Agent Systems: First International Workshop, ArgMAS 2004, New York, NY, USA, July 19, 2004: Revised Selected and Invited Papers*, 2005.

[34] K. Atkinson, T. Bench-Capon, and P. McBurney. Computational representation of practical argument. *Synthese*, 152(2):157–206, 2006.

[35] K. Atkinson, T. Bench-Capon, and S. Modgil. Argumentation for decision support. In *Database and Expert Systems Applications*, pages 822–831. Springer, 2006.

[36] K. Atkinson, T. J. M. Bench-Capon, and P. McBurney. Implementation of a Dialogue Game for Persuasion over Action. Technical Report ULCS-04-005, University of Liverpool, 2004.

[37] K. Atkinson, T. J. M. Bench-Capon, and P. McBurney. Arguing about cases as practical reasoning. In *ICAIL*, pages 35–44, 2005.

[38] K. M. Atkinson, T. J. M. Bench-Capon, and P. McBurney. A dialogue game protocol for multi-agent argument for proposals over action. In *Proc. First International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2004)*, 2004.

[39] M. Aulinas. *Management of industrial wastewater discharges in river basins through agents' argumentation*. PhD thesis, Laboratory of Chemical and Environmental Engineering (LEQUIA). University of Girona, 2009.

[40] M. Aulines, J.C. Nieves, M. Poch, and U. Cortés. Supporting Decision Making in Urban Wastewater Systems Using a Knowledge-Based Approach. *Environmental Modelling and Software*, 2011.

[41] J. L. Austin. How to do things with words. In *Clarendon Press, Oxford Uk*, 1962.

[42] T. Bench-Capon and H. Prakken. Using argument schemes for hypothetical reasoning in law. *Artificial Intelligence and Law*, pages 1–22, 2010.

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

[44] T. J. M. Bench-Capon, D. Lowes, and A. M. McEnery. Argument-based explanation of logic programs. *Knowledge Based Systems*, 4(3):177–183, 1991.

[45] T. J. M. Bench-Capon and G. Sartor. A model of legal reasoning with cases incorporating theories and values. *Artificial Intelligence*, 150:97–143, 2003.

[46] T.J.M. Bench-Capon. Agreeing to differ: modelling persuasive dialogue between parties with different values. *INFORMAL LOGIC-WINDSOR ONTARIO-*, 22:231–246, 2002.

[47] T.J.M. Bench-Capon. Value based argumentation frameworks. In *Non Monotonic Reasoning*, pages 444–453, 2002.

[48] T.J.M. Bench-Capon. Persuasion in practical argument using value based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–48, 2003.

[49] T.J.M. Bench-Capon and P. E. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171(10-15):619–641, 2007.

[50] T.J.M. Bench-Capon and P.E. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171(10-15):619–641, 2007.

[51] J. Bentahar, B. Moulin, and M. Bélanger. A taxonomy of argumentation models used for knowledge representation. *Artificial Intelligence Review*, 33(3):211–259, 2010.

[52] J. Bentahar, B. Moulin, and B. Chaib-draa. Specifying and implementing a persuasion dialogue game using commitments and arguments. *Argumentation in multi-agent systems*, pages 130–148, 2005.

[53] P. Bishop and R. Bloomfield. A methodology for safety case development. In *Safety-Critical Systems Symposium, Birmingham, UK*. Citeseer, 1998.

[54] A.K. Biswas. Integrated water resources management: a reassessment. *Water International*, 29(2):248–256, 2004.

[55] E. Black and K. Atkinson. Dialogues that account for different perspectives in collaborative argumentation. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 867–874. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[56] E. Black and K. Atkinson. Agreeing what to do. *ArgMAS 2010*, page 1, 2010.

[57] E. Black and A. Hunter. An inquiry dialogue system. *Autonomous Agents and Multi-Agent Systems*, 19(2):173–209, 2009.

[58] A. Bondarenko, P.M. Dung, R.A. Kowalski, and F. Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93:63–101, 1997.

[59] D. Bouyssou and P. Vincke. Ranking alternatives on the basis of preference relations: a progress report with special emphasis on outranking relations. *Journal of Multi-Criteria Decision Analysis*, 6(2):77–85, 1997.

[60] M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artif. Intell.*, 171(5-6):286–310, 2007.

[61] C.S. Carr. Using computer supported argument visualization to teach legal argumentation. *Visualizing argumentation: Software tools for collaborative and educational sense-making*, pages 75–96, 2003.

[62] D. Cartwright and K. Atkinson. Using computational argumentation to support e-participation. *Intelligent Systems, IEEE*, 24(5):42–52, 2009.

[63] A. Cawsey, F. Grasso, and R. Jones. A conversational model for health promotion on the World Wide Web. *Artificial Intelligence in Medicine*, pages 379–388, 1999.

[64] M. Chalamish and S. Kraus. AutoMed: an automated mediator for bilateral negotiations under time constraints. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, page 248. ACM, 2007.

