# ARGUING FROM EXPERIENCE:

*Persuasive Dialogue Based on Association Rules.*

Thesis submitted in accordance with the requirements of the
University of Liverpool for the degree of Doctor in
Philosophy

by

MAYA WARDEH

OCTOBER 2009

# Abstract: "*Arguing from Experience*" by Maya Wardeh

The development of autonomous software agents requires consideration of a number of elements. One interesting aspect of the study of software agency is to enable effective inductive reasoning, amongst agents, using accumulated experience of an agent. However, this experience may vary from agent to another, and only by exploiting these differences, in the right way, can the agents come to an agreement regarding some issue. This thesis is concerned with one particular aspect of such agency: modelling the process of arguing from "*experience*" to equip autonomous agents (entities) with the capability to jointly coming to a "*view*" regarding some case, using the experience they have independently gathered over time. The background setting for this work deals with the topic of induction as a dialectical form of reasoning, and attempts to address some issues regarding its treatment in philosophy, as well as the problems inherent in the computational modelling of such reasoning. The main output of the study is a model to enable "*Arguing from Experience*" which uses techniques from the field of argumentation theory and knowledge discovery in databases, to enable agents to pool and construct arguments in support of and against proposals for "*views*" regarding some given case. Arguments are pooled from the agent's experience by means of Association Rule Mining (ARM) techniques. The proposed model is intended to describe how "*Arguments from Experience*" can be put forward and be systematically attacked in a variety of ways. This enables the agents to consider all available options and come to a conclusion about the best "*view*" to associate with the given case. The underlying theory extends a well established account from the field of philosophy, based on the use of argument schemes and critical questions. The account given is then formalised in terms to enable its representation in agent systems.

The underlying model has formed the basis for two applications: an implementation of a dialogue game protocol to provide a proof of concept, as well as means to enable automated two-party *“Arguing from Experience”* for the purposes of classification; and a framework to aid multiparty *“Arguing from Experience”*, intended as means to facilitate argumentation between more than two parties. Both applications are evaluated using variety of setups in various domains. The obtained results provided empirical evidence to the efficacy of “*Arguing from Experience*”.

# Acknowledgements

This thesis would have not been completed if not for the help I have received from a number of people whose contribution to my research deserves a special mention. It is a pleasure to convey my gratitude to them all in this modest acknowledgement.

In the first place, I am most grateful for the Agha Khan Foundation (AKF) and Agha Khan Development Network (AKDN) for awarding me their generous scholarship to pursue my life-dream in studying overseas. Without their support I would have not been able to finish my research and write this thesis. I would also like to thank The Department of Computer Science at the University of Liverpool for providing me with sufficient financial assistance to attend a number of conferences/workshops and seminars throughout my studying years.

The Department was also the perfect venue, in which I have met many kind people, who have helped me during the entire course of my studies by providing me with endless encouragement and constructive feedback. In particular, I would like to thank Katie Atkinson, Peter Mcburney, Paul Dunne and Michele Zito. I would also like to thank my fellow PhD students at Liverpool. Above all, thanks go to the students with whom I shared room 211 during the past three years, and who had been always there for me when I needed them the most: Justin Wang, Omar Baquerio Espinosa, Kamal Ali Al-Bashiri, Ji Ruan, Santhana Chaimontree and Stephanie Chua, to name but a few.

My deepest and most grateful thanks go to my family and friends in Syria. Their support, trust, encouragement and thoughtfulness have been priceless to me. My parents, Eman and Ali Wardeh, and my brother Tim, have always been there for me. I would have not made it today if not for their kind words and their warm hugs. I cannot express how very grateful I am to the three of them for the love and support they have given to me, and for all the jasmines they have sent from Damascus to make my flat smell like home. Great thanks are also due to my dearest friends Lana Meiqari and Hani Abu-Shaer, who stood by my side and provided me with invaluable emotional support: They always listened to my complaints, justified and unjustified, and provided me with motivation whenever my self-motivation ran out. I would also like to thank Rob Higgs for showing interest in my research, and for chatting with me for hours and hours about the details of this research.

There is also one more person, that I feel most grateful for the everlasting help he had given me during the past few years, and who has been by my side for the most part of my PhD studies: my partner George Kidd. I cannot express how grateful I am for having you in my life. Not only that you have provided me with home to return to, and a shoulder to cry on every time my research hit a dead-end. But you also were my anchor, attaching me to reality so I do not get lost. Your patience, love, warm heart and uniqueness kept me going through the hardest times, and the coldest winters. George thanks a lot for being there for me, and hopefully one day I shall be able to return the favour.

Finally, there are two people to whom I am more grateful than words can convey: my supervisors Frans Coenen and Trevor Bench-Capon. They were the best supervisors any one could hope for, and more. Their extraordinary experience, which they happily shared with me, was most invaluable. Their constructive supervision and their generous feedbacks helped me, more than I can measure or express, in conducting my research in most efficient way. Their combined knowledge was the most perfect resource, which inspired me and enriched my growth, not only as a student but also as a researcher, an intellectual and a person. Frans, Trevor, I am grateful in every possible way to you both for your contribution in making me the person that I am today. It has been a pleasure and an honour to have been supervised by you both.

# Contents

# List of Figure

List of Figures.

# List of Tables

# Chapter 1: Introduction

*"Where shall I begin, please your Majesty?"*

*"'Begin at the beginning," the King said gravely, "and go on till you come to the end: then stop."*

***Lewis Carroll, British author (1832, 1898).***
***Alice's Adventures in Wonderland. Alice's Evidence.***

This thesis is concerned with the automation of the process by which humans come to a "*view*", regarding a given subject, by consulting the experience they have gathered over time. This usage of experience is rather common feature of day-to-day life. We often make use of our (personal) experience when conversing with other people, by observing certain "*regularities"* in this experience, and then employing these "*regularities"* to back up what is being said. For instance, it is not strange to hear someone saying: "*Person x will be late today, because she has always been late every time we had an appointment*", or: "*We should go to restaurant y because every time we have been there the service was excellent*".

This thesis aims to model this mode of inference, such that it can be automated and deployed by software agents, or other forms of autonomous software entities. This automation is intended to provide the agents with the capability to reason from their accumulated experience, and to employ this form of reasoning to argue with other agents to come to a conclusion regarding a given situation. Thus, experience will contribute to the agents reasoning about the unknown from the basis of what they have experienced. However, this style of reasoning is rather defeasible - we may well agree with the facts of a particular case but may reject the conclusion presented to us because it does not fit within our own experience. A successful application of reasoning from experience should enable agents to argue with each other from the basis of their individual experiences, so that they can learn from one another, and come up with a solution that is compatible with the experience of them both. Such usage of experience will be shown invaluable when it is not possible to use other types of reasoning, such as proof or reasoning from beliefs. To deliver this style of reasoning this thesis proposes the concept of "*Arguing from Experience*", which is intended to enable software agents to "*argue with each other*" on the basis of their experience, so as to come to a decision with respect to some given issue.

The rest of this chapter is organised as follows. In Section 1.1 the research question addressed by this thesis is outlined. A general overview of the research area that the investigation undertaken by this thesis falls under is provided in Section 1.2. An outline of the structure of this thesis is presented in Section 1.3.

## 1.1. Research Question

There are many research challenges to be met if we are to realise the full implementation of reasoning from past experience within software entities, especially if these entities are intended to be autonomous. In this latter case, accumulating and processing experience from the world, as well as employing this experience in the decision making process, are seen as essential requirements that are to be addressed should we wish to equip these software entities with the capability to reason from their past experience to accommodate a variety of situations. This is, of course, a significant task and the work presented in this thesis focuses on one particular part of the latter requirement. The fundamental problem addressed in this thesis is:

> *By what means may a model, that enables software entities to make use of their accumulated experience to jointly reason about a given situation, be realised; and how might such a model be evaluated?*

This question arose from observing the process by which we learn from experience and employ this experience to persuade and convince others to change their minds regarding a given issue, and from the belief that designing and implementing a model to enable agents to argue from their past experience in a similar manner to humans provides an interesting academic challenge. The research undertaken to answer the above research question draws upon a number of disciplines, including: philosophy, artificial intelligence, knowledge discovery, data mining and law, as will be made clear in the forthcoming chapters. The next section gives a brief overview of the research domain identified with the work described in this thesis

## 1.2. Research Domain

The advent of artificial intelligence (AI) was based on the conjecture that the process of human reasoning can be automated:

> *"The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it".*
>
> ***From the invitation to the Dartmouth 1956 conference which some see as the beginnings of AI.***

One of the aims of AI is to enable the construction of useful software entities that are equipped with the capabilities to reason about a variety of things, and to display some form of intelligence, rather than mechanically performing pre-defined tasks. One increasingly popular branch of AI that is attempting to deal with these issues is "*multi-agent systems*" (MAS). MAS has witnessed significant development in the past few decades, mainly through research aimed at building systems of distributed, autonomous software agents. Although agents are now firmly established within computer science there is still no universally accepted definition of an agent. One commonly used definition was given by Wooldridge:

> *"An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives." (Wooldridge, 2001, p. 15).*

This thesis makes use of the above definition when referring to (intelligent) agents. However, the main focus of this thesis is to provide a model for reasoning and arguing from experience; regardless of whether this model is to be embodied in software agents or any other form of autonomous software entities. Nevertheless, it is believed that the work undertaken by this thesis is a step toward the realisation of the above capabilities, and the delivery of fully autonomous and intelligent agents.

To give some indication of how we might go about equipping software agents with a mechanism to reason from gathered experience, we can turn to the field of philosophy to give us insights into how this reasoning process is manifest in humans. This type of reasoning is broadly inductive: we induce inferences from our experience rather than deducing facts. Thus, this thesis looks to the field of inductive reasoning in philosophy: numerous philosophers dating from recent times and going back to the time of the ancient Greek Philosophers have given in depth analyses of the nature of this reasoning. Inductive reasoning embodies a number of distinctive and interesting features that have been described and accounted for in a variety of different ways. However, two issues are of interest if we are to enable the effective representation of reasoning from experience in software agents. Firstly, the subject of reasoning must be identified. We can turn to our experience to reason about almost everything, whether it is a belief or an action. However, this thesis will make use of experience in one context - to come to a "*view*" regarding a given case. Secondly and most importantly, experience differs from entity to entity, whether human or agent. Therefore, any successful integration of reasoning from experience into autonomous software systems, including agents, should cater for (and exploit) this fact.

Taking the above two points into consideration, part of the process of reasoning from experience involves considering all the possible "*views*" available in any given case. There may be competing options that require careful consideration to enable the best decision to be taken. A sub field of philosophy that can help in dealing with this issue is argumentation theory. This field is concerned with the presentation, interaction and evaluation of arguments that support or reject a particular position on a matter. Argumentation provides an arena in which the critical evaluation of the issues in question can be reasoned about. It is also extremely useful in situations where knowledge is incomplete or inconsistent.

Thus, argumentation provides a means to accommodate the differences in the experience of the participants taking part in the argument. This is particularly relevant when it is desired that a number of autonomous software systems should collectively come to some decision regarding a given situation where there may be a range of possible outcomes and/or the agents have different experiences which may promote some option and demote others. The work presented in this thesis adopts argumentation theory to model argumentative techniques that can be deployed by autonomous systems in scenarios where there is a range of options available. More specifically this thesis makes use of one method from this field, the concept of argument schemes and critical questions, which provides a precise structure by which justifications for each "*view*" can be presented and criticised.

Having established argumentation as a means to enable reasoning from experience, hereafter referred to as "*Arguing from Experience*", the principal issue to consider is how best to capture the experience (knowledge) to be held by individual agents. In established argumentation systems the debate takes place in a context where the participants have hand-engineered knowledge bases and the dialogue serves to exchange this knowledge; persuasion takes place due to inconsistencies or gaps in the participants' knowledge. In the proposed "*Arguing from Experience*" persuasion happens because of differences in the experience of participants regarding the subject matter, or due to the fact that one participant has gained more experience than the others. Thus a mechanism other than knowledge base engineering needs to be developed such that arguments can be pooled directly from each participant's experience. In order to address this issue, this thesis makes use of association rule mining technology as studied in the context of knowledge discovery in databases. The advent of association rule mining as a means to uncover interesting inferences in a large collection of data provides a means to represent arguments from experience. As will be demonstrated, association rule mining not only provide the means to enable the automatic generation of arguments from experience; but, once these arguments are generated, association rules can be easily evaluated against each other.

Aside from equipping agents with coherent mechanisms for reasoning, agents will also need to be able to communicate effectively with their counterparts, in order to come with a decision regarding the issue at hand. To fully realise the process of "*Arguing from Experience*", a mechanism for facilitating dialogues amongst a number of agents is also of essence to this thesis. Here, this thesis considers the field of dialogue games to look for a means to enable communication between the agents engaged in "*Arguing from Experience*".

Once the process for reasoning and arguing from experience is incorporated into autonomous agents (software entities), this process will offer exciting prospects for the development of technology that can be of benefit to a wide variety of disciplines. One potential field that may benefit from reasoning from experience is classification. "*Arguing from Experience*" allows an agent to draw directly from past experience to find reasons for coming to a "*view*" on some current example, without the need to analyse this experience into rules and rule priorities. A "*view*" on a current example is expressed in terms of a classification, thus the discovered associations provide support for a given example to be categorised as being of one class or another. This application of reasoning from experience requires an extensive study of the field of classification in databases. In fact this latter topic is closely related to inductive reasoning as studied in philosophy.

Given the above, the main research goals of this thesis are now summarised:

1. To provide a theory of persuasion within the setting of reasoning from experience that accounts for the defeasible nature this style of reasoning, and to provide the means by which the advocated theory can be implemented to enable different participants to draw arguments directly from their past experience. The later would avoid the knowledge engineering bottleneck that occurs when belief bases must be constructed. The suggestion is that past experience can be captured through the use of the notion of association rules to represent arguments.
2. To show how this theory can be transformed into a computational framework that can be effectively deployed in autonomous software

systems. Two instantiations of this framework are thought necessary, to enable dialogues for *"Arguing from Experience*".

3. To evaluate the two instantiations of the framework by applying the concept of "*Arguing from Experience*" to classification problems. Thus, incorporating the process argumentation into the field of data mining. The success of the "*Arguing from Experience*" model will be measured by the quality of the classifications it produces.
4. To assess the application of "*Arguing from Experience*" to classification by means of comparative empirical experiments; providing means to evaluate the promoted incorporation of argumentation and data mining.

The concluding chapter of this thesis, Chapter 9, will return to these research goals to discuss how well they have been met.

## 1.3. Thesis Structure

This thesis is structured into nine chapters as follows:

**Chapter 1**, this chapter, in which the research issues addressed by this thesis has been identified.

**Chapter 2** presents a literature survey of existing research which is relevant to the contributions presented in this thesis.

**Chapter 3** introduces a theoretical model of reasoning from experience, which builds upon the existing accounts of inductive reasoning and argumentation schemes in philosophy. This chapter also presents a scheme for "*Argument from Experience*" to support the derivation of a desired claim for a given case by the means of association rules linking some features in the case to the claim. Formalism for "*Arguing from Experience*" is also given to enable software entities (agents) to engage in debates to promote a "*view*" with respect to a given case.

In the proposed model, "*views*" are presented by possible classification of cases from given domains. Thus, a potential by-product of the promoted model will be its application to solve classification problems. Subsequent chapters will present empirical evidence showing that this approach can compete with other well known classification solutions.

**Chapter 4** proposes a protocol for two-party dialogues for "*Arguing from Experience*", referred to as PADUA. The proposed protocol takes the theory articulated in Chapter 3 as its underlying model to enable persuasive dialogues to be undertaken by two participants to consider a classification problem. This chapter also gives a concise discussion of an implementation of the protocol for use in dialogues for the purposes of binary classification.

**Chapter 5** provides a detailed assessment of the PADUA protocol by the means of empirical experiments designed to investigate the process of two-party "*Arguing from Experience*", as embodied in PADUA, and the resulting dialogues. The obtained results demonstrate that PADUA can facilitate dialogues between two participants in variety of situations.

**Chapter 6** is concerned with the taking the theory of "*Arguing from Experience*" forward for use to aid joint reasoning from experience amongst any number of agents. Multiparty dialogues, of this style, raise a number of significant issues, necessitating appropriate design choices. A new system, called PISA, is presented directed at these issues. Both the design and the implementation of PISA are also fully described in this chapter.

**Chapter 7** investigates different areas of interest regarding multiparty "*Arguing from Experience*", as manifested in PISA, and their possible treatments within the promoted framework. The discussion given in this chapter provides an insight as to how altering the setups of the promoted structure for PISA may influence the resulting dialogues.

**Chapter 8** further establishes the promoted PISA Framework by the means of empirical evidence. A series of experiments will be discussed to demonstrate the ability of PISA to administer and facilitate multiparty dialogues amongst any number of participants.

**Chapter 9** is the final chapter in this thesis and it provides a summary of all the work presented, as well as a discussion of possible avenues for future research work.