[65] BM Chavers, JJ Snyder, MA Skeans, ED Weinhandl, and BL Kasiske. Racial disparity trends for graft failure in the US pediatric kidney transplant population, 1980–2004. *American Journal of Transplantation*, 9(3):543–549, 2009.

[66] C. Chesñevar, J. McGinnis, S. Modgil, I. Rahwan, C. Reed, G. Simari, M. South, G. Vreeswijk, and S. Willmott. Towards an argument interchange format. *Knowl. Eng. Rev.*, 21(4):293–316, 2006.

[67] C.I. Chesnevar, G.R. Simari, and L. Godo. Computing dialectical trees efficiently in possibilistic defeasible logic programming. *Logic Programming and Nonmonotonic Reasoning*, pages 158–171, 2005.

[68] R.T. Chung, S. Feng, and F.L. Delmonico. Approach to the management of allograft recipients following the detection of hepatitis B virus in the prospective organ donor. *American Journal of Transplantation*, 1(2):185–191, 2001.

[69] J. Comas, I. Rodríguez-Roda, KV Gernaey, C. Rosen, U. Jeppsson, and M. Poch. Risk assessment modelling of microbiology-related solids separation problems in activated sludge systems. *Environmental Modelling and Software*, 23(10-11):1250–1261, 2008.

[70] J. Comas, I. Rodríguez-Roda, M. Sànchez-Marrè, U. Cortés, A. Freixó, J. Arráez, and M. Poch. A knowledge-based approach to the deflocculation problem: integrating on-line, off-line, and heuristic information. *Water Research*, 37(10):2377–2387, 2003.

[71] European Community. Water framework directive (2000/60/ec). http://ec.europa.eu/environment/water/water-framework/index_en.html (last checking: 21/06/2010), 2000.

[72] U. Cortés, M. Martínez, J. Comas, M. Sànchez-Marrè, M. Poch, and I. Rodríguez-Roda. A conceptual model to facilitate knowledge sharing for bulking solving in wastewater treatment plants. *AI Communications*, 16(4):279–289, 2003.

[73] U. Cortés and M. Poch, editors. *Advanced Agent-Based Environmental Management Systems*, Whitestein Series in Software Agent Technologies and Autonomic Computing. Birkhuser Basel book, Springer, 2009.

[74] U. Cortés, J. Vázquez-Salceda, A. López-Navidad, and F. Caballero. UCTx: a multi-agent approach to model a transplant coordination unit. *Journal of Applied Intelligence*, 20(1):59–70, 2004.

[75] M.R. Costanzo, A. Dipchand, R. Starling, A. Anderson, M. Chan, S. Desai, S. Fedson, P. Fisher, G. Gonzales-Stawinski, L. Martinelli, et al. The International Society of Heart and Lung Transplantation Guidelines for the care of heart transplant recipients. *The Journal of Heart and Lung Transplantation*, 29(8):914–956, 2010.

[76] AS Coulson, DW Glasspool, J. Fox, and J. Emery. RAGs: A novel approach to computerized genetic risk assessment and decision support from pedigrees. *Methods of Information in Medicine*, 40(4):315–322, 2001.

[77] F.L. Delmonico, B. Domínguez-Gil, R. Matesanz, and L. Noel. A call for government accountability to achieve national self-sufficiency in organ donation and transplantation. *The Lancet*, 378(9800):1414–1418, 2011.

[78] Y. Dimopoulos, P. Moraitis, and L. Amgoud. Theoretical and computational properties of preference-based argumentation. *PAIS 2008*, page 463, 2008.

[79] J. Doyle. Prospects for preferences. *Computational Intelligence*, 20(2):111–136, 2004.

[80] P. M. 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.

[81] P.M. Dung, P. Mancarella, and F. Toni. Computing ideal sceptical argumentation. *Artificial Intelligence*, 171(10-15):642–674, 2007.

[82] P.E. Dunne. Computational properties of argument systems satisfying graph-theoretic constraints. *Artificial Intelligence*, 171(10-15):701–729, 2007.

[83] L. Emmet and G. Cleland. Graphical notations, narratives and persuasion: a pliant systems approach to hypertext tool design. In *Proceedings of the thirteenth ACM conference on Hypertext and hypermedia*, page 64. ACM, 2002.

[84] J. Eriksson Lundström, J. Fischer Nilsson, and A. Hamfelt. Legal rules and argumentation in a metalogic framework. In *Proceeding of the 2007 conference on Legal Knowledge and Information Systems: JURIX 2007: The Twentieth Annual Conference*, pages 39–48. IOS Press, 2007.

[85] T. Finin, Y. Labrou, and J. Mayfield. KQML as an agent communication language. In *J. Bradshaw, ed. Software agents, MIT Press, Cambridge*, 1995.

[86] F. FIPA. Communicative Act Library Specification.

[87] J. Fox and S. Das. *Safe and Sound. Artificial Intelligence in Hazardous Applications*. AAAI Press, The MIT Press, 2000.

[88] J. Fox, P. Hammond, D. Elsdon, T. Khabaza, A. Montgomery, I. Khabaza, R. Mantil, R. Susskind, C. Tapper, G. Swaffield, et al. Expert systems for safety-critical applications: theory, technology and applications. In *Knowledge-Based Systems for Safety Critical Applications, IEE Colloquium on*, page 5. IET, 2002.

[89] J. Fox, N. Johns, C. Lyons, A. Rahmanzadeh, R. Thomson, and P. Wilson. PROforma: a general technology for clinical decision support systems. *Computer Methods and Programs in Biomedicine*, 54(1-2):59–67, 1997.

[90] J. Fox, P. Krause, and S. Ambler. Arguments, contradictions and practical reasoning. In *ECAI '92: Proceedings of the 10th European conference on Artificial intelligence*, pages 623–627, New York, NY, USA, 1992. John Wiley & Sons, Inc.

[91] BR Gaines, DH Norrie, and AZ Lapsley. Mediator: an intelligent information system supporting the virtual manufacturing enterprise. In *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on*, volume 1, pages 964–969. IEEE, 2002.

[92] A.J. García and G.R. Simari. Defeasible logic programming: an argumentative approach. *Theory and Practice of Logic Programming*, 4(1):95–138, 2004.

[93] D. Garcia, J.C. Nieves, and Cortés. Reasoning about Actions for the Management of Urban Wastewater Systems using a Causal Logic. In *8th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS'10)*, volume 70 of *Advances in Intelligence and Soft Computing*, pages 247–257. Springer, 2010.

[94] M. P. Georgeff and A. L. Lansky. Reactive reasoning and planning. In *AAAI*, pages 677–682, 1987.

[95] R. Girle, D. L. Hitchcock, P. McBurney, and B. Verheij. *Argumentation machines: New frontiers in argument and computation*, chapter Decision support for practical reasoning: A theoretical and computational perspective, page 5584. Kluwer Academic Publishers, 2004.

[96] D.W. Glasspool and J. Fox. REACTa decision-support system for medical planning. In *Proceedings of the AMIA Symposium*, page 911. American Medical Informatics Association, 2001.

[97] D.W. Glasspool, J. Fox, F.D. Castillo, and V.E.L. Monaghan. Interactive decision support for medical planning. In *Artificial Intelligence in Medicine: 9th Conference on Artificial Intelligence in Medicine in Europe, AIME 2003, Protaras, Cyprus, October 18-22, 2003 Proceedings*, page 335. Springer-Verlag New York Inc, 2003.

[98] T. F. Gordon and N. Karacapilidis. The zeno argumentation framework. In *Proceedings of the sixth international conference on Artificial intelligence and law*, pages 10 – 18. ACM Press, 1997.

[99] T. F. Gordon, H. Prakken, and D. Walton. The carneades model of argument and burden of proof. *Artif. Intell.*, 171(10-15):875–896, 2007.

[100] T.F. Gordon. An overview of the carneades argumentation support system. *Dialectics, Dialogue and Argumentation. An Examination of Douglas Walton's Theories of Reasoning*, pages 145–156, 2010.

[101] T.F. Gordon, H. Prakken, and D. Walton. The Carneades model of argument and burden of proof. *Artificial Intelligence*, 171(10-15):875–896, 2007.

[102] T.F. Gordon and D. Walton. Legal reasoning with argumentation schemes. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pages 137–146. ACM, 2009.

[103] N. Gorogiannis, A. Hunter, V. Patkar, and M. Williams. Argumentation about Treatment Efficacy. *Knowledge Representation for Health-Care. Data, Processes and Guidelines*, pages 169–179, 2010.

[104] F. Grasso, A. Cawsey, and R. Jones. Dialectical argumentation to solve conflicts in advice giving: a case study in the promotion of healthy nutrition. *International Journal of Human-Computer Studies*, 53(6):1077–1115, 2000.

[105] N.A. Gray et al. Current status of the total artificial heart. *American heart journal*, 152(1):4–10, 2006.

[106] W. Grennan. *Informal Logic*. McGill-Queens University Press, 1997.

[107] T. Grondsma. River Basin Management, Pollution and the Bargaining Arena. In H. Laikari, editor, *River BasinManagement*, volume V. Advances in Water Pollution Control, IAWPRC, pages 419–425. Pergamon Press, 1989.

[108] J. Habermas. *The Theory of Communicative Action: Volume 1: Reason and the Rationalization of Society*. Heinemann, London, UK, 1984. (Translation by T. McCarthy of: *Theorie des Kommunikativen Handelns, Band I, Handlungsrationalitat und gesellschaftliche Rationalisierung.* Suhrkamp, Frankfurt, Germany. 1981.).