Additional elements are included as appendices. **Appendix A** contains further details of the specification, and design of the Java software used to realise the PADUA protocol. The material in Appendix B is intended to supplement the descriptions and discussions of the implementation of the protocol given in Chapter 4. **Appendix B** provides details of the specification and design of the Java application of the PISA Framework, highlighting which units are inherited from PADUA and which are designed to enable multiparty dialogues as embodied in PISA. The application details listed in Appendices A and B are intended to help the reader, should they wish, in understanding and executing both PADUA and PISA applications, available for anonymous download from the author's personal WWW page at www.csc.liv.ac.uk/~maya. **Appendix C** contains supplementary extensions to the proposed structure of PISA. **Appendix D** discusses the results of an experiment intended to evaluate the operation of PISA with certain type of noise.

Some of the work discussed in previous chapters has been developed jointly with other co-authors. A part of this work was also presented at various refereed conferences, workshops and seminars. Segments of work presented in this thesis have been published, or accepted for publication, as joint work with the author's supervisors Frans Coenen and Trevor Bench-Capon as follows:

- The figures in Section 3.3 which present the pseudo code for the algorithms implemented to enable the agents (entities) to mine adequate arguments from their data have been published in (Wardeh et al., 2007a). Table 3.2 which models the legal next speech acts in the proposed "*Arguing from*

*Experience*" has also been published in (Wardeh et al., 2009a, 2008a, 2008b, 2007a, 2007b).

- The general framework for "*Arguing from Experience*" (Section 3.4) appears in (Wardeh et al., 2009a, 2008a, 2008b).
- The details of the PADUA protocol have been published in (Wardeh et al., 2009a, 2008a, 2008b) and the associated strategy mode in (Wardeh et al., 2007b). Also, the design details of the PISA Framework appear in (Wardeh et al., 2009b, 2009c).
- Some of the empirical results reported in Chapters 5 and 8, have been published in (Wardeh et al., 2009a, 2008a), and (Wardeh et al., 2009b).

# Chapter 2: Literature Review

This chapter presents an overview of the existing research literature that is relevant to the issues addressed in this thesis. As discussed in the introductory chapter, the main concern of this thesis is forming a theory for "*Arguing from Experience*", and implementing this theory so that it can be tested and evaluated. In order to place the research work described in this thesis in the appropriate broader context, this chapter is divided into three separate sections:

1. *Argumentation in Philosophy and AI:* Section 2.1 discusses the theoretical ideas behind "*Arguing from Experience*". These come from variety of fields such as Argumentation Schemes in informal logic, AI and Law, CBR, and Dialogue Games.
2. *Association Rule Mining (ARM) and Knowledge Discovery (KDD)*: Section 2.2 surveys the domain of ARM in the field of KDD. Note that the Association Rules will be used to provide the basis for the promoted "*Arguments from Experience*", as will be discussed in Chapter 3.
3. *Classification*: Section 2.3 presents an overview of the treatment of "*Classification*" problems in the field of KDD and discusses different approaches to these problems. These approaches will provide a benchmark against which the promoted model can be tested.

Each section concludes with a summary of the key points addressed.

## 2.1. Argumentation in Philosophy and AI

In this Section an overview of the issues composing the theoretical background upon which the theory model for "*Arguing from Experience*", proposed in the forthcoming chapters of this thesis, is based.

### 2.1.1. Inductive Reasoning

This sub-section examines the topic of "*Inductive Reasoning*", from its early philosophical roots to its treatment in more recent literature. A number of definitions and examples of this form of reasoning are given along with discussion of some of the features and problems inherent in it. In particular, Swinburne's account of "*Inductive Arguments*" (Swinburne, 1974) is discussed in detail.

#### *2.1.1.1. Overview*

Inductive inferences are basic to our everyday life and to human scientific thinking. We often assume that the unobserved will be, largely, like the observed. Or, as Hume puts it: "*From causes which seem similar we expect similar effects*." (Hume, 1902, p.6). Induction tells us that the sun will (probably) rise tomorrow, and that ice is cold and fire is hot, and thus takes us "*beyond the confines of our current evidence or knowledge to conclusions about the unknown*" (Sloman and Lagando, 2005). In computer science, induction has been adopted within the methodology of AI (Chalmers, 1982) as an invaluable tool by which intelligent systems can learn from their environment; indeed many machine learning algorithms, such as (Quinlan, 1993, 1998) and (Clark and Niblett, 1989), apply inductive reasoning. Moreover, the recent growth of interest in agents' technologies have brought to attention the fact that software agents cannot always encompass a well-specified model of the world at large on which they can optimise their behaviour, and therefore cannot always rely on deduction to provide the information they need when deciding how to act. Agents may instead choose to adapt gradually to their environment by "*inducing*" decisions based on their gathered experience. Many definitions for the term "*inductive reasoning*" exist from the large body of research on the topics of reasoning, philosophy of science, logic, knowledge discovery and mathematics. Without intending to discount the other definitions given by the many existing sources, a number of accounts of inductive reasoning have been chosen for discussion in the following sub-sections, since these are the ones closely related to the issues explored in this thesis. In the following narrative the origins of inductive reasoning is traced back to the times of Aristotle.

In The Topics, Aristotle (1997) defines "*Induction*" as the "*advance from particulars to universals*". According to Aristotle, we may proceed from examples such as the skilled navigator is the best and the skilled charioteer is the best, to conclude that the best in any occupation is the one who has learnt his job well ('*the one who knows*'). In comparison with syllogism, or deduction, Aristotle notes that induction has a number of distinguishing general characteristics such that it is more persuasive and clear, more easily learnt through the senses, and more readily available to human kind. The notion of inductive argument is also found in the works of Cicero. Gorman (2005) notes that Cicero defines "*inductive argument*" as the type of argument where one is led from easy and obvious cases to see an analogy in more difficult and darker matters. Of note regarding this account, is that the ancient philosophers treated inductive reasoning in the narrow sense of "*enumerative induction*" or "*induction by simple enumeration*". Thus if it is reported that a number of objects of one kind all have some property, then it can be concluded that all objects (or some further object) of that kind also have that property (Swinburne, 1974). For example, according to this type of induction, one may conclude from the premise that each swan observed thus far has been white, that all swans are white[1]. The same treatment can be found in more recent literature. For instance, Musgrave (2004) defines inductive reasoning as arguing from the premise that all observed *As* are *Bs* to the conclusion that the next *A* will be *B* or that all *As* are *Bs*. In other words, it is reasonable to believe something if it has been shown to be true, or probable. Note that this type of inductive reasoning operates in two ways: it either advances a conjecture by what are called confirming instances, or it falsifies a conjecture by contrary or disconfirming evidence (Gardner, 2001). For example, the hypothesis that all swans are white is increasingly confirmed each time a new swan is observed and found to be white. But once one swan is found to be not white the conjecture is falsified.

[1] This erroneous conclusion shows both that induction of this sort is always defeasible by a counter example, and that one may be misled by a limited range of experience.

### *2.1.1.2. Swinburne's (Correct) Inductive Arguments*

In the introduction to his collection of essays on the philosophy of induction, Swinburne (1974) makes several important points about inductive reasoning. In the following, these points are discussed in some length. Swinburne begins by proposing a definition for inductive argument as: "*an argument which is not deductively valid but one in which, it is claimed, the premises 'make it reasonable' for us to accept the conclusion".* Then he distinguishes between "*correct*" and "*incorrect*" Inductive Arguments[2] as follows:

> *"…correct inductive argument is one in which the premises do 'make it reasonable' for us to accept the conclusion, as claimed; and that an incorrect one is one in which they do not, but it is falsely claimed that they do"* (Swinburne, 1974, p.2).

Most of our everyday commonsense reasoning is based on this form of inference. For example, if you know that person *X* is always on time and that she has seldom been late for her appointments in the past, you may argue that if you have an appointment with her and she is fifteen minutes later, that she will not show up. This argument is not deductive: there is no contradiction in admitting the premises but denying the conclusion. Perhaps today she has been held up by another appointment, and so is late. Nevertheless, it would generally be supposed that in the absence of further evidence there is no reason to believe these conjectures, it is reasonable to assume that person *X* has no previous appointments on that particular day, unless we have some reason to suppose otherwise. Thus, according to Swinburne: "*We judge that the premises make it reasonable for us to accept the conclusion, even though no contradiction is involved in asserting the premises and denying the conclusion*". (Swinburne, 1974, p.2.). Russell (1974) explains that inferences of this kind are based on past regularities in our experience, which lead us to associate certain outcomes with experiences rather than others. Thus, inductive inferences are contingent, and in this they differ from deductive inferences which may be described as being necessary. Deductive inference can never support contingent judgments such as

[2] As opposed to "*valid*" and "*invalid*" deductive arguments.

stock market forecasts, nor can deduction alone explain why one particular chess strategy works well against one opponent and fails against another. Inductive inference can do these things, more or less successfully, because inductions are *ampliative*[3] (Kneale, 1949). They can amplify and generalise our experience, and broaden and deepen our empirical knowledge. Deduction on the other hand is of an *explicative* nature: it orders and rearranges our knowledge without adding to its content.

As mentioned above Swinburne distinguishes between *correct* and *incorrect* inductive arguments. Swinburne also notes two further important properties of correct inductive arguments:

- Unlike a valid deductive argument, correct inductive arguments yield only probable knowledge or reasonable belief. For example: "*The man who has fed the chicken every day throughout its life at last wrings its neck instead, showing that more refined views as to the uniformity of nature would have been useful to the chicken*", (Russell, 1974, p.21).
- "*Correct inductive arguments will only yield probable knowledge if we know nothing else which affects the probability of the conclusion, apart from what is stated by the premises*" (Swinburne, 1974). For instance, if, while waiting for person *X* to show up we learn of serious traffic congestion at some part of the route she normally takes, then it is no longer reasonable to conclude she will not show up to her appointment.

Thus *inductively correct* arguments, unlike *deductively valid* ones, have conclusions that go beyond what is contained in their premises. They are based on *learning from experience*. We often observe *patterns, resemblances,* and other kinds of *regularities* in our experiences, some quite simple (cakes are tasty); some very complicated (objects moving according to Newton's laws). This idea of learning from experience touches upon a particular point which is of great relevance in this thesis – the notion of "*arguing*" using "*past*

[3] "*One of the most striking characteristics of the induction used in natural sciences is that it goes in some sense beyond its premises, which are the singular facts of experience; I propose, therefore, to call it ampliative induction*" (Kneale, 1949).

*experience"*. This thesis argues that "*Arguing from Experience*" is an important and distinctive method of reasoning that may have great benefits once automated. This type of "*inductive argument*" is used in everyday human conversations to provide support for what is being said. In this way people often employ their past experience to validate their reasoning about their daily life. Such use of "*experience*" can span anything from experiences held within a particular group or community, to more personal, individual experiences. This thesis aims at enabling similar reasoning to computer systems, or indeed agents, such that experience can direct agents' reasoning regarding the categorisation of unknown cases. Thus, experience will contribute to agents' reasoning about the unknown from the basis of what they have experienced. Such experience based procedures provide an explanation as to why it is not always possible to persuade others to accept an opinion simply by demonstrating facts and proofs based on experiences. It may well be that a particular individual (human or agent) will accept the facts of a particular case but may reject the conclusion of its opposing argument because it does not fit within their own experience.

#### *2.1.1.3. The problem of induction*

The problem of justifying induction can be traced back to Hume (1902) and his famous argument aimed at enumerative induction. Essentially Hume's critique proceeds as follows: inductive arguments arise because we observe uniformities in nature. For example, that all observed swans have been white, and as such they are based on experience. However, we have no grounds for assuming that nature will continue to behave uniformly, other than the appeal to experience. But: "*If there be any suspicion that the course of nature may change, and that the past may be no rule for the future, all experiences become useless, and can give no rise to inference or conclusion. It is impossible, therefore, that any argument from experience can prove this resemblance of the past to the future; since all these arguments are founded on the supposition of that resemblance"*. (Hume, 1902, pp. 37-8). As things do not always behave as we expect, "*why may (this) not happen always, and with regard to all objects? What logic, what process of argument secures you against this suspicion?*" (Hume, 1902, p. 38).

Hume's answer to this dilemma is that there is no justification for believing that things will continue to behave as they have behaved, or as Russell puts it: "*The mere fact that something has happened a certain number of times causes animals and men to believe that it will happen again. Thus our instincts certainly cause us to believe that the sun will rise tomorrow, but we may be in no better position than the chicken which unexpectedly has its neck wrung*" (Russell, 1974 p.21). Various justifications for induction have been proposed. An excellent collection of a number of different justifications can be found in the collection of essays in (Swinburne, 1974). These justifications are mainly either pragmatic (deductive) or predictionist (inductive): the "*pragmatic*" approach is attractive because it appeals to the idea of correct reasoning methods. One drawback is that it is difficult to carry the argument through. The "*predictionist*" justification seeks to rationalise inductive arguments on the basis of their success in the past. The attraction of this approach is the appeal to the usefulness or otherwise of an argument. Note that inductive arguments are persuasive since the audience to which they are addressed will certainly believe that the future will resemble the past. As Hume said: "*On the contrary, the abstruse philosophy, being founded on a turn of mind, which cannot enter into business and action, vanishes when the philosopher leaves the shade, and comes into open day; nor can its principles easily retain any influence over our conduct and behaviour. The feelings of our heart, the agitation of our passions, the vehemence of our affections, dissipate all its conclusions, and reduce the profound philosopher to a mere plebeian.*" (Hume, 1902).

The accounts of inductive reasoning discussed above represent a small sample taken from a large amount of research on the topic. The above is intended to give a broad overview of the subject. The work presented in the forthcoming chapters articulates an account of inductive arguments in a model for "*Arguing from Experience*" tailored for the automation of pooling such arguments from a dataset representing a set of past examples (experience) in a particular domain. Mainly, it considers the notions of enumerative induction, and argument from analogy, as used in the context of CBR, and builds upon these two notions to form a theory for "*Arguing from Experience*".

#### *2.1.1.4. Probability and Induction*

So far only straightforward non-probabilistic accounts of inductive arguments have been discussed. The addition of probability to induction does not only yield a generalisation; probabilistic induction is much deeper and more complex than induction without probability. This sub-sub-section looks at several different approaches to specifying the problem of probabilistic induction.

Carnap in his work on philosophical foundations of probability and induction, (Carnap, 1952), lists five sorts of inductive inferences:

- *Direct inference* typically infers the relative frequency of a feature (attribute, trait) in a sample from its relative frequency in the population from which the sample is drawn.
- *Predictive inference* is inference from one sample to another sample not overlapping the first.
- *Inference by analogy* is inference from the features of one individual case to those of another on the basis of the features that they share.
- *Inverse inference* infers something about a population on the basis of premises about a sample from that population.
- *Universal inference* is inference from a sample to a hypothesis of universal form. Simple enumerative induction, mentioned above, is the typical example of universal inference.

Carnap initially held that the problem of induction was a logical problem; that assertions of degree of confirmation by evidence of a hypothesis should be analytic and depend only upon the logical relations of the hypothesis and evidence. Carnap's logical probability generalised the relation of logical implication to a numerical function, *c(h, e)*, that expresses the extent to which an evidence sentence *e* confirms a hypothesis *h*. Reichenbach's probability implication (Reichenbach, 1949) is also a generalisation of a deductive concept. Reichenbach (1949) argued, roughly, that induction works in the long run if anything works in the long run. On Reichenbach's view, the problem of induction is just the problem of determining probability on the basis of evidence (Reichenbach, 1949). The conclusions of inductions are not asserted, they are posited. Reichenbach divides inductions into several sorts, not quite analogous to the ones suggested by Carnap (1952). These are:

- *Induction by enumeration*, in which an observed initial frequency is conjectured to hold for the limit of the sequence;
- *Explanatory inference*, in which a theory or hypothesis is inferred from observations;
- *Cross induction*, in which distinct but similar inductions are compared and, perhaps, corrected;
- *Concatenation* is a sort of induction by enumeration that amounts to reiterated applications of the inductive rule.

Another approach to the problem of probabilistic induction assumes one has an initial *known subjective* probability distribution satisfying certain more or less weak conditions along with a method for updating one's probabilities and proves theorems about the results of such a method. (e.g. (Savage 1954), (Jeffrey, 2005)).

Statistical learning theory represents a different paradigm which assumes there is an *unknown objective* probability distribution that characterizes the data and the new cases about which inferences are to be made. The basic theory attempts to specify what can be proved about various methods for using data to reach conclusions about new cases. Formal learning theory formulates the problem of induction in general terms as the question of how an agent should use empirical data to confirm and reject hypotheses about the world. In specific instances the theory sets goals of inquiry and compares methods for pursuing those goals. Formal learning theory, like many other inductive methods, seeks deductive proof of the reliability of chosen inductive methods. See (Suppes,1998) for a critical discussion of this theory.

### 2.1.2. Argumentation Theory

Philosophy and argumentation have had a close connection since the time of the Ancient Greek philosophers - argumentation provided philosophers, deprived of scientific tools, with a source to acquire knowledge. Arguments can be simply defined as combinations of statements that are intended to change the minds of other people regarding some subject. The structure of an argument is similar to that of a proof in that both structures consist of premises leading to a particular conclusion. Arguments, however, diverge from proofs in that while in the latter the premises always entail the conclusion. In argumentation the premises give a reason for believing the conclusion is true - therefore it remains possible that the conclusion may not co-exist with the truth of the given premises. Bench-Capon and Prakken (2006) give a summary of the characteristic differences between arguments and proofs as follows: (i) the goal of an argument is to persuade, whereas a proof compels acceptance; (ii) arguments leave things implicit, whereas proofs make everything explicit; (iii) more information can be added to arguments, whereas proofs begin from complete information; and (iv) in consequence arguments are intrinsically defeasible. Due to these differences, arguments are usually used in contexts where proofs are inapplicable, such as in domains where information is uncertain, incomplete or implicit. This is because arguments are less tightly constrained than proofs and so allow for new information to be brought to bear on an issue and the reasoning can proceed non-monotonically.

Since the time of Aristotle, philosophers have studied argumentation in two different ways, one using the tools of deductive (formal) logic while the other is more practical and informal. Hence it is often referred to as "*informal logic*". Nevertheless, this field of study has changed dramatically in the recent years. Argumentation Theory has emerged, since the 1950s, as an area of scholarly pursuit drawing upon many other fields such as communication theory, discourse analysis and linguistics. This "*contemporary*" Argumentation Theory is distinguished from its ancient roots in its strong emphasis on the dialectical aspects of arguments rather than the traditional single person encountering a problem and reasoning about it. It is worth noting that this distinction also dates back to Aristotle who stressed that rhetoric is closely related to dialectic. Walton (1985) refers to the Aristotelian classification of argument models and distinguishes between "*demonstrative arguments"* in which premises are better known than the conclusion, so that the conclusion can be established on the basis of the premises, and "*dialectical arguments"* in which premises are presumed to be true, or thought to be true by the wise (or by some other sort of guarantee). This thesis makes use of the dialectical notion of arguments as defined and studied in modern *Argumentation Theory* as a means to inductive reasoning of the type highlighted in the previous sub-section.

### *2.1.2.1. Arguments Schemes*

One issue in argumentation theory concerns argument representation. The approach that is used in this thesis is based upon argument schemes and critical questions. Stephen Toulmin, one of the founders of contemporary argumentation theory, argued that it is impossible to divorce the criticism of "*reasoning"* and "*decision making*" entirely from the *people* giving the reasons and making the decisions (Toulmin, 1979), and so formal logics cannot be used to fully represent human reasoning. To override the limitations of formal logic, Toulmin proposed a scheme for analysing everyday arguments, referred to as Toulmin's Schema, which provided the basis for the argument schemes approach to argument representation by which arguments are presented as general inference rules whereby, given a set of premises, a conclusion can be drawn. In this sense, argument schemes are the historical descendant of Aristotle's topics (Aristotle, 1997). They are not, however deductively strict because of the defeasible nature of the underlining arguments. Such schemes have proved to be of benefit in a number of areas including informal logic and the study of fallacies and AI; in particular AI and Law as will be discussed in later sub-sections. One of the main features of Toulmin's Schema is that, in contrast to previous schemes for argument that have been based upon logical proofs consisting of the traditional premises and conclusion, Toulmin's account allows for more expressive arguments to be asserted through the incorporation of additional elements to describe the different roles that premises can play in an argument. Toulmin's Schema comprises the following three "*core*" elements (Figure 2.1):

- *The data*, considered to be a traditional premise: some fact or observation about the situation under discussion.
- *A claim*, the conclusion of the argument: some further, potentially controversial, observation, prediction or characterization.
- *The warrant*, which licenses the derivation of the claim from the data.

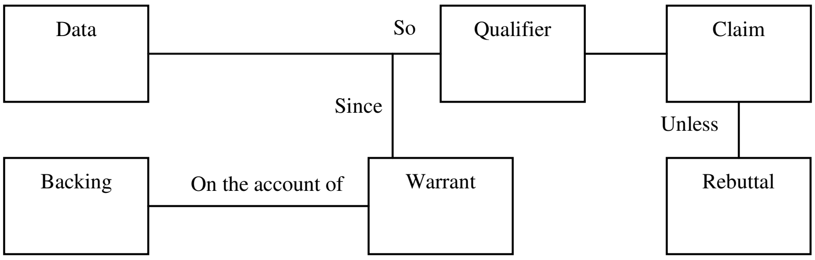


**Figure 2.1. Toulmin's Argument Schema (Toulmin, 1979).**

This Data–claim–warrant structure constitutes the inferential core of the argument. To capture the informal aspects of human reasoning Toulmin included three additional elements in his Schema:

- *A qualifier*: which gives the strength of the argument for the claim: it represents the degree of certainty for the claim.
- *A rebuttal*: a proposition that would refute the claim, if the rebuttal were to be proved true.
- *A backing*: some form of knowledge structure that represents the authority for the warrant.

Since its introduction, Toulmin's Schema has been the focus of a number of implemented systems to present arguments to the users (e.g. (Bench-Capon and Staniford, 1995), (Zeleznikow and Stranieri, 1995)). It has also been used in (Bench-Capon, 1998) as the basis of a dialogue game in which the moves relate to providing various elements of the scheme proposed by Toulmin. However, although the contribution of Toulmin's Schema has proved to be of influence, it lacks some elements that have been shown to be valuable in dealing with the precise identification of conflicts in arguments. Unlike the critical questions associated with certain argument schemes such as the ones presented by Walton (1996) and as will be discussed below; Toulmin's Schema does not provide sufficient description of the manner in which the argument can be attacked. Even though the proposed Schema accounts for rebuttals, by which claims could be challenged, it does not provide a detailed mechanism by which an opponent can explicitly attack elements of the argument. Toulmin's Schema has no room for distinguishing between different types of attack such as *rebutters* (arguments whose conclusions negate the conclusion of the original argument) and *undercutters* (counterarguments that attack the inferential link between the premises and conclusion in the original argument) identified by Pollock (1995). One significant contribution to solving this issue has been Walton's notion of argument schemes and the associated critical questions.

While Toulmin attempts to supply a general scheme for arguments, others have attempted to classify arguments in terms of various specific schemes. Walton (1996) identified some 26 argumentation schemes presented as a classification. What is interesting about Walton's account is the notion of critical questions he associates with each scheme. These questions provide the means to criticise any argument fitting the structure of the scheme, by subjecting the argument to appropriate challenges that can be identified, thus provoking consideration of the alternatives that may require consideration, and consequently prompting the best choice of argument in the given context. The asking of a question, along with its response, implies a dialectical structure in the schemes. The two devices of the scheme and the critical questions work together. The scheme is used to identify the premises and conclusion. The critical questions are used to evaluate the argument by probing into its potentially weak points that might cause the argument to default. Thus, they are used to distinguish correct from incorrect use of the scheme, similar to Swinburne's (1974) account of induction. This implies that Walton's schemes are defeasible in the sense that, in order for any argument to withstand critique, satisfactory answers must be given to any critical questions that are posed in the given situation. Additionally, such argument schemes may be contradicted by conflicting applications of the same or another scheme. For instance, a positive instance of "*Argument from Analogy*"[4] scheme can be attacked by a negative instance of the same scheme.

Note that critical questions have become linked to the more general problem of how to represent and evaluate defeasible arguments. To this end the critical question associated with any scheme provides the means to define the rebuttal and undercutter attacks (Pollock, 1995) that could be used against the arguments presented by the scheme these questions are associated with. Bench-Capon and Prakken (2006) note that Walton, in his account, classifies argument schemes according to their content. Accordingly, different schemes may be required in different domains. On the other hand, the customised set of critical questions associated with each scheme has to be considered when assessing whether the application of the scheme is warranted in a specific case or domain. Thus argumentation schemes differ from the purely logical systems in which attacks are uniform and entirely independent of content. The following sub-section presents a detailed discussion of the applications of three different argumentation schemes. One of these schemes, argument from analogy, relates to CBR, a short overview of this type of reasoning is therefore given first before listing the details of these schemes.

#### *2.1.2.2. Representing CBR by the means of argument schemes*

The intuition behind Case Based Reasoning (CBR) is that given a problem to solve humans often formulate a solution according to their previous experience: they compare new problems to be solved (*cases*) with a repository of past cases that they have solved previously (the *case base*). We often make judgements in our daily life, as we draw conclusions about given situations, on the basis of their similarity to other situations experienced in the past. The method by which we make our decision as to whether two cases are similar or not is a subject for psychology. Nevertheless, the application of CBR within the domain of AI has proved to be fruitful. One well documented application of CBR is in the context of natural language understanding (Schank, 1982). CBR has also been applied to other areas, such as legal reasoning where arguments and counterarguments are

[4] This scheme is discussed in details in Sub-section 2.1.2.2

assembled from a database of statute and legal precedent. A review of legal CBR systems is given in Sub-sub-section 2.1.4.3 below. Many definitions of CBR can be found in the literature. However, the one most related to the topic of this thesis can be found in (Aamodt and Plaza, 1994): "*CBR is the solving of a new problem by remembering a previous similar situation and by reusing information and knowledge of that situation*". In this sense CBR attempts to model the human way of reasoning by analogies to past examples.

The term "*case*" in CBR, as noted above, can denote a problem situation that was experienced beforehand, and was captured and learned so that it can be reused to solve new problems and new situations. New problems are matched against past cases held in the *case base*. Similar cases are then retrieved and used to suggest a solution to the new problems, possibly after some adaptation. CBR is thus distinguished from other forms of inference by the characteristics of the underlying process by which specific knowledge of previously experienced situations is utilised to solve new ones. This mode of inference consists basically of reasoning by analogy through the recognition of similarities or differences between two cases and carries over from the one case what can be applied, plausibly, to the other. The use of previously experienced cases to provide a solution for a new problem is also catered for by the theory presented in this thesis, as will become clear in later chapters. The proposed theory will allow software agents (entities) to jointly reason about a new case, using the information available to them, that they have previously gathered about cases from the same domain. However, the model presented in this thesis, differs from the CBR approach, and from reasoning by analogy, in that it allows agents to jointly reason about the new case, by means of argumentation. Additionally, whereas CBR generally selects a single case, the proposed theory uses generalisations from a number of cases.

The philosophical origins of reasoning by analogy can be found in Aristotle's account of reasoning by example in the *Prior Analytics* (Aristotle, 1938). This mode of reasoning, which essentially involves forming plausible hypotheses on the basis of recognised similarities, serves as a warrant for the tentative inference that what is known in the one case can be asserted to hold in the other.

Aristotle outlines the basic features of reasoning by example and distinguishes this form of inference from induction. Both rely on cases, but the function that the cases serve in each differs fundamentally. In induction the effort is to proceed from properties of individual cases to a conclusion regarding properties of the class of objects of which the cases are instances - if each swan-like bird that one has encountered happens to be white, one may argue that all swans are white. In contrast to this attempt to step from the particular to the general, in reasoning by example the objective is to argue from one particular case to another. Because two cases are similar in some respect, what is true in one may be conjectured to be true of the other. Aristotle's example is a political one: because the war of Thebes on its neighbour Phocis had led to bad consequences, it can be argued that a war by Athens on Thebes will also lead to bad consequences. Note that, induction is more useful when a large number of examples are readily available, while reasoning by analogy is more useful when only a limited number of cases are obtainable.

This Sub-section turns now to the treatment of reasoning by analogy in modern Argumentation Theory. "*Arguments from Analogy*" has been the focus of much research into argumentation theory, for which many schemes have been proposed, and on which many other forms of arguments are based, such as arguments from precedent in law (Gordon, 1995). This form of argument is the foundation of all CBR systems in which the argumentation turns on a comparison of one case to another (e.g. (Ashley, 1990), (Ashley and Rissland, 2003) and (Walton et al., 2008)). Walton (1998) proposes the following scheme for *Argument from Analogy* based on an assumption that two cases can be judged similar to each other:

*Argument from Analogy (AA) Scheme:*
*Major Premise: generally case C1 is similar to case C2*
*Minor Premise: Proposition A is true (false) in case C1.*
*Conclusion: Proposition A is true (false) in case C2.*

Walton associates four critical questions with this scheme:

- **A-CQ1**: Is A true (false) in C1?
- **A-CQ2**: Are C1 and C2 similar, in the respects cited?
- **A-CQ3**: Are there important differences between C1 and C2?
- **A-CQ4**: Is there another case C3 that is also similar to C1 except that A is false (true) in C3?

This set of critical questions lends itself nicely to the defeasible nature or arguments from analogy: A-CQ1 and A-CQ2 are redundant in the sense that they merely question whether given premises in the scheme are true or not. A-CQ4 is similar to a rebuttal attack aimed at producing a counter-analogy. A-CQ3, on the other hand, could be applied either as a rebuttal or an undercutter; depending on how important the differences between the two cases are. The significance of Walton's scheme is that it is used in some of the more sophisticated systems that have been developed in the field of AI and Law, such as CATO (Aleven, 1997) and HYPO (Ashley, 1990), both to be discussed in detail in Sub-sub-section 2.1.4.3, which provide a more exact foundation for reasoning from analogy on the basis of similarities between cases.

Arguments from analogy are closely related to arguments from classification, which are based on two main components: (i) a description of the facts or events, and (ii) their classification consequent from properties available in the description itself. Walton (1996, p.54) suggests the following scheme to cover the case of defeasible verbal classification:

> *Argument from Verbal Classification (AVC) Scheme*
> *Major Premise: if some particular thing A can be classified as falling under verbal category C; then a has property F (in virtue of such classification);*
> *Minor Premise: A can be classified as falling under verbal category C;*
> *Conclusion: A has property F.*

Walton associates two critical questions with this scheme:

- **VC-CQ1**: Does A definitely have F; or is there room for doubt?

- **VC-CQ2**: Can the verbal classification (in the second premise) be said to hold strongly, or is it a weak classification that is subject to doubt?

The above scheme, and the accompanying critical questions, is one of many similar schemes, the purpose of which is to describe the semantics of the inferential structure of arguments from classification. Walton and Macagno (2009) provide an analysis of these schemes. Moreover, Walton et al. (2008) argue that "*Argument from Analogy*" is based on "*Argument from Classification*". "*Argument from Analogy*" categorises two cases as belonging to the same class according to their similarity under a particular point of view. Meanwhile "*Argument from Classification*" leads to the conclusion that one case has a determined property, because it may be classified as generally having that property. To highlight these similarities, Walton et al. (2008) combined the previous two schemes in a new scheme, which they called a scheme for "*Argument from Analogy based on Classification*" (AAC):

*Argument from Analogy based on Classification (ACC) Scheme*
*The analogue has feature set A.*
*The case under discussion has feature set A.*
*It is by virtue of feature set A that the analogue is properly classified as W.*
*So, the case under discussion ought to be classified as W.*

A discussion of the ACC scheme is given in Chapter 3, where a number of critical questions are derived from the AA and AVC schemes to fit the ACC scheme. ACC, and the associated critical question, will then be used to derive a new argumentation scheme to cater for "*Arguing for Experience*" and address the disadvantages of the ACC scheme with respect to the process of reasoning from experience. In addition to the three schemes described above. Walton also details 24 further argument schemes (Walton, 1996), Walton et al. (2008) describe some of these schemes in more detail and introduces some new ones. Other such typologies of schemes, of varying sizes, have also been given by Kienpointner (1986), and Katzav and Reed (2004), amongst others.

It is important to emphasise that, as has been discussed earlier, reasoning by analogy differs from that of arguing by experience, in that analogies relate one case to another, while arguments from experience induces inferences from one's experience in relation to the issue at hand. Chapter 3 will present a theory for "*Arguing from Experience*" which will utilise a variation of the ACC scheme for the purpose of presenting "*Arguments from Experience*". However, elements from the theoretical foundations of CBR discussed above will be referred to later in the context of legal reasoning from past cases and its applications in AI; in particular, case-based argumentation (e.g. (Ashley, 1990)) and legal CBR (Bench-Capon, 1997). These elements have motivated a large block of the work presented in forthcoming chapters. However, Walton views argument schemes as a way of representing arguments embedded within dialogues. Together with Krabbe, Walton has provided a typology of the different dialogues that can be used in human communication (Walton and Krabbe, 1995). The following provides a review of this well know typology of dialogues.