[109] C.B. Haley, J.D. Moffett, R. Laney, and B. Nuseibeh. Arguing Security: Validating Security Requirements Using Structured Argumentation. In *Proceedings of the Third Symposium on Requirements Engineering for Information Security (SREIS'05), co-located with the 13th International Requirements Engineering Conference (RE'05*. Citeseer, 2005.

[110] C. L. Hamblin. *Fallacies*. Methuen, London, UK, 1970.

[111] A. C. Hastings. *A Reformulation of the Modes of Reasoning in Argumentation*. PhD thesis, Evanston, Illinois, 1963.

[112] M. Henze, R. Dupont, P. Grau, and A. de la Sota. Rising sludge in secondary settlers due to denitrification. *Water Research*, 27(2):231–236, 1993.

[113] D. Hitchcock, P. McBurney, and S. Parsons. A framework for deliberation dialogues. In *Proceedings of the Fourth Biennial Conference of the Ontario Society for the Study of Argumentation (OSSA 2001), Windsor, Ontario, Canada*, volume 2, page 275, 2001.

[114] A. Hron, F.W. Hesse, U. Cress, and C. Giovis. Implicit and explicit dialogue structuring in virtual learning groups. *British Journal of Educational Psychology*, 70(1):53–64, 2000.

[115] J. Hulstijn. Dialogue models for inquiry and transaction. *PhD, University of Twente*, 2000.

[116] D. Isern, D. Sánchez, and A. Moreno. Agents applied in health care: A review. *International Journal of Medical Informatics*, 2010.

[117] D. Jenkins, M.G. Richard, and G.T. Daigger. *Manual on the Causes and Control of Activated Sludge Bulking, Foaming, and Other Solids Separation Problems*. third ed., IWA Publishing, London, UK, 2003.

[118] PN Judson, J. Fox, and PJ Krause. Using new reasoning technology in chemical information systems. *J. Chem. Inf. Comput. Sci*, 36(4):621–624, 1996.

[119] A. Kakas and P. Moraitis. Argumentation based decision making for autonomous agents. In *Proc. Second international joint conference on Autonomous agents and multiagent systems*, pages 883–890. ACM Press, 2003.

[120] N. Karacapilidis and D. Papadias. Computer supported argumentation and collaborative decision making: The hermes system. *Information Systems*, 26(4):259–277, 2001.

[121] N. Karacapilidis and D. Papadias. Computer supported argumentation and collaborative decision making: the HERMES system. *Information Systems*, 26(4):259–277, 2001.

[122] N. Karacapilidis, B. Trousse, and D. Papadias. Using case-based reasoning for argumentation with multiple viewpoints. In *ICCBR 1997*.

[123] M. Karlins and H.I. Abelson. *Persuasion: How opinions and attitudes are changed*. Springer, 1970.

[124] LH Katz, M. Paul, DG Guy, and R. Tur-Kaspa. Prevention of recurrent hepatitis B virus infection after liver transplantation: hepatitis B immunoglobulin, antiviral drugs, or both? Systematic review and meta-analysis. *Transplant Infectious Disease*, 12(4):292–308, 2010.

[125] J. Katzav and C.A. Reed. *Argumentation*, volume 18, chapter On Argumentation Schemes and the Natural Classification of Arguments, pages 239–259(21). Springer, 2004.

[126] H.M. Kauffman, M.A. McBride, and F.L. Delmonico. First Report of the United Network for Organ Sharing Transplant Tumor Registry: Donors With A History of Cancer1. *Transplantation*, 70(12):1747, 2000.

[127] T. Kelly. Arguing safety – a systematic approach to managing safety cases, 1998.

[128] T.P. Kelly and J. McDermid. Safety case construction and reuse using patterns. In *16th International Conference on Computer Safety and Reliability (SAFECOMP'97), York*. Citeseer, 1997.

[129] M. Kienpointner. Towards a typology of argument schemes. In *ISSA 1986*. Amsterdam University Press., 1986.

[130] P. A. Kirschner, S. J. Buckingham Shum, and C. S. Carr, editors. *Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making*. Springer-Verlag, 2003. ISBN 1-85233-6641-1.

[131] D.L. Kleinman and D. Serfaty. Team performance assessment in distributed decision making. In *Proceedings of the Symposium on Interactive Networked Simulation for Training*, pages 22–27, 1989.

[132] E.M. Kok, J.J.C. Meyer, H. Prakken, and G.A.W. Vreeswijk. A Formal Argumentation Framework for Deliberation Dialogues. *ArgMAS 2010*, page 73, 2010.

[133] J. Kolodner. *Case-based reasoning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[134] J. L. Kolodner. Indexing and retrieval strategies for natural language fact retrieval. *ACM Trans. Database Syst.*, 8(3):434–464, 1983.

[135] P. Krause, S. Ambler, M. Elvang-Gøransson, and J. Fox. A logic of argumentation for reasoning under uncertainty. *Computational Intelligence*, 11(1):113–131, 1995.

[136] P. Krause, J. Fox, and P. Judson. An argumentation based approach to risk assessment. *IMA Journal of Mathematics Applied in Business and Industry*, 5:249–263, 1994.

[137] P. Krause, P. Judson, and M. Patel. Qualitative risk assessment fulfills a need. In A. Hunter and S. Parsons, editors, *Appplications of Uncertainty Formalisms*. Springer Verlag, Berlin, 1998.

[138] W. Kunz and H. W. J. Rittel. Issues as elements of information systems. Working Paper No 131, 1970.

[139] Y. Labrou, T. Finin, and Y. Peng. Agent communication languages: The current landscape. *IEEE Intelligent systems*, pages 45–52, 1999.

[140] M. Lebowitz. Memory-based parsing. *Artif. Intell.*, 21(4):363–404, 1983.

[141] J. Lee. SIBYL: A qualitative decision management system. In *Artificial intelligence at MIT expanding frontiers*, page 133. MIT Press, 1991.

[142] S. Levinson. *Pragmatics*. Cambridge University Press, Cambridge, 1983.

[143] A. López-Navidad and F. Caballero. Extended criteria for organ acceptance. Strategies for achieving organ safety and for increasing organ pool. *Clinical transplantation*, 17(4):308–324, 2003.

[144] P. Lorenzen and K. Lorenz. *Dialogische logik*. Wissenschaftliche Buchgesellschaft Darmstadt, Germany, 1978.

[145] J. Lu and S.P. Lajoie. Supporting medical decision making with argumentation tools. *Contemporary Educational Psychology*, 33(3):425–442, 2008.

[146] D. Marelli, H. Laks, S. Bresson, A. Ardehali, J. Bresson, F. Esmailian, M. Plunkett, J. Moriguchi, and J. Kobashigawa. Results after transplantation using donor hearts with preexisting coronary artery disease. *The Journal of Thoracic and Cardiovascular Surgery*, 126(3):821–825, 2003.

[147] W. S. Mark. Case-based reasoning for autoclave management. In *Proc. of a Workshop on Case-Based Reasoning*, pages 176–180, Pensacola Beach, FL, 1989.

[148] M. Mbarki, J. Bentahar, and B. Moulin. Specification and complexity of strategic-based reasoning using argumentation. *Argumentation in multi-agent systems*, pages 142–160, 2007.

[149] P. McBurney, R. Van Eijk, S. Parsons, and L. Amgoud. A dialogue-game protocol for agent purchase negotiations. *Journal of Autonomous Agents and Multi-Agent Systems. Special Issue: Argumentation in Inter-Agent Communication*, 7(3):235–273, 2003.

[150] P. McBurney, D. Hitchcock, and S. Parsons. The eightfold way of deliberation dialogue. *International Journal of Intelligent Systems*, 22(1):95–132, 2007.

[151] P. McBurney and S. Parsons. Risk agoras: dialectical argumentation for scientific reasoning. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, 2000.

[152] P. McBurney and S. Parsons. Representing epistemic uncertainty by means of dialectical argumentation. *Annals of Mathematics and Artificial Intelligence*, 32(1–4):125–169, 2001.

[153] P. McBurney and S. Parsons. Games that agents play: A formal framework for dialogues between autonomous agents. *Journal of Logic, Language and Information*, 13:315–343, 2002.

[154] P. McBurney and S. Parsons. Locutions for argumentation in agent interaction protocols. In *AAMAS*, pages 1240–1241, 2004.

[155] P. McBurney and S. Parsons. Dialogue games for agent argumentation. In I. Rahwan and G. Simari, editors, *Argumentation in Artificial Intelligence*, chapter 13, pages 261–280. Springer, Berlin, Germany, 2009.

[156] P. Mcburney, R.M. Van Eijk, S. Parsons, and L. Amgoud. A dialogue game protocol for agent purchase negotiations. *Autonomous Agents and Multi-Agent Systems*, 7(3):235–273, 2003.

[157] J.A. McDermid. Support for safety cases and safety arguments using SAM. *Reliability Engineering & System Safety*, 43(2):111–127, 1994.

[158] B. Meyer. Introduction to the theory of programming languages. *Prentice-Hall International Series In Computer Science*, page 447, 1990.

[159] S. Modgil. Reasoning about preferences in argumentation frameworks. *Artificial Intelligence*, 173(9-10):901–934, 2009.

[160] S. Modgil and J. Fox. A guardian agent approach to safety in medical multi-agent systems. *Safety and Security in Multiagent Systems*, pages 67–79, 2009.

[161] M.F. Moens, E. Boiy, R.M. Palau, and C. Reed. Automatic detection of arguments in legal texts. In *Proceedings of the 11th international conference on Artificial intelligence and law*, pages 225–230. ACM, 2007.