### *2.1.2.3. Walton and Krabbe's Typology of Dialogues*

A *dialogue* is an exchange of speech acts amongst a number of participants in some sequence aimed at achieving a collective goal. The dialogue is *coherent* to the extent in which individual speech acts fit together to contribute to the *dialogue goal*. Thus, in coherent dialogues, utterances are allowed only if they further the goal of the dialogue in which they are made (Carlson, 1983). For instance, during the course of a persuasion dialogue only utterances that contribute to the resolution of the conflict that triggered the dispute are allowed to be made. Walton and Krabbe (1995) have identified a number of distinct dialogue types used in human communication: Persuasion, Negotiation, Inquiry, Information-Seeking, Deliberation and Eristic Dialogues. This typology has proved to be influential in the study of argumentation theory and its application to AI. Table 2.1 summarises the six ideal dialogue types in this typology. Walton and Krabbe base their categorisation upon: (i) the information available to each participant at the commencement of a dialogue, of relevance to the topic of discussion, (ii) the participant's individual goals for the dialogue, and (ii) the collective goal of the dialogue. Note that Walton and Krabbe distinguish between the goal of the dialogue, as a purpose of a type of conversation and the goal of each of the parties involved in the dialogue. The importance of this difference is discussed later.

| Type | Initial Situation | Main Goal | Participants Aims |
|---|---|---|---|
| **Persuasion** | Conflicting view points | Resolution of conflicts by verbal means | Persuade the other(s) |
| **Negotiation** | Conflict of interest and need for cooperation | Making a deal | Get the best out of it for oneself |
| **Inquiry** | General ignorance | Growth of knowledge and agreement | Find a proof or destroy one. |
| **Info-seeking** | Personal ignorance | Spreading knowledge and revealing positions. | Gain, pass on, show or hide personal knowledge. |
| **Deliberation** | Need for action | Reach a decision | Influence the outcome |
| **Eristic** | Conflict and antagonism | Reaching and accommodation in relationship | Strike the other party and win in the eyes of onlookers |

**Table 2.1. The six types of dialogues identified in (Walton and Krabbe, 1995).**

The Walton and Krabbe descriptions may be summarised as follows:

- **Persuasion dialogues** involve one participant seeking to persuade another to accept a statement they do not currently endorse. Here, a primary obligation is the burden of proof: a weight of presumption set for practical purposes to facilitate the successful carrying out of the obligations of the participants during the course of the dialogue. The device of burden of proof is useful because it enables discussion to come to an end in a reasonable time. If the participants are guided only by the force of argument, then whichever participant has the more convincing argument, taking into account the burden of proof, should be able to persuade the other to endorse the statement at issue, within finite time.
- **Negotiation dialogues** occur when two parties bargain to jointly divide some scarce resource, where the competing claims to this resource cannot all be satisfied at the same time. Negotiations require some level of co-operation between the involved parties. However, at the same time, each participant is assumed to be seeking to achieve the best possible deal for

themselves. If a negotiation dialogue terminates with an agreement, then the resource has been divided in a manner acceptable to all participants.

- **Inquiry dialogues** involve two participants collaborating to answer some question whose answer is not known to either participant; thus the participants in the dialogue will jointly seek to determine the desired answer. In contrast to persuasion, inquiry does not commence from a position of conflict, as here participants have not taken a particular position on the issue at question; they are trying to find out some knowledge.
- **Information-Seeking dialogues** are those where one participant seeks the answer to some question(s) from another participant, who is believed (perhaps erroneously) by the first to know the answer(s). The first party seeks to obtain the answer from the second by means of the dialogue.
- **Deliberation dialogues** occur when two parties attempt to decide on a course of action in some situation. This course may be performed by one or more of the parties in the dialogue or by others not present. Here the participants share a responsibility to decide the action(s) to be undertaken in the circumstances. As with negotiation dialogues, if a deliberation dialogue terminates with an agreement, then the participants have decided on a mutually-acceptable course of action.
- **Eristic dialogues** happen when participants quarrel verbally as a substitute for physical fighting, aiming to vent perceived grievances.

The above typology is often criticised for a number of shortcomings. Mainly, that most human, and to some extent agent, conversations involve combinations of these six types of dialogue, rather than a single type. For example, a conversation between a bookstore keeper and a potential client may commence with the client *seeking information* about a particular subject, the keeper will happily answer the client’s enquiries; but at some point the dialogue will shift to a persuasion dialogue, in which the keeper will try to persuade the client to buy a certain book which he/she believes contains information of interest to this client. The dialogue may then shift further, once the client is persuaded of the potential of buying a particular book, to a negotiation on how much the book will cost, with both parties attempting to get their “*best*” price. The two parties in this example may or may not be aware of the different nature of their discussions at each phase, or of the transitions between phases. Walton and Krabbe (1995) refer to instances of the atomic dialogue types which are contained within other dialogue types as "*embedded*". A study of embedded dialogues in the context of computational models for argumentation can be found in (Reed, 1998). Shifts between negotiation and persuasion were also discussed in (Wells and Reed, 2006).

Another criticism is that Walton and Krabbe's typology accommodates two participants only: a proponent and a respondent, each takes turns in making moves that represent speech acts like asking a question or putting forward an argument. There is no room in this proposal for third parties such as moderators or referees who ensure that procedures are followed or decide the outcome. Similarly, their typology does not explain how the dialogues proceed in cases where they involve more than two participants. Chapters 6 and 7 will present a solution, which fits with the theory presented in Chapter 3, by which any number of participants can engage in a persuasion dialogue. A third criticism is that although Walton and Krabbe distinguish between the *goal* of the dialogue type and the personal *goals* of the participants taking part in the dialogue, in reality only participants can have goals. The participants may believe that a dialogue they enter has some purpose, but their own goals or the goals of the other participants may not be consistent with this purpose. For example, in persuasion dialogues, participants may enter for the sake of arguing, without the intention of being persuaded by the other participants. Additionally, Walton and Krabbe themselves do not claim their typology is comprehensive, and some recent research has explored other types (e.g. (Cogan et al, 2006)).

Despite these criticisms The Walton and Krabbe's typology remains very influential in argumentation and its applications to AI. The work presented in the forthcoming chapters focuses on one dialogue type in particular: persuasion dialogues. The emphasis is on how persuasion may happen because of differences in the experience of participants regarding the subject matter or due to the fact that one participant has gained more experience than the others. This concludes the discussion of argumentation theory in philosophy. The next sub-section will consider the modelling of the dialectical approach to argumentation, such as Walton and Krabbe's, by means of dialogue games.

### 2.1.3. Dialogue Games for Argumentation

This sub-section examines the field of dialogue games and its application to AI. In particular, it presents an overview of dialogue games for argumentation and the related dialogue systems.

#### *2.1.3.1. Overview*

Recently, formal dialogue games have attracted the attention of researchers as a means to facilitate agent communications that allows for sufficient flexibility of expression. Dialogue games are rule-governed interactions between two or more players (agents), where each player "*moves*" by making utterances, according to a defined set of rules known as "*dialogue game protocols*". Each move has an identifying name associated with it and comprises some statement (represented in a suitable language) which contributes to the dialogue. Such moves are exchanged by participants until the dialogue terminates, according to some *termination rules*. Although the roots of dialogue games date back to at least the time of Aristotle (e.g. (Aristotle, 1997)); they have been the focus of some more recent research in philosophy to study fallacious reasoning (e.g. (Hamblin, 1970) and (MacKenzie, 1979)) and computational applications such as human-computer interaction (Bench-Capon et al., 1991). In AI, dialogue games have been applied to modelling complex human-like reasoning (e.g. (Prakken and Sartor, 1998), (Moore, 1993) and (Bench-Capon, 1998)). Of note regarding the application of dialogue games to AI is that these games differ from games of game theory (as applied in economics) in that the payoffs for winning or losing a game from the latter are not considered in the former. Several proposals for formal dialogue games have been presented for most of the atomic dialogue types in the dialogue typology of Walton and Krabbe (1995). A number of formal dialogue game protocols have been proposed to model persuasion dialogues (see (Prakken, 2006) for an extensive survey), negotiation dialogues (e.g. (McBurney et al, 2003)), inquiry dialogues (e.g. (McBurney and Parsons,

2001) and (Black and Hunter, 2007)) and information-seeking dialogues (e.g. (Hulstijn, 2000)). Other formal dialogue game systems have been proposed to model more than one atomic type of dialogue. For instance, Amgoud et al. (2000a) presented an argumentation-based formal two-party dialogue game protocol; their system was developed to handle inconsistent information and supports persuasion, inquiry and information-seeking dialogues. Amgoud et al. have also subsequently proposed an extension of their dialogue game in (Amgoud et al., 2000b) with additional locutions to support negotiation dialogues. Several formalisms have also been suggested for computational representation of combinations of dialogues. One notable example is Reed's (1998) *Dialogue Frames*, which enabled iterated, sequential and embedded dialogues to be represented; other examples include ((Miller and McBurney, 2007) and (McBurney and Parsons, 2002)).

Treating dialogues as abstract games makes it possible to develop formalisms for the modelling of dialogues between autonomous agents. Complex dialogues, including dialogues embedded in one another, can be represented in the formalisms as sequences of moves in a combination of dialogue games. One particular example of such formalisms can be found in the work of McBurney and Parsons (2002). Their formalism can represent different types of dialogue in the standard typology of Walton and Krabbe (1995) and it has three levels:

- At the lowest level are the topics which are the subjects of dialogues.
- At the next level are the dialogues themselves (represented by means of formal dialogue games).
- At the highest level control dialogues are represented which enable the agents to decide on which dialogues to enter, if any.

McBurney and Parsons (2002) suggest that formal dialogue games comprise the following components:

- **Commencement Rules**: define the circumstances under which the dialogue commences.
- **Locutions**: the utterances that are permitted at every stage of the dialogue.

- **Combination Rules:** describe the dialogical contexts under which particular locutions are permitted or not, or are obligatory or not.
- **Commitments:** define the circumstances under which participants express commitment to a proposition.
- **Rules for Speaker Order:** define the order in which players (agents) taking part in the dialogue game may make utterances. These rules are of particular importance in multiparty dialogues (comprising more than two players); as here players may either speak at any time, or there are rules regarding turn taking. Chapter 6 will return to this issue in the context of multiparty "*Arguing from Experience*".
- **Termination Rules:** define the conditions under which the dialogue ends.

The dialogue model proposed in this thesis caters for commitments which feature in a number of standard dialogue games. Commitments are often incorporated into dialogue games via commitment stores which derive from Hamblin's (1970) study of fallacious reasoning. The promoted dialogue model incorporates a different notion of commitment by which participants engaging in a dialogue game are committed to achieve their assigned goals, which follows the multi-agent sense of commitment which is often regarded as a persistent goal that the agent is trying to achieve (Cohen and Levesque, 1990).

In summary, dialogue games have proved to be a helpful method to model and reason about exchanges of information, presented by a pre-determined set of utterances, between a number of participants, in a number of different domains, including law, philosophy and AI. Moore (1993) presents a comprehensive discussion of Dialogue Game Theory in general, along with a number of examples of dialogue games and systems. Chapters 4 and 6 of this thesis return to the issue of dialogue games, where two-party and multiparty dialogue game protocols are proposed, respectively, to enable "*Arguing from Experience*". Both protocols follow a general process for reasoning from experience as embodied in the theory presented in Chapter 3.

### *2.1.3.2. Formal Systems for Persuasion Dialogues*

Persuasion Dialogue, as discussed earlier, allows for one participant, the *proponent*, to attempt to persuade another party: the *opponent*, that some particular proposition is true, using arguments that show or prove that the proposition holds. To this end, two distinctive types of persuasion dialogues can be identified: (i) "*Dispute"* (Walton, 1998) in which both participants have a positive burden of proof, thus each will try to persuade the other that their thesis is true while the other party's proposition is false, (ii) "*Dissent*" (Prakken et al., 2005), is more flexible as the burden of proof rests only with one party. Thus the party holding the positive burden of proof will start the dialogue by proposing an argument asserting her thesis. The other party may take the position of criticising this argument without actually committing themselves to the opposite proposition. In general, persuasion dialogues have a distinctive feature in that a participant's arguments are assumed to have as premises propositions that the other parties are committed to. This means that argumentation in a persuasion dialogue is an interactive process in which each party's arguments are always directed towards the other party and are based on premises that the other party is committed to. Persuasion dialogues have tended to presuppose that the agents have a rule-like representation of their knowledge. A thorough survey of a number of systems can be found in (Prakken, 2000, 2006). In this work Prakken identifies the speech acts typically used in such dialogues[5]:

- *Claim P* (assert, statement ...). The speaker asserts that P is the case.
- *Why P* (challenge, deny, question ...). The speaker challenges that P is the case and asks for reasons why it would be the case.
- *Concede P* (accept, admit ...). The speaker admits that P is the case.
- *Retract P* (withdraw, no commitment...). The speaker declares that they are not committed (anymore) to P. Retractions are real retractions if the speaker is committed to the retracted proposition, otherwise it is a mere declaration of non-commitment (e.g. in reply to a question).

[5] Prakken (2006) also notes that, regarding the structure of the dialogue, participants in persuasion dialogues may return to earlier choices and present alternative replies. Also participants may postpone their replies, sometimes even indefinitely.

- *P since S* (argue, argument ...). The speaker provides reasons why P is the case. Some protocols do not have this move but require instead that reasons be provided by a claim P or claim S move in reply to a why move (where S is a set of propositions). Also, in some systems the reasons provided for P can have structure (e.g. a proof tree or a deduction).
- *Question P* (...). The speaker asks another participant's opinion on whether P is the case.

These moves presuppose that the participant's knowledge is organized in a certain way, namely as a set of facts and rules (typically some strict and some defeasible) of the form *fact →conclusion*. Thus *why P* seeks the antecedent of a rule with *P* as consequent; *P since S* volunteers the antecedent of some rule for *P*, and the other questions suggest the ability to pose a query to a knowledge base of this sort. Prakken's own instantiation of this framework (Prakken, 2000) presupposes that the participants have *belief bases* comprising facts, defeasible rules, and priorities between rules. That the participants are presupposed to be equipped with such belief bases doubtless derives in part from the context in which these approaches have been developed. The original example of the approach was probably Hamblin (1970) who was interested in exploring a particular logical fallacy. The take up in Computer Science has largely been by those working in knowledge based systems and logic programming, where the form of the belief base is a natural one to assume. The result, however, is that the debate takes place in a context where the participants have knowledge (or at least belief), and the dialogue serves to exchange or pool this knowledge. Given these assumptions persuasion takes place in the following ways:

- One participant supplies the other with some fact unknown to that participant, which enables the claim to be deduced;
- One participant supplies the other with some rule unknown to that participant, which enables the claim to be deduced;
- An inconsistency in one participant's belief base is demonstrated, so that a claim or an objection to a claim is removed.

At least one of these must occur for persuasion to happen, but in a complicated persuasion dialogue all three may be required. This necessitates certain further assumptions about the context: that the beliefs of the participants are individually incomplete or collectively inconsistent. Although the participants have knowledge, it is defective in some way, and corrected or completed through the dialogue. Importantly the participants will have formed a theory of the domain, and so will have systemized their experience into what might be termed knowledge, or have been taught a theory. The formal description of this model can be found in Prakken's work (2000, 2005b and 2006). Prakken's formalisation is used in later chapters of this thesis as basis for the promoted formalism for "*Arguing from Experience*". While persuasion dialogues of the form modelled by Prakken do take place in practice, this thesis seeks to model a different style of persuasion dialogues, involving the sharing not of *knowledge*, but of the *experience* itself. In this situation the participants have not analysed their experiences into rules and rule priorities, but draw directly on past examples to find reasons for coming to a view on some current example.

### 2.1.4. Argumentation in AI

Over the last decade, argumentation has gained growing recognition as a promising research direction in Artificial Intelligence (AI) in that it provides means by which uncertain and incomplete information can be reasoned about. Incorporating elements from the theories of argumentation into AI applications has the obvious advantage of allowing these systems to make use of uncertain or incomplete knowledge available to them. Fields such as: multi-agent systems, machine learning and legal systems have benefited the most from argumentation, as in these domains there exists a need for decision making based on incomplete or uncertain information. In such situations argumentation can play the important role of providing tentative conclusions for or against a claim in the absence of further information to the contrary. To this end, the models of argumentation for AI require some method by which the relative worth of the arguments relevant to a particular debate can be assessed and evaluated; thus determining which arguments are the most convincing in a particular context. One such method has been proposed by Dung (1995) in which an argument for a claim is accepted or rejected on the basis of how well it and other available arguments can defend it against other arguments that attack and potentially defeat it. The system proposed by Dung is an example of an abstraction mechanism to represent the process of argumentation in AI called an "*Argumentation Framework*". Dung's proposal of such frameworks has proved to be a particularly influential formal system of defeasible argumentation, and has provided the basis for much of the subsequent work in the areas of representing and evaluating arguments in the context of AI. Dung (1995) defines an Argumentation Framework as a finite set of arguments *X*, and a binary relation between pairs of these arguments called an attack[6]. These relationships form a directed graph showing which arguments attack one another. However, Dung's model does not concern itself with the internal structure of the arguments. Instead the status of an argument can be evaluated by considering whether or not it is able to be defended from attacks from other arguments with respect to a set of arguments $S \subseteq X$. Dung (1995) provides the semantics of such argumentation frameworks through the notion of a *preferred extension*. A number of extensions and variations to Dung's model have been proposed (e.g. (Vreeswijk and Prakken, 2000) and (Cayrol et al., 2003)). One notable example is the "*Value-Based Argumentation Frameworks*" of Bench-Capon (2003), which includes the audience's values in the analysis of the acceptability of arguments, thus enabling distinctions to be made between different audience's preferences and so allowing the distinction between attack and defeat. Dung's formalism, and the subsequent variations, proved to be a milestone in the application of argumentation in AI. However, the work undertaken in this thesis makes little use of these contributions, since attacks are always successful and form a simple tree structure, and so no further consideration is given to the notion of argumentation frameworks in this chapter or subsequent chapters.

[6] Dung originally termed this relation "*defeat*". In Dung's system all attacks succeed and so can be said to be defeats, but in some developments of Dung's framework defeat is reserved for a successful attack, and we will therefore use "*attack*" for the relation.

#### *2.1.4.1. Argumentation for Agents Interaction*

The application of argumentation to multi-agent systems was first introduced by Parsons et al. (1998). Since then, there has been an increasing interest in making use of argumentation in different areas within multi-agent systems. Two sub-fields of agency have benefited the most from introducing concepts from argumentation theory to them: (i) agent reasoning and (ii) agent interaction. The application of argumentation to agent reasoning involves finding an adequate formalisation of an agent's knowledge to perform defeasible inferences in a computationally effective way. Examples of the application of argumentation to agent reasoning include: common sense reasoning (Chesñevar et al., 2000), practical reasoning (Atkinson, 2006), and normative reasoning (Oren et al., 2008). On the other hand, argumentation provides a means for "*social*" interaction amongst agents, and thus enhances the social capability of these agents; social interaction is important if agents are to fully achieve their assigned objectives, and be classified as intelligent (Wooldridge, 2001). With respect to agent interactions, argumentation has been shown to be an adequate method to design agent communication languages and frameworks (e.g. (Dignum et al. 2001) and (Reed, 1998)). However, the main block of research in argumentation-based agent interaction has been centred on the design of negotiation models for agents. The importance of negotiation, to multi-agent systems, comes from the fact that it provides a mechanism to facilitate conflict resolution. Such conflicts often take place in agent interaction because each of them has different goals and interests and maintain different knowledge. Argumentation provides a natural means to model negotiation because it follows the way conflicts are resolved in everyday life through the exchange of reasoned argument and justification of a stance. One notable approach to efficiently incorporate argumentation in negotiation has been pioneered by Rahwan *et al.* (2004) and is known as *argumentation-based negotiation.* The intuition behind which is that the likelihood and quality of an agreement amongst agents may be increased if they are to exchange arguments which influence each others' states.

The application of argumentation to agent interactions has proved to be an influential field of research. An important aspect regarding this, closely related to the work described by this thesis, is the strategy design that has been incorporated within some of the argumentation-based agent interaction systems. Different designs have been proposed to model strategies for argumentation-based agent interaction. An argumentation strategy enables an agent to select which argument to put forth, from the different possible arguments at every stage in the argumentation dialogue, in order to achieve the agent's goal (objective), taking into consideration the circumstances of the dialogue. Thus the "*strategy problem*" in these settings (a number of autonomous software agents arguing with each others) is concerned with "*enabling an agent to argue well*": while the rules of a protocol permit the agents to argue legally, the strategy is needed for them to argue well. One approach to the problem of strategy, proposed by Oren et al. (2006), applies a number of heuristics to assign a utility cost to various elements of the argument, in particular, the amount of information revealed by the argument. Individual agents then attempt to maximise this utility. This approach is closely related to the belief-based view of argumentation, and thus is not readily applicable to the model suggested in this thesis in which arguments are drawn from the agent's past experience rather than a handcrafted knowledge based. The strategy model of this thesis follows that of Moore (Moore, 1993). In his work with the DC dialectical system (based on *DC* the philosophical dialogue game of (MacKenzie, 1979)), Moore concludes that an agent's argumentation strategy is best analysed at three levels:

- Maintaining the focus of the dispute.
- Building its own point of view or attacking its opponent's and
- Selecting an argument that fulfils the objectives set at the previous levels.

The first two refer to the agent's strategy - the high level aims of the argumentation. The third level refers to the tactics - the means to achieve the aims fixed at the strategic levels. Moore's requirements form the basis of most other research into agent argumentation strategies.

Amgoud and Maudet (2002) suggest a computational system to capture some of the heuristics for argumentation suggested by Moore. This system requires a *preference* ordering over all the possible arguments, and a level of *prudence* to be assigned to each agent. The strength of an argument is defined according to the complexity of the chain of arguments required to defend this argument from the other arguments that attack it. An agent can have either a "*build*" or a "*destroy*" strategy. By applying a build strategy, an agent tries to assert arguments the strength of which satisfies its prudence level. If this fails, the agent switches to the destroy strategy, whereby the agent will consider any possible way to attack the opponent's arguments. One drawback of this approach is that computational limits may affect the agent's choice. However, this particular strategy model makes use of the underlying notion of an agent's profile, which will also be incorporated into the strategy model promoted by this thesis. Different notions of agent profiles have been proposed in the literature. The one most related to this thesis, is the one suggest by Amgoud and Parsons (2001), who propose five different profiles of dialogues to discriminate between different classes of agent types with varying degree of "*willingness to cooperate*" in the attitude of an agent. These profiles follow the rule-based representation of agents' knowledge and the speech acts associated with typical argumentation dialogues (as identified in (Prakken, 2006)). Chapter 4 will return to Amgoud and Parsons' notion and discuss it in more detail. The strategy model of Kakas et al. (2004) also makes use of the same notion of agent profiles within a three layer system for agent strategies in argumentation. The first contains "*default*" rules, of the form condition → utterance, while the two higher layers provide preference orderings over the rules. Assuming certain restrictions on the rules, they show that only one utterance will be selected using their system, a property they refer to as determinism. While their approach is able to represent strategies proposed by a number of other techniques, it does require hand crafting of the rules. Also, no suggestions are made regarding what a "*good*" set of rules would be. The account, promoted by this thesis, incorporates different elements from the systems discussed above in its strategy model to accommodate for the different aspect of "*Arguing from Experience*".

#### *2.1.4.2. Argumentation and Machine Learning*

Another area in which argumentation has attracted some attention is machine learning (e.g. (Mozina et al, 2005) and (Ontañón and Plaza, 2006)): a research field concerned with the construction of algorithms that automatically improve with experience. Machine learning algorithms allow for the detection and extraction of interesting data patterns for a variety of problems; yet most of these algorithms provide an output based on quantitative evidence, whereas the inference process which led to this output is often unknown (Gómez and Chesñevar, 2004). By integrating argumentation with existing machine learning techniques the inference model for the latter can be catered for. A number of different approaches have been proposed to integrate argumentation and machine learning. Governatori and Stranieri (2001) investigate the feasibility of KDD in order to facilitate the discovery of defeasible rules for legal decision making. In particular they argue in favour of Defeasible Logic as an appropriate formal system in which the extracted principles should be encoded in the context of obtaining defeasible rules by means of induction-based techniques. This thesis presents an approach to argumentation related to that of (Governatori and Stranieri, 2001) and bridges the gaps in their proposal (e.g. their technique can operate only on small datasets). More importantly, the promoted model offers a more efficient means to exploit databases for the production of "*Arguments from Experience*" as will be discussed in later chapters. Gómez and Chesñevar (2004) list a number of proposals to integrating argumentation and machine learning. One particular area in their account concerns building arguments from stored data to explain unseen instances. The work described in this thesis provides means for automatic formation of arguments on the basis of past experience presented by data records, and then applied to classify unseen records from the same domain. This work "*borrows*" elements from machine learning and argumentation and incorporates them into a model to enable "*Arguing from Experience*".

### *2.1.4.3. Argumentation in AI and Law*

The application of argumentation to one classic field of AI research, namely AI and Law, has proved to be most rewarding. This is due to the central role arguments play in the process of law; where legal disputes result from disagreements between two (or more) parties. These disputes are then resolved by each party presenting arguments for their position to a third party (e.g. judge or jury). This third party will evaluate the arguments put forward by the parties in dispute to come to a conclusion with respect to the case at hand. Often this decision is itself justified by an argument as to why the arguments of one side should be preferred. Much early research in AI and law was centred on rule based reasoning: the application of a proof model which involved the representation of legal knowledge in the form of First Order Logic from which legal consequences could be deduced. One notable example was developed by Gardner (1987) in the field of "*offer and acceptance*" in American contract law. The focus of this work was what happens "*when the rules run out*" and it drew attention to the fact, well-known in law, that one cannot reason by rules alone, which often either fail to cover every case or conflict, but rather often one examines examples in response to many situations. Argumentation was proposed by Bench-Capon and Sergot (1989) to solve the problems associated with the proof model. Mainly that formalisation of legal knowledge typically involves a degree of interpretation, thus several competing theories often emerge; on the other hand, the inescapable defeasibility of legal rules led to conflicts and gaps in the coverage. However, while progress continued on rule-based reasoning (RBR) systems (e.g. (Gordon, 1991), (Prakken, 1993) and (Hage, 1996)), another strand of work existed within AI and Law focusing on reasoning with cases and analogies and applying elements from CBR into AI and Law. One of the first projects in AI and Law, the TAXMAN project (McCarty and Sridharan, 1982) had as its goal providing a computational means of generating the majority and minority opinions in a celebrated case in the American corporate tax law. This system was intended to produce analysis of the tax consequences of a given corporate transactions. Another early example is The HYPO program (e.g. (Ashley, 1990)).

Since the introduction of CBR in the above examples, its application to AI and law has become the focus of much research. This is because while most AI and Law systems recognise the importance of precedent cases as a source of legal knowledge; rule based systems do not make a direct use of such cases; rather they extract the rationales of the past cases and encode them as a set of rules. To be applicable to a new case the extracted rules may require some re-processing to match the new facts. In consequence, the CBR approach to AI and Law has attempted to avoid using rules altogether, instead representing the input cases, often interpreted as a set of "*factors*"[7], and the decisions of these cases. Additionally, a number argument moves for interpreting the relation between the input case, the precedent cases and decision were devised (e.g. (Ashley, 1990) and (Aleven, 1997), both to be discussed below). These moves are particularly relevant to this thesis as they formed the inspiration for the moves in the "*Arguing from Experience*" protocol that will be proposed later. The CBR approach to legal reasoning catered for "*factor-based domains*" (Branting, 2003), in which problems could be solved by considering a number of factors that plead for or against a verdict. The representation of cases in these domains thus comprised a set of these factors. Therefore, the main source of conflict in "*factor-based domains*" is that a new case often does not exactly match a precedent on all its factors but will share some features with it, lack some of its other features, and/or have some additional features. Moreover, cases are more than simple rationales: matters such as the context and the procedural setting can influence the way the case should be used. The following illuminates the process of CBR as applied to legal argumentation, before surveying some of the more significant systems for arguing from precedent cases.

Bench-Capon (1997) argued that CBR as modelled in HYPO and its progenies was an extension of the original models of CBR (e.g. (Schank, 1982)). Because the application of CBR in AI and Law is a system in which the output is an argument and not simply a past case that is similar to current situation. This model of CBR required as an input the side to argue for. Besides, CBR in AI and Law retrieve and deploy cases that best suit the case under discussion and the view point of the sides of the argument. This means that these cases are not determined based on the notion of similarity alone, but rather by the role they can play at given points in the argument. Bench-Capon (1997) made use of this definition to identify how past cases are used by CBR systems, and to

[7] The term "*factor*" was adopted in AI and Law (e.g. (Ashley, 1990)), partly because it was a term more familiar to those in the legal community. For instance, BankXX had a "*domain factor space" in which a case was represented by a vector consisting of its magnitudes on each* "*domain dependent factor or dimension*"" that applies to it from the two dozen or so used by BankXX (Rissland et al., 1996).

distinguish these cases from the other uses of cases in Law, such as case retrieval which is more related to information retrieval systems. Bench-Capon defined the requirements of CBR systems in law as follows:

- A position to argue for.
- A structure for a case based argument, determining a variety of moves.
- Consideration of cases with reference to the argument moves they support.

Note that legal CBR is not just retrieving a past case that is most similar to a new case, but rather retrieving a case that gives a presumptive reason for applying the decision, which is then subjected to "*argument*" to determine whether it should be followed or not. The three-ply argument structure, a rather distinctive feature of these systems introduced by HYPO, is of major importance to this thesis. Here the proponent cites a past case similar to the case under discussion. The opponent then argues against this case by distinguishing it from the current case or presenting counter examples (and therefore undermining the proponent's claim that the original case should be followed). The original proponent also has the opportunity to distinguish these counter examples in a rebuttal phase. The analysis underlying these systems, with its patterns of citation, distinguishing and counter example will provide a starting point for the model for "*Arguing from Experience*" presented in later chapters of this thesis. In the following some systems for reasoning from precedents as applied to AI and Law are discussed. A number of observations with respect to these systems are made that are instrumental to the theory for "*Arguing from Experience*" that Chapter 3 presents. This theory aims to make use of elements of legal CBR, especially the argumentation structure some of these systems apply.

Perhaps the most influential AI and Law system to make use of elements of CBR is the HYPO system (e.g. (Ashley, 1990)), introduced above. Originally developed by Edwina Rissland and Kevin Ashley in the domain of US Trade Secrets Law, HYPO was the first such CBR system to be ever developed. HYPO attempts to construct an argument which can be advanced concerning a new case, and not to make decisions or to take actions: it is concerned with justifying a conclusion about a problem by drawing an analogy to similar past cases, then arguing that the problem under discussion should be treated in a similar manner. Cases are stored in a case knowledge base and are represented using dimensions, which are essentially stereotypical fact situations relating to the legal issues from the legal domain of HYPO. HYPO makes use of "*dimensions*" to focus on the knowledge representation methodology for representing factors. "*Factors*", on the other hand, focus on the object (entity) to be represented: the stereotypical patterns of facts that tended to strengthen or weaken a side's legal claim (Ashley 1990). Each dimension represents a factor and encodes knowledge about it. For instance, the factors that the secret can or cannot be re-engineered are represented by a dimension which represents the degree of ease with which the secret can be re-engineered. The factors thus represent the extreme pro-defendant and pro-plaintiff points on the dimension. Later, as described below, Ashley (1990) used factors also to refer to the simplified dimensional representations employed in CATO (e.g. (Aleven, 1997)). A useful account contrasting factors and dimensions can be found in (Rissland and Ashley, 2002). Every dimension has a value representing its strength and a direction indicating the side the factor favours (opponent or proponent). The structure of these dimensions allows HYPO to decide if a dimension applies to a case or not. They also make the new case more or less favourable. HYPO analyses the current case, generates the current fact situation, and finds all the applicable dimensions by checking the prerequisites of the dimension and comparing them to the factual predicates of the current fact situation. A dimension is applicable if and only if no prerequisite is unknown or negated. As described above, HYPO applies these dimensions to construct the *three-ply arguments structure*; which consists of arguments supporting a proposed solution, responses opposing those arguments, and a rebuttal. This structure is achieved by retrieving the legal pros and cons of the issues raised in the fact situation of the case under consideration. These "*pros and cons*" are then used to argue in support of the claim or to make counter-arguments. This three-ply structure is identified as follows:

- **State Point:** The proponent analogises a precedent case to the current fact situation and makes the claim that the court should find for them. The precedent must have some dimensions favouring the proposition advocated by the proponent and they can be ranked according to some order.

- **Respond:** The opponent responds, either by citing a counter example or by distinguishing the cited case. Counter examples are cases with a different outcome that have at least as many similarities with the case under consideration than a case previously stated by the other side. Distinguishing a case in HYPO is simply highlighting dimensions present in the case cited by the other side, but missing from the case under consideration, which strengthen the precedent case.
- **Rebut:** The cycle turns back to the proponent who tries to distinguish any counter examples cited by the opponent.

The model for "*Arguing from Experience*", presented in Chapter 3, is directly inspired by the HYPO- based model described above.

Another major system for reasoning from legal precedents is CATO (most fully reported in (Aleven, 1997)), originally designed to help law students to reason with past precedents by generating examples of arguments based on such reasoning, and to enable them to explore the underlying structure of the arguments produced by the system. Ashley and Aleven (1991) argue that dimensions (as applied in HYPO) often relate to each other and to higher-level legal reasons (abstract factors). For this reason they have introduced the notion of "*factors*" to replace HYPO's dimensions as a means to index cases in CATO – a case was represented simply as a set of applicable factors. These factors symbolise the factual strengths and weaknesses of cases: the presence of a factor in a case makes it stronger on one side and weaker on the other (each of the dimensions in HYPO, as well as some newly identified factual patterns, were labelled as being either pro-plaintiff or pro-defendant factors). However CATO's factors do not cater for any criteria by which one can tell to what degree they strengthen or weaken the position of any of the sides. A distinctive feature of CATO is that these factors are organised into a hierarchy of increasingly abstract factors, so that several different factors can be seen as meaning that the same abstract factor is present. There could be several layers of abstract factors, until parentless nodes (issues) are reached. This hierarchy allows for additional arguments that interpret the relation between an input case and its decision, such as emphasising or downplaying distinctions. The argument model of CATO consists of eight moves (Bench-Capon, 1997), including additional moves that do not feature in HYPO; the new moves relating to the abstract factors and the factors hierarchy which does not exist in HYPO. The progeny of HYPO extends to cover many systems other than CATO such as CABARET (Rissland and Skalak, 1992), BankXX (Rissland et al., 1996) and IBP (Brüninghaus and Ashley, 2003).