[162] R.A. Montgomery. Living donor exchange programs: theory and practice. *British medical bulletin*, 98(1):21, 2011.

[163] M. Morge. The hedgehog and the fox: An argumentation-based decision support system. In *Proceedings of the 4th international conference on Argumentation in multi-agent systems*, pages 114–131. Springer-Verlag, 2007.

[164] B. Moulin, H. Irandoust, M. Bélanger, and G. Desbordes. Explanation and argumentation capabilities: Towards the creation of more persuasive agents. *Artificial Intelligence Review*, 17(3):169–222, May 2002.

[165] A. Munne and N. Prat. Defining river types in a Mediterranean area: a methodology for the implementation of the EU Water Framework Directive. *Environmental management*, 34(5):711–729, 2004.

[166] OCATT. Organització Catalana de Transplantaments.
http://www10.gencat.net/catsalut/ocatt/en/htm/index.htm.

[167] E. Oliva, P. McBurney, A. Omicini, and M. Viroli. Argumentation and Artifacts for Negotiation Support. *International Journal of Artificial Intelligence*, 4(S10):90, 2010.

[168] S. Ontañón and E. Plaza. Learning and joint deliberation through argumentation in multiagent systems. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, page 159. ACM, 2007.

[169] S. Parsons. *Qualitative methods for reasoning under uncertainty*. The MIT Press, 2001.

[170] S. Parsons, P. McBurney, E. Sklar, and M. Wooldridge. On the relevance of utterances in formal inter-agent dialogues. In *Proceedings of the 4th international conference on Argumentation in multi-agent systems*, pages 47–62. Springer-Verlag, 2007.

[171] C. Perelman and L. Olbrechts-Tyteca. *The New Rhetoric: a Treatise on Argumentation.* Notre Dame Press, University of Notre Dame, 1969.

[172] R.N. Pierson III, A. Dorling, D. Ayares, M.A. Rees, J.D. Seebach, J.A. Fishman, B.J. Hering, and D.K.C. Cooper. Current status of xenotransplantation and prospects for clinical application. *Xenotransplantation*, 16(5):263–280, 2009.

[173] M. Poch, J. Comas, I. Rodríguez-Roda, M. Sànchez-Marrè, and U. Cortés. Designing and building real environmental decision support systems. *Environmental Modelling & Software*, 19(9):857–873, 2004.

[174] J. L. Pollock. Defeasible reasoning. *Cognitive Science*, 11:481–518, 1987.

[175] EA Pomfret, RS Sung, J. Allan, M. Kinkhabwala, JK Melancon, and JP Roberts. Solving the organ shortage crisis: The 7th annual american society of transplant surgeons state-of-the-art winter symposium. *American Journal of Transplantation*, 8(4):745–752, 2008.

[176] H. Prakken. Coherence and flexibility in dialogue games for argumentation. *Journal of logic and computation*, 15(6):1009, 2005.

[177] H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7:25–75, 1997.

[178] H. Prakken and G. Vreeswijk. *Handbook of Philosophical Logic, second edition*, chapter Logics for Defeasible Argumentation. Kluwer Academic Publishers, 2002.

[179] N. Prat and A. Munne. Water use and quality and stream flow in a Mediterranean stream. *Water Research*, 34(15):3876–3881, 2000.

[180] Quintilian. *Institutio Oratoria*. Harvard University Press, 1920. Translated H.E. Butler.

[181] I. Rahwan and L. Amgoud. An argumentation-based approach for practical reasoning. *Argumentation in Multi-Agent Systems*, pages 74–90, 2007.

[182] I. Rahwan, B. Banihashemi, C. Reed, D. Walton, and S. Abdallah. Representing and classifying arguments on the semantic web. *The Knowledge Engineering Review (to appear)*, 2010.

[183] I. Rahwan and P. McBurney. Guest Editors' Introduction: Argumentation Technology. *IEEE Intelligent Systems*, pages 21–23, 2007.

[184] I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons, and L. Sonenberg. Argumentation-based negotiation. *Knowledge engineering review*, 2004. In press.

[185] I. Rahwan and G.R. Simari. *Argumentation in artificial intelligence*. Springer Publishing Company, Incorporated, 2009.

[186] I. Rahwan, F. Zablith, and C. Reed. Laying the foundations for a world wide argument web. *Artificial Intelligence*, 171(10-15):897–921, 2007.

[187] H. Raiffa. *Decision Analysis – Introductory Lectures on Choices under Uncertainty*. Addison-Wesley, Reading, MA, 1968.

[188] C. Reed. Dialogue frames in agent communication. In *International Conference on Multi Agent Systems, 1998. Proceedings*, pages 246–253, 1998.

[189] C. Reed and F. Grasso. Recent advances in computational models of natural argument. *International Journal of Intelligent Systems*, 22(1):1–15, 2007.

[190] C. Reed and T. J. Norman, editors. *Argumentation machines: New frontiers in argument and computation*. Kluwer Academic Publishers, 2004.