From the systems mentioned above, one notable example is IBP (Brüninghaus and Ashley, 2003). While Hypo and CATO (and CABARET) identify but do not resolve conflicting arguments, IBP provides a means of adjudicating between conflicting arguments. IBP is an adaption of CATO for predicting outcomes that combines reasoning with an abstract model and CBR techniques to predict the outcome of case based legal arguments, and to provide an explanation of this prediction. IBP, however, separates arguments by issues so that that conflicting arguments can be identified separately for each issue, instead of using CATO's factors. A distinctive feature of IBP is the "*logical model*" of the domain, which results from domain analysis intended to identify and organise any "*intermediate predicates*". This analysis is at a high level and does not require the consideration of individual cases. Chapter 4 will return to this notion of a logical model in the context of arguing about intermediate concepts, in which logical models are incorporated in the proposed model to set agenda for the dialogues.

### 2.1.5. Summary of Argumentation in Philosophy and AI

This section has covered the essential research upon which the theoretical model for "*Arguing from Experience*" has been based. Chapter 3 will discuss the fact that this type of argument borrows elements from a number of areas; in particular inductive reasoning as a means to infer from past experience to unprecedented situations, and reasoning from analogy which judges if two cases are similar or not. This section has also discussed the treatment of argumentation in philosophy and its application to AI. An overview of dialogue games as formal means to argumentation representation was also given, and thus bridging the theory and the application of argumentation. The prior research presented in this literature survey provides numerous key points that will be taken forward by the work that will be presented in the forthcoming chapters of this thesis. These key points are summarised below:

- The account of reasoning from experience presented in this thesis is intended to accommodate numerous distinct features of inductive reasoning, as identified in the literature. In particular, this form of reasoning is undertaken in the context of a debate which incorporates a set of arguments for and against judging that a particular situation should be associated with a certain conclusion. The proposed model aims at exploiting differences in the experience of participants engaging in the debate to aid reaching a resolution of the debate. This thesis argues that this form of arguing is a distinctive mode of reasoning
- The view of argumentation that will be used in this thesis follows that of Walton (1996) who has given a proposal for treating argument as presumptive justification subject to critical questioning. This is manifest through the notion of argument schemes and characteristic critical questions. In particular, a number of schemes were discussed, in the context of arguing by analogy and from classification. One particular scheme that integrating both forms of arguments has inspired the scheme for "*Arguing from Experience*" presented in Chapter 3.
- In order for structured and meaningful dialogue to take place, a number of proposals have been given for dialogue game protocols that are designed to facilitate the conduct of particular types of dialogue, in particular persuasion dialogues. One account for formal persuasion dialogue systems (Prakken, 2006) was also discussed. This thesis, however, aims at modelling arguments that can be formed on the basis of experience, rather than handcrafting arguments into a knowledge base. Nevertheless, elements from the discussed account (Prakken, 2000) and other formal systems will be included in the formal representation of the proposed model.

- Argumentation theory provides a number of mechanisms which are useful in their application to AI, in particular AI and Law. Argumentation has proved to be an invaluable tool to legal reasoning systems, a survey of these systems was given, in particular the well documented HYPO system. An adaption of the argumentation model of this latter system will be incorporated in the theory presented in the next chapter.
- Case Based Reasoning in AI and Law also provided the inspiration for the particular moves used in the proposed protocol.

Each of the above areas has influenced the proposals that will be presented in the following chapters in presenting the theory of "*Arguments from Experience*". However, in order to implement this theory, another field of research needs to be covered: association rule mining. This field will provide this thesis with elements to discover arguments, formed as association rules, from agent's experience, presented by collection of examples. The subsequent section will provide an overview of the subject of association rule mining.

## 2.2. Association Rule Mining and Knowledge Discovery

Having discussed the key ideas with regard to argumentation and its application to AI most relevant to the subject matter of this thesis, this chapter turns now to examine the details of one data mining technique of essential importance to this thesis: namely Association Rule Mining (ARM). ARM is a process whereby Association Rules (ARs), representing relationships between attributes in a collection of records (dataset), can be discovered. Chapter 3 will present a theory to enable agents to argue on the basis of their past experience, arguments will be constructed as ARs mined from each agent's experience. This section is intended to examine the field of ARM and to make observations regarding which of the many techniques reported in the literature is best suited for the purposes of realising the proposed model for "*Arguing from Experience*".

### 2.2.1. The process of Knowledge Discovery in Databases (KDD)

The subject of analysing large volumes of data to discover new interesting knowledge has been the focus of extensive research, the origins of which can be traced back as far as the first days of philosophy of science. Statistics was often regarded as the proper scientific discipline of data analysis. However, the revolution in computer science in the 1950s enabled new techniques, such as machine learning, pattern recognition, data mining, etc, as an alternative means to data analysis. One particular approach to computerise data analysis, *Knowledge Discovery in Databases* (KDD) (e.g. (Piatetsky-Shapiro, 2000)), has become a popular research area in the past decade or so. The concept of KDD was first introduced by Frawley *et al.* (1991), to quote:

> *"Knowledge discovery is the nontrivial extraction of implicit, previously unknown, and potentially useful information from data"*.

The large amount of data being stored in databases, covering a variety of domains from marketing and sales to bioinformatics and nanotechnologies, has provided a fertile background for research in KDD. Additionally the hypothesis that some hidden knowledge is likely to exist in the form of rules, patterns, or trends in a set of data, especially when the size of a data set becomes large, is most attractive. However, the process of KDD is not trivial, and it involves many stages in order to extract knowledge from large volumes of data. One proposed outline for KDD comprises seven stages and proceeds as follows (Ahmed, 2004):

- *Problem Specification*: This is the first stage in any KDD application and aims at creating a domain oriented specification of the target problem.
- *Resourcing*: Aims at collecting (or creating) a sufficient amount of data suitable for the target application.
- *Data Cleaning*: Aims at removing noise and inconsistencies from a given data set.

- *Data Integration*: Involves combining data residing at different sources thus providing the user with a unified view of these data.
- *Pre-processing*: Comprises two tasks, (i) data transformation and (ii) data reduction. The first transforms the collected data into a structured representation. The second selects the data most significant for the target application - any other data is discarded.
- *Information Mining*: This is the core task in the overall KDD process. The purpose of which is identifying the most valuable information in the prepared data by utilising data analysis and data mining techniques, and produces a particular enumeration of patterns over the data.
- *Interpretation and Evaluation of Results*: Evaluates the results of the information mining step, thus the overall quality of the mining performance could be assessed. The discovered knowledge is presented to the user in a readable format for them to interpret and assess.

The above list is intended to give a brief summary of the process of KDD. The stages are usually applied iteratively; with results of one stage providing output to the later and feedback to earlier stages. However, this thesis focuses mainly on the last two stages although with some references to the data cleansing stage. While the others are of great importance to the overall KDD process, they are outside the scope of the work presented in the rest of this thesis.

The existing literature of KDD research is rich in examples as to how the process of KDD is applied to mining information from a variety of sources such as: Tabular Data Mining (e.g. (Han and Kamber, 2006)), Text Mining (Feldman and Sanger, 2006) and Web Mining (Chang *et al.*, 2006). The work presented in this thesis makes use of one application of KDD only: *tabular data mining*, the process of mining information from tables where the data is stored in the form of a database-like format. *Tabular data mining* combines elements from different fields such as databases, machine learning, statistics, AI, etc.

### 2.2.2. Association Rules Mining (ARM)

The task of ARM is to find "*interesting*" correlations between attributes in a given database. These correlations are inferred empirically from examination of the records in the database. ARM was first introduced by Agrawal et al. (1993), and since then has received considerable attention, particularly after the publication of the Apriori algorithm (Agrawal and Srikant, 1994). The discovered correlations are referred to as Association Rules (ARs): an AR describes an implicative co-occurring relationship between two disjoint sets of database attributes A and B, expressed in the form of an antecedent (A) → consequent (B) rule. Initial research on ARM was largely motivated by the analysis of super-market basket data, the results of which allowed companies to understand, more precisely, purchasing behaviour and, as a result, to direct their advertising efforts towards the most promising market audiences. Consider the following example: a retailing company can implement a better-targeted marketing policy by discovering ARs representing knowledge about its customers' purchasing behaviour. For instance, suppose a strong correlation is found between two attributes, say bed linen and pillow cases; namely an AR of the form: bed linen → pillow cases, indicating that customers who bought bed linen also bought pillow cases in the same transaction. In this case the company can more efficiently target the marketing of pillow cases through advertising to those clients who have bought bed linen but not pillow cases. The company can offer, for instance, a discount on pillow cases when buying bed linen, or by situating the pillow cases on the same aisle as the bed linen. ARM techniques have also been applied to other areas such as risk analysis in commercial environments, epidemiology, clinical medicine and crime prevention; all areas in which the relationship between objects can potentially provide useful knowledge.

With this in mind, this thesis aims at applying ARs in a dialectical context, interpreting the AR as: "*Antecedents are reasons to believe that the consequent is true*", allowing arguments, in the form of ARS to be exchanged among a number of participants to come to a resolution regarding conflicts of opinion. Thus, ARs will be used to model "*Arguments from Experience*", advocated in this thesis. The reasons behind using ARs, rather than other types of inductive rules, are as follows:

- ARs provide an understandable means to represent argumentation "*rules*". Other techniques use domain-specific biases and calculations to produce a small set of rules. Such rules are of no use to the argumentation process suggested in this thesis.
- ARs have room for more than one attribute in their consequences, which allows for more complex arguments to be generated (mined), from the agent's experience.
- ARM paradigms enable the discovery of *interesting* rules. The interest measures associated with the discovered ARs provide means to assess/evaluate/prioritise "*Arguments from Experience*".
- The summarising data structures associated with some of the ARM paradigms enable the generation of ARs with a given set of attributes. Thus allow for an effective mining of the desired "*Arguments from Experience*" as will be made clear in Chapter 3.

In summation, ARs provide a tool to arguments generation that allows for sophisticated rules, such as ones with more than one attribute in their consequences, or ones with negative values, to be automatically mined from a given dataset in an understandable form. The main drawback is that ARs does not have room for rules such as A or B →X, which can play an interesting role in some argumentation frameworks. However, the work on argumentation in general, uses simple "*AND*" rules. Rather than the more complicated "*OR"/"XOR"* rules

Agrawal et al. (1993) present a formal statement of ARM whereby if $I = \{i_1, i_2,.., i_n\}$ is a set of items and $D$ is a set of database records, each record $R \in D$ is a subset of the items in $I$ such that $R \subseteq I$. Note that items here refer to binary (Boolean) attributes. An AR is identified as an implication of the form $A \rightarrow B$, where $A \subseteq I$, $B \subseteq I$, and $A \cap B = \emptyset$. In general, a set of items (the antecedent or the consequent of a rule) is called an "*itemset*", the *length* of which equates to the number of items in the itemset, so that itemsets of length k are referred to as *k-itemsets*.

Clearly, the total number of ARs contained in D can be very large, especially considering the increasing sizes of modern day datasets. To limit the number of identified ARs, ARM aims to extract only potentially *useful* (or *interesting*) ARs from which new knowledge can be derived. The criteria of *usefulness* established in the literature states that ARs should be *novel*, *externally significant*, *unexpected*, *nontrivial*, and *actionable* (e.g. (Hilderman and Hamilton, 1999) and (Roddick and Rice, 2001)). The role of ARM systems in this elicitation process is: to facilitate the discovery of ARs, and filter these rules on the basis of heuristics, and to enable the presentation of resulting ARs for subsequent interpretation by the user to determine their worth.

Since the publication of (Agrawal et al., 1993) the analysis of the process of ARM has become a mature field of research. The fundamentals of ARM are now well established. The majority of current research involves the specialisation of fundamental ARM algorithms to address specific issues, such as the development of incremental algorithms to facilitate dynamic data mining or the inclusion of additional semantics (e.g. time or space) to discover. However, this review is concerned with providing a clear understanding as to how the process of ARM functions, along with a summary of some influential ARM algorithms, relative to the work undertaken by this thesis. Additionally, one particular approach to ARM: Dynamic ARM is also discussed. Other more advanced ARM research, while interesting, is outside the scope of this review.

#### 2.2.2.1. The Process of ARM

ARM is a two part process: firstly, *all Frequent Itemsets* (FIs) in a given dataset are identified. Secondly, ARs are identified from the FIs according to some measure of interest. An FI is some subset of I (as defined above) that occurs more than, or equal to, a given threshold (e.g. support). Due to the increasing sizes of datasets, the first task is the most time consuming, whereas the second is a straightforward inference process. Basically, once all the frequent itemsets (FIs) are generated, one can easily generate an AR from a given frequent itemset

$F \in$ FIs, by first identifying all the subsets from F that are in FIs (i.e. all the subsets that are themselves frequent itemsets). Then for every such subset *A* of *FI*, the association rule A→(FI-A) is generated if the interesting measure of such a rule is greater than the interesting threshold determined by the user (for instance if the confidence of the rule is larger than the confidence threshold given by the user). The majority of ARM related research to date has focused upon the efficient discovery of FIs. Given the set of items *I* there are $2^{|I|-1}$ possible combinations of items to explore and given that $|I|$ is often large, "*brute force*" exploration techniques are often intractable. Relevant research can be organised into four categories:

- Constraining the number of "*interesting*" ARs through the incorporation of *Measures of Interest* (MOI) in the exploration process, and the application of efficient pruning strategies according to these MOIs.
- Reducing the number of passes over the database required to mine all the interesting ARs by reducing the number of I/O operations.
- The implementation of efficient and useful data structures to represent the databases. This research has resulted in the evolution of tree based data structures to efficiently represent the exploration space.
- Producing a condensed set of FI allowing the entire result set of ARs to be explored and inferred from this reduced set, thus achieving more efficient management of the storage space, and facilitating user interpretation.

One of the most applied MOIs is the *Support and Confidence* criteria (Agrawal et al., 1993). Given an AR $R: A \rightarrow B$:

- The *support* (*s*) is the percentage of records that contains $A \cup B$. Support measures the frequency with which an itemset ($A \cup B$) occurs in the database, and this itemset is said to be frequent (large) if it has a support higher than the specified threshold. Consider a dataset of recent purchase transactions in an electronics stores, then if the support of a certain type of laptops is 1% then that would mean that 1% of the transaction records include a purchase of this particular laptop.

- The c*onfidence* (c) is the percentage of the number of transactions that contain ($A \cup B$) to the total number of records that contain $A$ (but not $B$), thus the conditional probability of $B$ given $A$. Confidence is a measure of the strength of ARs. In the electronics stores transactional dataset, if laptop → laptop bag holds with 80% confidence, this means that 80% of people who bought a laptop from this store also bought a laptop bag in the same time.

Over the past two decades a variety of ARM algorithms have been developed using a variety of techniques including, but not limited to, the refinement of search strategies, pruning techniques, data structures, and the use of alternative dataset organizations. The algorithms most relevant to the work undertaken in this thesis are summarised in the following.

#### *2.2.2.2. A survey of ARM algorithms*

The most computationally demanding part of the process of ARM is the task of identifying the FIs. The number of possible itemsets in any given dataset is exponential in the number of items. Generally speaking most existing ARM approaches attempt to identify *candidate* itemsets before validating them with respect to the implemented MOI, where the generation of candidates is based upon previously identified FIs. These ARM methods are referred to as "*Candidate Generation*" techniques. The performance of these methods depends both on the size of the original data and on the number of candidates being considered. The number of possible candidates grows with the increasing number of items present in data records and with decreasing support thresholds. In order to achieve better performances these algorithms generally exploit some type of tree-based data structures to represent the discovered itemsets (e.g. hash, set-enumeration or prefix trees).

The most widely quoted ARM algorithm is the Apriori algorithm (Agrawal and Srikant, 1994), which acted as a catalyst for the development of ARM algorithms. The main idea behind Apriori is the "*downward closure property*" whereby if any given itemset is not supported then any superset of this set will also not be supported. Hence any effort to calculate the support of such supersets is redundant. The Apriori algorithm involves multiple scans of the given database. The first pass counts the occurrences of single items in the database to determine the frequent 1-itemsets. In each of the subsequent passes a new set of candidate itemsets is generated using the FIs found in the previous pass, and then the database is scanned to count the actual support count of the identified candidates. At each pass, the discovered candidate itemsets are stored in a *hash tree*. (Hash trees are essentially b-trees for which every internal node is a hash table, and every leaf node contains a set of itemsets). Since its introduction Apriori has proved to be very influential to the field of ARM. However, Apriori suffers three inherent drawbacks: (i) many candidate sets, which might be proven infrequent, are still generated; (ii) it requires repeated scans of the database which might be a problem with respect to large candidate sets; and (iii) the hash tree data structure is not particularly efficient.

However, the advent of Apriori and the downward closure property has provided a standard pruning technique, mainly through the use of the support heuristic. Subsequent research in ARM has focused on reducing I/O through condensed representations, dataset partitioning, dataset pruning, and dataset access reduction. The result of this research has been a large body of Apriori-like algorithms following the style of operation adopted by the Apriori algorithm, while achieving better performance by reducing the number of the I/O operations. For instance, the *Partition* algorithm (Savasere et al., 1995) adopts the heuristic that in order for an itemset to be frequent in the whole database it must be locally frequent in at least one partition of the database. The *Partition* algorithm works well with datasets where the count of an itemset is evenly distributed in each partition. But with an irregular data distribution a considerable amount of CPU time is wasted counting false itemsets, alternatively itemsets may be missed. Similar thinking motivated another set of algorithms the intuition behind which was that approximate answers often suffice and therefore adequate answers can be obtained by mining a compressed representation of dataset *D*. The main issue in developing sampling techniques is to maximise the extent to which the sample reflects the generic characteristics of *D* while maintaining efficiency through sample size constraint. One notable sampling algorithm is that of Toivonen (Toivonen, 1996) which only requires a single scan of D to discover all FIs. However, both sampling and partitioning based approaches share the same weakness with Apriori: the number of candidate sets to be generated grows exponentially. Also these methods assume a normal distribution across the dataset in order to support sampling or partitioning.

Another set of ARM algorithms have focused on using the support/confidence MOI to discard any generated candidate k-itemsets that fall below the minimum support/confidence thresholds. One notable example in this genre of algorithms is the *Apriori-TFP* algorithm (e.g. (Coenen et al., 2004a)) which is described at length in the following sub-section. Other examples include: DIC (Brin et al., 1997), Eclat and Clique (Zaki, 2000).

In contrast to the more prolific candidate generation techniques, pattern growth algorithms eliminate the need for candidate generation through the creation of complex data storage structures referred to as *hyper-structures*. In general, a hyper structure comprises two linked structures, a *pattern frame* and an *item list*, which together provide a concise representation of the relevant information contained within the data set. The first stage of analysis populates the hyper structure and, so long as the representation can be maintained in memory, further dataset access is not required. Subsequent mining involves depth first analysis of the pattern frame, accessed through the item list. However the nature of the hyper structure is algorithm dependant, varying in relation to the substructures and the underlying semantics. The best known pattern growth algorithm, FP-Growth (Han et al., 2000), uses a tree-based pattern frame (*FP-Tree*) and an associated header table (*FP-Link*) within the analysis process. FP-Growth is a recursive procedure during which many sub FP-trees and header tables are constructed, it begins by examining each item in the FP-tree header table, starting with the least frequent. For each entry the support value for the item is counted by following the links connecting all occurrences of the current item in the FP-tree. The advantages of the FP-growth algorithm, and other similar algorithms (e.g. FP-growth* (Grahne and Zhu, 2003)), are partly dependent on the ordering process, which reduces the overall size of the input dataset, as the unsupported items are eliminated during this order process; and also reduces processing time by allowing the most common items to be processed most efficiently. This concludes this survey of some of the most popular ARM methods. While other interesting approach exists such as Condensed Representation ARM (e.g. (Pei et al., 2000)) and Maximal FI ARM (e.g. (Burdick *et al.*, 2001)), these approaches fall outside the scope of the work presented in the forthcoming chapters. In the following one particular approach is examined in greater details.

### *2.2.2.3. Apriori-TFP and the related data structures*

The previous survey provided an overview of a variety of ARM algorithms. This section returns to one particular approach: Apriori-TFP. This technique, as will be made clear in later chapters, has particular significance with respect to the work described in this thesis, as it forms the basis for the discovery of the ARs used in forming "*Arguments from Experience*". The intuition behind Apriori-TFP (Total From Partial) (Coenen et al. 2004a, b) is to compute support counts, for candidate itemsets, starting from an initial incomplete computation stored as a set enumeration tree, referred to as P-tree (Partial-support tree), instead of operating with the raw input data. *Set-enumeration trees* are ordered trees (usually lexicographic) where each node represents an itemset, and every edge represents a single item extension of that itemset. Apriori-TFP delivers its results in an efficient manner due to the pre-processing advantages offered by the P-tree structure. Once the P-tree has been created the TFP algorithm determines ARs by creating another data structure from the P-tree; this second data structure is referred to as T-tree (Total-support tree), from which the final ARs are produced. Apriori-TFP algorithms have been further applied in a number of different ARM directions such as Distributed and Parallel ARM (e.g. (Coenen and Leng, 2006)), mining very large DBs that cannot be held in primary storage (e.g. (Ahmed, 2004)) and classification (e.g. (Coenen et al. 2005)). In the following, Apriori-TFP is discussed in detail. An extensive account of the data structures associated with this approach is also given.

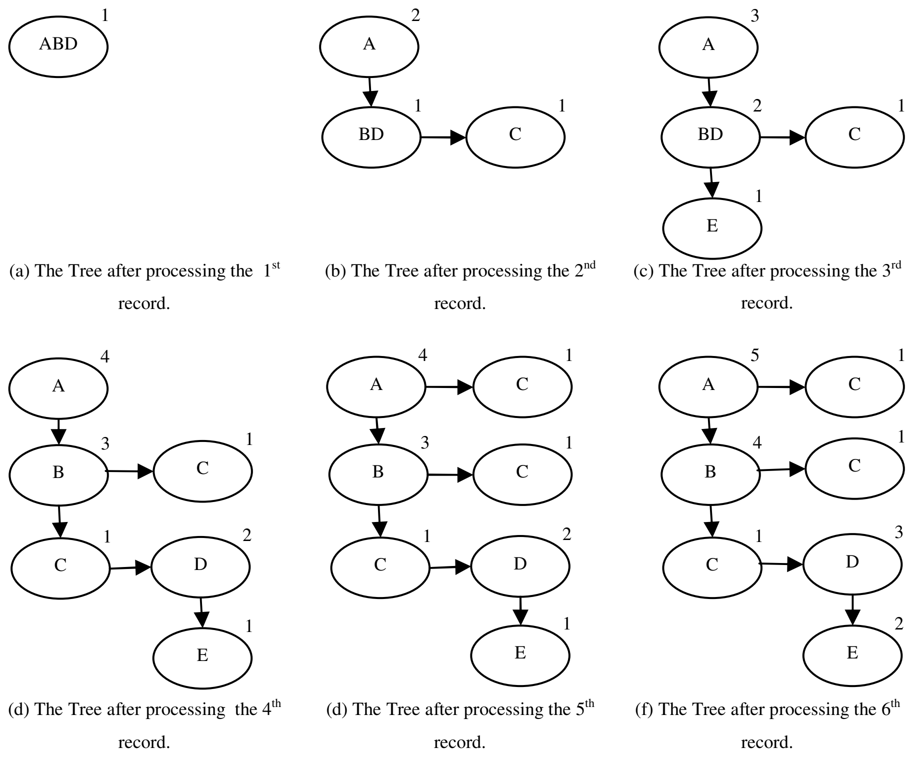
The P-tree summarises the input data into a "*compressed*" form, with the inclusion of partial support counts. Coenen et al. (2004a) define these partial counts as incomplete support totals. The P-tree consists of all the itemsets present as distinct records in the database, plus some additional sets that are leading subsets of these. The P-trees construction algorithm is presented in Figure 2.2 using pseudo code.

```
Input: Dataset (D).

int n =1;
PT = an empty P-tree structure;
While (n≤ |D|)
 let $D_n$ be the nth record in D.
 traverse PT with $D_n$;
 update the support count for each node in the traverse path
 as required
 **if** $D_n$ or any trailing subsets are missing from PT **then**
  add new node for each missing subset
 n= n + 1;
**return** (PT);
```

**Figure 2.2. P-tree Generation Algorithm.**

Figure 2.3 shows the steps by which a P-tree is generated from a dataset DB={{A,B,D}, {A,C}, {A,B,D,E}, {A,B,C}, {C}, {A,B,D}}. The P-tree generation process begins by scanning the first records in DB ({A,B,D}); a new P-tree node (ABD) is created to represent this record and it is given the support count of 1 (Figure 2.3 (a)). After which the algorithm proceeds to process the second record in DB, ({A, C}). Again a new node is added to the P-tree representing this record, but since the two nodes on the tree share a common prefix (A) a "*dummy*" node is created to represent this "*leading substring*", and both nodes are assigned as children of the new node; the support count of node (A) is calculated as the sum of the supports of its children (Figure 2.3 (b)). The P-tree generation process continues in the same manner for each record in DB, Figures 2.3 (c), (d), (e), (f) illustrate the progression of the P-tree after processing each of the remaining records in DB.



(a) The Tree after processing the 1st record.

(b) The Tree after processing the 2nd record.

(c) The Tree after processing the 3rd record.

(d) The Tree after processing the 4th record.

(d) The Tree after processing the 5th record.

(f) The Tree after processing the 6th record.

**Figure 2.3. Example of the P-tree generation process.** *Horizontal arrows represent "sibling" relationship" between the nodes. Vertical arrows represent "parent-child" relationship.*

Ahmed et al. (2004) provide distinctions between the P-tree data structure compared with other trees structures, in particular, the FP-Tree (Han et al., 2000). Ahmed et al. note that both the P-tree and the FP-tree structures share many similarities, but they differ on two main points, both lead to a more compact tree structure:

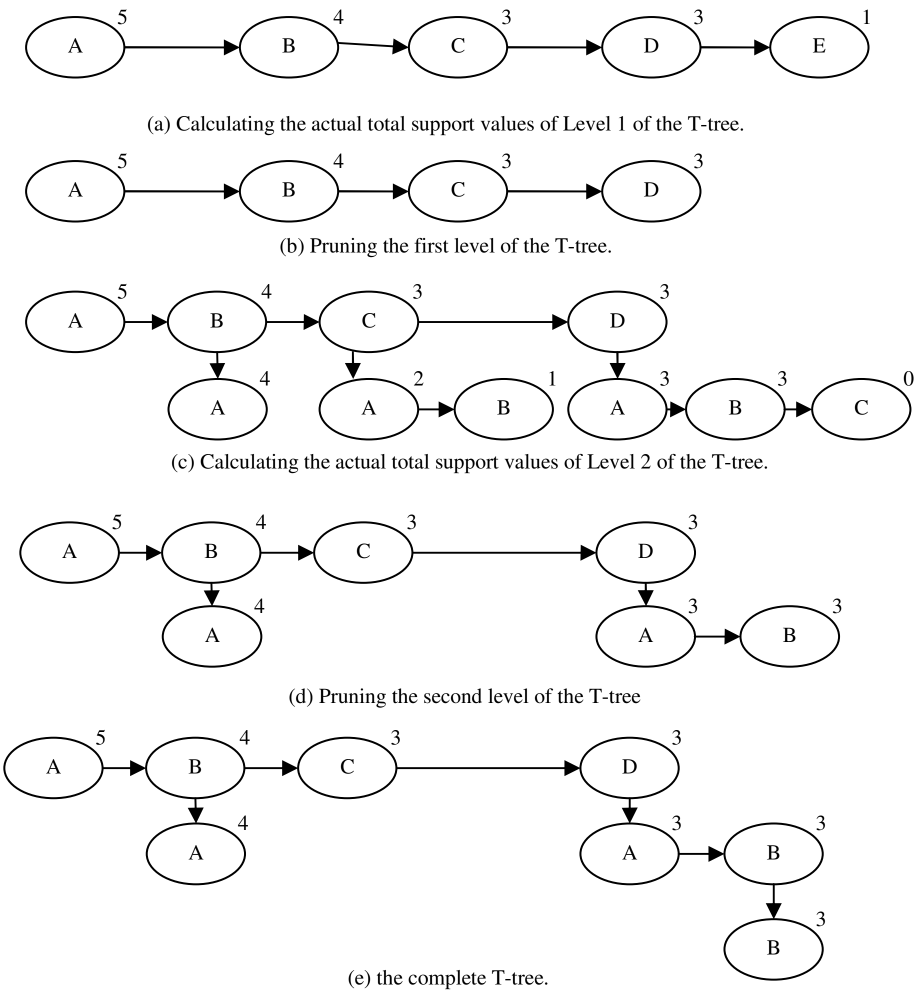
- The nodes of the P-tree correspond to a sequence of items which is partially closed (has no leading subsequence with greater support in the tree), whereas FP-tree is composed of nodes expressing individual data items.
- The implementation of the FP-Growth algorithm requires storing pointers at each node to link all nodes representing the same item in the FP-tree, whereas Apriori-TFP treats the P-tree essentially as a set of nodes which can be processed in any order (P-tree is simpler). This makes it possible, once the P-tree has been constructed, to store it in a tabular form in which no pointers are required.

Once the P-tree representation of the dataset is built, the Apriori-TFP algorithm constructs a second tree-structure, referred to as the T-tree (Total-support Tree). This structure is a reversed set enumeration tree representing the total support counts of the FIs (Coenen et al., 2001). The process of constructing the entire T-tree could imply an exponential storage requirement. However, Coenen et al (2001) argue that in practice it is only necessary to create that subset of the tree corresponding to the current candidate set being considered. Thus the concept of Apriori could be applied to build a T-tree based on a P-tree using the Apriori-TFP algorithm (Coenen et al., 2001) (Coenen et al., 2004b). Apriori-TFP completes the computation of the final support counts, storing the results in a *T*-tree, ordered in the opposite way to the *P*-tree (reversed lexicographic order). The final *T*-tree contains all frequent sets with their complete support-counts. An example demonstrating how the T-tree is constructed using the P-tree given in Figure 2.3 is given in Figure 2.4. The example assumes a support threshold of s=3 (50% of the records).The TFP algorithm generates the T-tree level by level in an Apriori manner, commencing by listing the candidate 1-item nodes with their total support counts initialised to 0. Next, the P-tree is traversed to add the interim support counts of the corresponding P-tree nodes to each candidate 1-item node in the T-tree (Figure 2.4(a)). After this initial calculation, any unsupported 1-item nodes are pruned from the tree (Figure 2.4(b)), this completes the construction of the first level of the T-tree. The subsequent levels are constructed by first generating the nodes representing the candidate itemsets at the level in question (K), the total support counts of these nodes are initialised to 0. The P-tree is again traversed to compute the total support for each candidate K-itemsets in the T-tree (Figure 2.4(c), (e)) and again the FIs are pruned from the tree (Figure 2.4(d)).

The T-tree data structure, discussed above, offers a number of, mainly:

- Reduced storage requirements compared to those required by more traditional tree structures.
- Fast look up facilities (by indexing from level to level), and finally,

- The structure offers computational advantages because FIs with particular consequences are stored in a single branch of the tree.



(a) Calculating the actual total support values of Level 1 of the T-tree.

(b) Pruning the first level of the T-tree.

(c) Calculating the actual total support values of Level 2 of the T-tree.

(d) Pruning the second level of the T-tree

(e) the complete T-tree.

**Figure 2.4. Example of the T-tree generation process.** *Horizontal arrows represent "sibling" relationship" between the nodes. Vertical arrows represent "parent-child" relationship.*

The above merits have motivated the usage of the P- and T-tree data structures in the work described in the following chapters, where it will be argued that a particular attribute is present. To sum up, the P- and T-trees provide an interesting and appealing approach to ARM because:

- The T-tree offers significant advantages in terms of generation time and storage requirements compared to hash tree structures.

- The P-tree offers significant pre-processing advantages in terms of generation time and storage requirements compared to the FP-tree.
- The T-tree is a very versatile structure that can be used in conjunction with many established ARM methods.

The above list of advantages, in particular when the P- and T-trees structures are used for mining ARs comprising of certain itemsets (as will be discussed in Chapter 3), have provided sufficient motivation to apply an Apriori-TFP like approach to mine ARs to provide "*Arguments from Experience"*.

### *2.2.2.4. Dynamic (On-Line) ARM*

The original objective of Dynamic ARM (D-ARM), also sometimes referred to as On-line ARM, was to address the increasing computational requirements for exploratory ARM (usually involving manual parameter tuning). D-ARM has also been used in applications where repeated ARM invocations are required to obtain different sets of rules either with different content or different thresholds. The fundamental idea is to summarise the dataset so that all information required for future ARM is encoded in an appropriate data structure that will facilitate fast interaction. D-ARM was, arguably, first proposed by Amir et al. (1997) who used a tree data structure to store the datasets and conducted experiments using the (sparse) Reuters benchmark document set. Although Amir et al. enabled questions such as "*find all the ARs with a given support and confidence threshold*" to be answered, their system could not answer questions such as "*find the association rules that contain a given item set*". The approach by Amir et al. is essentially not dissimilar to later approaches to ARM, such as TFP (Coenen et al., 2004) and FP-growth (Han et al., 2000), that used an intermediate (summarising) data structure within the overall ARM process, although these later approaches did not explicitly consider the advantages with respect to D-ARM that their data structures offered.