[191] C. Reed and G. Rowe. Araucaria: Software for puzzles in argument diagramming and XML'. Technical report, Department of Applied Computing, University of Dundee, Dundee, Scotland, UK, 2001.

[192] C. Reed and G. Rowe. Araucaria: Software for argument analysis, diagramming and representation. *International Journal on Artificial Intelligence Tools*, 13(4):983–, 2004.

[193] C. Reed and G. Rowe. Araucaria: Software for argument analysis, diagramming and representation. *International Journal of AI Tools*, 14(3-4):961–980, 2004.

[194] C. Reed and D. Walton. Towards a formal and implemented model of argumentation schemes in agent communication. *Autonomous Agents and Multi-Agent Systems*, 11(2):173–188, 2005.

[195] C. Reed and S. Wells. Dialogical Argument as an Interface to Complex Debates. *IEEE Intelligent Systems*, pages 60–65, 2007.

[196] C. Reed, S. Wells, J. Devereux, and G. Rowe. Aif+: Dialogue in the argument interchange format. In *Proceeding of the 2008 conference on Computational Models of Argument: Proceedings of COMMA 2008*, pages 311–323. IOS Press, 2008.

[197] I. Rodríguez-Roda, M. Sànchez-Marrè, J. Comas, J. Baeza, J. Colprim, J. Lafuente, U. Cortés, and M. Poch. A hybrid supervisory system to support WWTP operation: implementation and validation. *Water Science & Technology*, 45(4):289–297, 2002.

[198] B. Roth and B. Verheij. Cases and dialectical arguments - an approach to case-based reasoning. In *OTM Workshops*, pages 634–651, 2004.

[199] F. Sadri, F. Toni, and P. Torroni. Logic agents, dialogues and negotiation: An abductive approach. In *In Proceedings AISB'01 Convention*. AISB, 2001.

[200] Roger Schank. *Dynamic Memory: A Theory of Learning in Computers and People*. New York: Cambridge University Press, 1982.

[201] J.R. Searle. *Speech acts: An essay in the philosophy of language*. Cambridge university press, 1969.

[202] J.R. Searle. *Rationality in action*. The MIT Press, 2003.

[203] P. Serra, M. Sànchez, J. Lafuente, U. Cortés, and M. Poch. ISCWAP: A knowledge-based system for supervising activated sludge processes. *Computers and Chemical Engineering*, 21(2):211–221, 1997.

[204] R.D. Shankar and M.A. Musen. Justification of automated decision-making: medical explanations as medical arguments. In *Proceedings of the AMIA Symposium*, page 395. American Medical Informatics Association, 1999.

[205] W. Shen, F. Maturana, and D. H. Norrie. MetaMorph II: an agent-based architecture for distributed intelligent design and manufacturing. *Journal of Intelligent Manufacturing*, 11(3):237251., 2000.

[206] S.B. Shum et al. Cohere: Towards web 2.0 argumentation. *Computational Models of Argument (COMMA)*, 44, 2008.

[207] C. Sierra, N. R. Jennings, P. Noriega, and S. Parsons. A framework for argumentation-based negotiation. In M. P. Singh, A. Rao, and M. J. Wooldridge, editors, *Intelligent Agents IV: Agent Theories, Architectures, and Languages. Proceedings of the Fourth International ATAL Workshop*, Lecture Notes in Artificial Intelligence 1365, pages 177–192, Berlin, Germany, 1998. Springer.

[208] G.R. Simari and R.P. Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence*, 53:125–157, 1992.

[209] S. Simoff, C. Sierra, and R.L. De Mántaras. Requirements towards automated mediation agents. In *Pre-proceedings of the KR2008-workshop on Knowledge Representation for Agents and Multi-Agent Systems (KRAMAS), Sydney, September 2008*, page 171. Citeseer, 2008.

[210] D.B. Skalak and E.L. Rissland. Arguments and cases. an inevitable intertwining. *Artificial Intelligence and Law*, 1:3–44, 1992.

[211] H. Smit and J. De Jong. Historical and present day management of the river Rhine. *Water Science & Technology*, 23(1/3):111–120, 1991.

[212] L.K. Soh and C. Tsatsoulis. A real-time negotiation model and a multi-agent sensor network implementation. *Autonomous Agents and Multi-Agent Systems*, 11(3):215–271, 2005.

[213] D. Suthers, A. Weiner, J. Connelly, and M. Paolucci. Belvedere: Engaging students in critical discussion of science and public policy issues. In *Proceedings of the 37th World Conference on Artificial Intelligence in Education*, pages 266–273, 1995.

[214] K. Sycara. Persuasive argumentation in negotiation. *Theory and Decision*, 28:203–242, 1990.

[215] Katia Sycara. Arguments of persuasion in labour mediation. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 294–296, 1985.

[216] M. Sykes, A. d'Apice, and M. Sandrin. Position paper of the ethics committee of the international xenotransplantation association. *Xenotransplantation*, 10(3):194–203, 2003.

[217] G. Tchobanoglous, F.L. Burton, and H.D. Stensel. *Wastewater engineering: treatment and reuse*. McGraw-Hill Science/Engineering/Math, 2003.

[218] R.H. Thomason. Towards a logical theory of practical reasoning. In *AAAI Spring Symposium on Reasoning About Mental States: Formal Theories and Applications*, 1993.

[219] P. Tolchinsky, K. Atkinson, P. McBurney, S. Modgil, and U. Cortés. Agents deliberating over action proposals using the *proclaim* model. In Hans-Dieter Burkhard, Gabriela Lindemann, Rineke Verbrugge, and László Zsolt Varga, editors, *CEEMAS*, volume 4696 of *Lecture Notes in Computer Science*, pages 32–41. Springer, 2007.

[220] S. Toulmin. *The Uses of Argument*. Cambridge University Press, 1958.

[221] RM van Eijk. *Programming languages for agent communications*. PhD thesis, PhD thesis, Department of Computer Science, Utrecht University, The Netherlands, 2000.

[222] T. van Gelder. A Reason!Able Approach to Critical Thinking. *Principal Matters: The Journal for Australasian Secondary School Leaders*, 34(6), 2002.

[223] J. Vázquez-Salceda. *The Role of Norms and Electronic Institutions in Multi-Agent Systems*. Whitestein Series in Software Agent Technologies and Autonomic Computing. Springer, 2004.

[224] J. Vázquez-Salceda, U. Cortés, J. Padget, A. López-Navidad, and F. Caballero. The organ allocation process: a natural extension of the CARREL Agent-Mediated Electronic Institution. *AiCommunications. The European Journal on Artificial Intelligence*, 3(16), 2003.

[225] B. Verheij. Dialectical argumentation with argumentation schemes: An approach to legal logic. *Artif. Intell. Law*, 11(2-3):167–195, 2003.

[226] M. Veroux, D. Corona, M. Gagliano, M. Sorbello, M. Macarone, M. Cutuli, G. Giuffrida, G. Morello, A. Paratore, and P. Veroux. Voriconazole in the treatment of invasive aspergillosis in kidney transplant recipients. In *Transplantation proceedings*, volume 39, pages 1838–1840. Elsevier, 2007.

[227] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[228] G. A. W. Vreeswijk. *Studies in Defeasible Argumentation*. Doctoral dissertation Free University Amsterdam, 1993.

[229] G. Vreeswik and H. Prakken. Credulous and sceptical argument games for preferred semantics. *Logics in Artificial Intelligence*, pages 239–253, 2009.

[230] D. Walton and O. Windsor. An Overview of the Use of Argumentation Schemes in Case Modeling. *Modelling Legal Cases*, page 77, 2009.

[231] D. N. Walton. *Argument Schemes for Presumptive Reasoning*. Lawrence Erlbaum Associates, Mahwah, NJ, USA, 1996.

[232] D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. SUNY Series in Logic and Language. State University of New York Press, Albany, NY, USA, 1995.

[233] D.N. Walton. *Argumentation Schemes for Presumptive Reasoning*. Lawrence Erlbaum Associates, 1996.

[234] D.N. Walton and E.C.W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of New York Press, New York, 1995.

[235] J. Wanner. Stable Foams and Sludge Bulking: The Largest Remaining Problems. *Journal of the Chartered Institution of Water and Environmental Management*, 12(10):368–374, 1998.

[236] R.H. Wiesner. Patient selection in an era of donor liver shortage: current us policy. *Nature Clinical Practice Gastroenterology & Hepatology*, 2(1):24–30, 2005.

[237] M. Williams and A. Hunter. Harnessing ontologies for argument-based decision-making in breast cancer. In *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*, volume 2, pages 254–261. Citeseer, 2007.

[238] T. Wittwer and T. Wahlers. Marginal donor grafts in heart transplantation: lessons learned from 25 years of experience. *Transplant International*, 21(2):113–125, 2007.

[239] A. Wyner and T. Bench-Capon. Argument schemes for legal case-based reasoning. *Legal knowledge and information systems. JURIX*, pages 139–149, 2007.

[240] L.R. Ye and P.E. Johnson. The impact of explanation facilities on user acceptance of expert systems advice. *Mis Quarterly*, 19(2):157–172, 1995.

[241] R. Ye et al. The value of explanation in expert systems for auditing: An experimental investigation. *Expert Systems with Applications*, 9(4):543–556, 1995.

[242] S. Zink, H. Smolen, J. Catalano, V. Marwin, and S. Wertlieb. NATCO, the organization for transplant professionals public policy statement. HIV-to-HIV transplantation. *Progress in transplantation (Aliso Viejo, Calif.)*, 15(1):86, 2005.