The term On-line ARM was introduced by Aggarwal and Yu in 1998 in a technical report. In a subsequent publication, Aggarwal and Yu (1998), the authors state that "*The idea of on-line mining is that an end user ought to be able to query the database for association rules at differing values of support* *and confidence without excessive I/O or commutation*". Aggarwal and Yu define an adjacency lattice, where two nodes are adjacent if one is a superset of the other, and use this structure for fast (on-line) rule generation. The lattice contains only itemsets whose support is greater than some minimum and consequently only ARs with support above this value can be generated. Hidber (1999) has presented another lattice-based algorithm, CARMA (Continuous ARM Algorithm); but here, the user can influence its growth by reducing the support threshold as the algorithm proceeds. Chapter 3 will return to this notion of on-line ARM, where a number of ARM algorithms are discussed, each providing the means to mine ARs to support different types of queries, and each query relates to one move in the proposed model for "*Arguing from Experience*". These queries will enable the agents engaged in a dialogue over some case to look to their "*experience*" for ARs composed of a determined set of items, or to uncover rules with varying confidence values.

### 2.2.3. Summary of ARM and KDD

This section has provided a discussion with regard to the notion ARM within the context of the field of KDD. The process of ARM was explained and one particular approach to ARM, namely Apriori-TFP was discussed in detail because of its relevance to this thesis. The key points discussed in this section are summarised as follows:

- ARs represent "*interesting*" correlations between data items; thus they can be used to present inferences from experience, represented by a collection of records. Therefore, ARs are considered suitable for presenting "*Arguments from Experience*" as will be discussed in the impending chapters.
- Research in the field of ARM comprises a substantial body of algorithms and techniques, some of which were discussed, or referred to, in the above sub-sections. However, for the purposes of mining ARs to represent "*Arguments from Experience*" the candidate algorithms should:
  - Provide adequate means to represent the FIs thus enabling fast mining

  - Aid online/dynamic ARM, by which associations between a defined set of items can be mined, with respect to varying confidence/support values.

Consequently, the following chapters of this thesis will make use of the P- and T-trees data structures associated with Apriori-TFP to represent the underlining datasets in which the experience of agents is gathered.

## 2.3. Classification in KDD

The promoted model for "*Arguing from Experience*" is directed at providing a means for exploiting the experience gathered by a number of agents to come to a decision regarding a given case. As will be seen, this decision is attained via a dialectical process involving the arguing parties. This process is akin to determining a class label for a case: the advocated model can be used to assign class labels to data instances. Therefore, to evaluate the process of "*Arguing from Experience*" espoused by this thesis a number of classification problems will be used and by comparing the results obtained from the advocated model to those obtained from a selection of established classification technique, the following can be evaluated:

- The operation of the promoted model and the various features of the resulting dialogues.
- The quality of the resulting classifications. By selecting a number of well-known classification techniques, an assessment can be made as to whether the process of "*Arguing from Experience*" can deliver results competitive with the determined classifiers in all (or some) domains.

Chapters 5 and 8 will present the results of collections of such comparative evaluations and provide discussions of the main findings. Given the above this section provides an overview of the process of *Classification (Categorisation*) in the context of tabular data mining.

Classification algorithms (or approaches), promoted in data mining research, are directed at building classifiers that can be used to assign class labels to "*unseen*" data instances. Formally, the problem of *Classification* in tabular databases is described as follows. Given a collection of records $D_C$ which consists of *N* cases ($|D_C| = n$) described by *I* distinct attributes. Assuming these *N* cases have been classified into $|C|$ known classes where C comprises a set of pre-defined class labels ($C = \{c_1, c_2, \ldots, c_{|C|-1}, c_{|C|}\}$), a classification approach can then be applied to produce a classifier, based upon $D_C$, to assign a class label $c \in C$ to any "*unseen*" ("*future*") records. The process of generating a classifier consists of two phases: (i) a training phase where a classifier is built from a set of training data instances $D_R \subseteq D_C$; and (ii) a test phase where "*unseen*" instances in a test data set $D_E \subseteq D_C$ are classified, using the generated classifier, into the pre-defined classes so as to provide a measure of the accuracy of the generated classifier. $D_C$ is established as $D_R \cup D_E$, where $D_R \cap D_E = \varnothing$. A substantial number of techniques have been developed and adapted to generate classifiers, including: Neural Networks, Support Vector Machine, Decision Trees, Association Rules and various mechanisms influenced by ideas take from genetic programming and bio-computation. The work presented in the forthcoming chapters makes use of nine classification algorithms that were thought to be most related to the subject matter of this thesis. The following Sub-section provides a brief discussion of the nine classification algorithms, with the reason for their inclusion.

### 2.3.1. Summary of the Classification Algorithms used

For the purposes of evaluating the process of "*Arguing from Experience*" a total of nine classification algorithms were applied and/or used, as follows:

- C4.5 (Quinlan, 1993). CN2 (Clark and Niblett, 1989).
- ABCN2 – Argument Based CN2 (e.g. (Mozina et al, 2005)).
- CBA – Classification Based Associations (Liu et al., 1998).
- CMAR – Classification based on Multiple ARs (Li et al, 2001).
- TFPC – Total From Partial Classification (e.g. (Coenen et al, 2005)).
- FOIL – First Order Inductive Learner (Quinlan and Cameron-Jones, 1993), as applied for Classification ARM (CARM).

- CPAR – Classification based on Predictive ARs (Yin and Han, 2003).
- PRM – Predictive Rule Mining (Yin and Han, 2003).

Note that the software implementation for C4.5, TFPC, CMAR, CBA, FOIL, CPAR and PRM was obtained from the LUCS-KDD research team in the Department of Computer Science, at the University of Liverpool. This software is publicly available for download from the following web page: **http://www.csc.liv.ac.uk/~frans/KDD/Software/**. CN2 and ABCN2 were not implemented. Rather the results provided in (Mozina et al., 2005) were used in Chapter 5, to provide comparison with PADUA. A concise overview of each of these algorithms is given below. A summary table is also provided at the end of this sub-section.

C4.5 is probably the most referenced classification algorithm in both KDD and machine learning. C4.5 was introduced by Quinlan (1993) and since then it has provided a default technique against which other approaches are evaluated (e.g. (Liu et al, 1998), (Li et al, 2001) and (Yin and Han, 2003)). For these reasons, the work presented in this thesis makes use of C4.5 for the purposes of evaluating the operation of "*Arguing from Experience*". In brief, C4.5 is a decision tree algorithm: it processes a training set $D_R$ and creates a tree data structure that can be used to classify unseen instances. The leaf nodes of the constructed decision tree represent the class labels, while internal nodes represent attribute-based tests with a branch for each possible outcome. In order to classify a new data instance, C4.5 commences at the root of the constructed tree, evaluates the test, and take the branch appropriate to the outcome. The process continues until a leaf node is encountered, and then the instance is assigned the class label this leaf represents. The main distinction between different decision tree algorithms is the criteria for identifying the attributes on which to "*split*" at each node. C4.5 applies the gain ratio measure and operates by recursively splitting the dataset on the attribute with the maximum gain ratio to generate the decision tree. This tree is then pruned according to an error estimate and the result is used to classify new data instances.

Another well established approach to classification is that of rule induction. Rule induction algorithms typically operate using the *cover principle* whereby rules are iteratively inferred, and once a rule is established all training data records associated with it are discarded. The process continues until the training set is empty or no more acceptable rules can be inferred. One such algorithm is CN2 (e.g. (Clark and Niblett, 1989). CN2 is a well established algorithm that was consequently also selected for comparison purposes in this thesis. CN2 uses a covering algorithm and a search procedure that finds individual rules by performing a beam search. *ABCN2* - Argument Based CN2 - (Mozina et al., 2005) is an extension of CN2 which augmented the original CN2 to take into account arguments that explain misclassified examples. Another pass uses these arguments to constrain the rules generated. Chapter 8 will return to this algorithm and provide further details of how it functions. Also, Chapter 5 will make use of some of the results given in (Mozina et al., 2005) to compare the application of "*Arguing from Experience*" with both CN2 and ABCN2.

Another set of well known rule induction algorithms is the FOIL family of algorithms (Quinlan and Cameron-Jones, 1993). The FOIL - First Order Inductive Learner - algorithm heuristically builds rules, for each class label, from items (attributes) in the training dataset using the FOIL-gain method. On each iteration, FOIL seeks the item that yields the largest FOIL-gain for a particular class in the training set. Once the rule is identified, all training records associated with it are discarded (as per the cover principle) and the process is repeated until positive data records for the current class are covered. An implementation of the FOIL algorithm for generating CARs will be used in this thesis[8] for evaluation purposes. In addition two extensions of FOIL will also be used: (i) CPAR, Classification based on Predictive Association Rules, (ii) and PRM, Predictive Rule Mining; both proposed by Yin and Han (2003).

In PRM, the weight of the rule is decreased by a factor if it correctly classifies an example. By using this weighting strategy instead of removing the rules, PRM generates more rules and a positive example might be covered several times. CPAR combines the advantages of both FOIL and PRM to generate a

[8] The details of this application can be found in (Coenen, 2004c).

smaller set of high quality predictive rules by considering the set of previously generated rules to avoid redundancy. CPAR and PRM differ from FOIL in that not all the records associated with one item are removed once it is determined. Instead, weights of records associated with that item are reduced by a multiplying factor and the process is repeated until all positive data objects for the current class are covered. This weighted application extracts more rules, as it is possible for each record from the training set to be covered by more than one rule. Nevertheless, the rule set produced is still relatively small compared with CARM techniques (see below).

The rest of the classification algorithms that feature in this thesis, and that are reviewed in this sub-section, are all Classification Association Rule Mining (CARM) algorithms. CARM utilises ARM techniques to identify the desired classification rules. Originally, the application of ARs for classifying data was motivated to tackle situations where traditional classification techniques would be ineffective (see (Ali et al., 1997)) such as when the data records comprise a large number of attributes. CARM is an integration of ARM and classification. This integration is achieved by modifying existing ARM algorithms to focus on the subset of ARs whose right-hand-side is restricted to the set of class attribute, here after referred to as Classification Association Rules (CARs). A disadvantage of CARM is that classification datasets often contain many continuous (or numeric) attributes; thus data must be discretised before CARM can be applied. As will be seen in later chapters the proposed "*Arguing from Experience*" paradigm makes extensive use of ARM technology in a similar way to CARM. Comparison between the proposed approach and three CARM algorithms (CBA, CMAR, TFPC), is therefore undertaken. In the remainder of this sub-section these algorithms will be briefly reviewed.

The CBA - Classification Based Associations - algorithm described in (Liu et al., 1998) employs the Apriori candidate generation step to mine CARs. CBA involves three steps as follows. First, the Apriori algorithm (Agrawal and Srikant, 1994) is applied to generate frequent *ruleitem* sets, where a *ruleitem* is an itemset associated with a class label, and thus defines a potential CAR. The second step involves pruning the sets generated during the previous stage using the calculated confidence to eliminate those that fail to meet the required confidence threshold or which conflict with higher-confidence rules. Finally, a classifier is built by selecting an ordered subset of the remaining CARs.

To avoid the efficiency drawbacks of Apriori a number of CARM algorithms were developed on the basis of pattern growth approaches. One notable example, which will be used later in this thesis, is the CMAR - Classification based on Multiple Association Rules - algorithm (Li et al, 2001). CMAR stores rules in a prefix tree data structure, known as a CR-tree, in descending order according to the frequency of attribute values appearing in their antecedent. The algorithm inserts the CARs produced at each level to the CR-tree with respect to a path from the root node. The utilisation of the CR-tree considers the common attribute values contained in the rules. CMAR thus uses less storage than CBA.

The TFPC - Total From Partial Classification - algorithm ((Coenen et al., 2005), and (Coenen and Leng, 2005)) is founded on the TFP algorithm described previously. TFPC employs the same structures as in TFP to identify CARs in a given set. For this purpose, the class labels in the training set are held at the end of the item list so that all frequent sets associated with a single class are held in the same branch of the T-tree. TFPC was motivated by the desire to avoid overfitting. In essence, overfitting occurs when the induced model (the rule set) reflects the idiosyncrasies of the particular data being mined that are not reliable generalisations for the purpose of predictions involving new data. The intuition behind TFPC was that: for a given confidence threshold the algorithm would record rules that satisfy this threshold without exploring further to determine if more specific rules, with higher confidence, exist. TFPC is seen to be important in the context of this thesis because it utilises similar data structures to the TFP algorithm. This concludes this overview of the classification algorithms used in this thesis. Table 2.2 provides a summary of the algorithms.

### 2.3.2. Ensemble Methods

Another approach to classification, which is considered relevant to the work presented in later chapters, is that of ensemble methods which aim at improving the predictive performance of classification by combining a number of classifiers into one model (hereafter referred to as an "*ensemble*"). The advocated model for "*Arguing from Experience*" will be seen as an ensemble of classifiers, in which each participant in the argumentation process is considered as a classifier. And the overall argumentation dialogue is considered as means to select (by arguing about) the best class that meets the case under discussion. Chapter 8 will return to this point and provide more in-depth discussion.

| Method | Type | Base Technique | Use in Thesis |
|---|---|---|---|
| **C4.5** | CRM | Decision trees | PADUA and PISA- normal and noisy settings. |
| **CN2** | CRM | Rules Induction (RI) | PADUA analysis for systematic noise. |
| **ABCN2** | CRM | Argumented RI | PADUA analysis for systematic noise. |
| **FOIL** | CRM | RI | PADUA and PISA- normal and noisy settings. |
| **CBA** | CARM | Apriori | PADUA and PISA- normal and noisy settings. |
| **CMAR** | CARM | FP-growth | PADUA and PISA- normal and noisy settings. |
| **CPAR** | CARM | FOIL | PADUA and PISA- normal and noisy settings. |
| **PRM** | CARM | FOIL | PADUA and PISA- normal and noisy settings. |
| **TFPC** | CARM | Apriori-TFP | PADUA and PISA- normal and noisy settings. |

**Table 2.2. The classification algorithms used in this thesis.**

Both theoretical and empirical research (e.g. (Optiz and Maclin, 1999)) have demonstrated that a good ensemble is one comprising individual classifiers that are relatively accurate but make their errors on different parts of the input training set. This is because even though a given classifier may outperform all others for a specific subset of the input data, it is highly unlikely to find a single classifier achieving the best results on every instance in the problem domain. Consequently a good ensemble will attempt to exploit the differences in the behaviour of the base classifiers to enhance the accuracy and the reliability of the overall inductive learning system. In general, the output of several classifiers is useful only if there is disagreement among them. Ensemble methods therefore rely upon producing classifiers that disagree on their predictions; generally, this is achieved by altering the training process in the hope that the resulting classifiers will produce different predictions.

Two popular methods for creating accurate ensembles are Bagging (Breiman, 1996) and Boosting (e.g. (Freund and Schapire, 1996) and (Schapire, 1990)). Both techniques rely on varying the data to obtain different training sets for each of the classifiers in the ensemble. Methods of varying the data include: sampling, the use of different data sources, the use of different pre-processing methods, and adaptive re-sampling. Nevertheless, both techniques depart on two major points. Firstly, boosting changes the distribution of the training set in an adaptive way based on the performance of previously created classifiers, while bagging alternates the distribution of the training set in a stochastic manner. Secondly, boosting assigns weights (votes) to the results produced by each classifier according to some function of the performance (accuracy) of this classifier. Bagging uses equal weight voting. Boosting algorithms are generally considered to be more accurate than bagging for noise free data. However, bagging is more robust than boosting in noisy settings.

Bagging (Breiman, 1996) is a "*bootstrap*" (Efron and Tibshirani, 1993) ensemble technique aiming at creating individuals for its ensemble by training each classifier on a random redistribution of the training set. The training set for each classifier is composed by randomly drawing, with replacement, $|D_R|$ records. Thus many of the original records may be repeated in the resulting training set while others may be left out. Each individual classifier in the ensemble is associated with a different random sampling of the training set. Once a "*bagged*" ensemble is created, it classifies an unseen instance by having each of its base models classify the instance; then returning the class label that has received the maximum number of votes. This technique aims at generating classifiers from the different bootstrapped training sets in the hope that these classifiers disagree often enough to enable the ensemble to perform better than its individual classifiers. A bagged ensemble therefore relies on the instability of its base classifiers towards changes in the training data as a prerequisite for its effectiveness. If the ensemble individuals agree in all circumstances, then they will produce identical results, and thus the ensemble will not be any better than any of its members. Also, if there is too little data available, then the gain by the ensemble will not compensate for the decrease in the accuracy of its members, each of which now predicting classifications from a very small training set. Chapter 8 will return to these observations when evaluation the operation of "*Arguing from Experience*" when applied to multi-class classification problems. This operation will be viewed as an ensemble-like technique. However, the application of the proposed model to classification problems differs from bagging in that it makes use of an argumentation process in addition to applying the same classifier on different sets of the data. This argumentation step will further enhance the quality of the final output as will be made clear in Chapter 8.

Another approach that exploits different subsets of the training set to deliver better classification accuracy is boosting (Freund and Schapire, 1996, and Schapire, 1990). Boosting comprises a combination of methods, the focus of which is to produce a series of classifiers. Each classifier in the series is assigned a training set chosen on the basis of the performance of earlier classifiers. Boosting approaches feed the most recent classifier, in the series under construction, instances that have been misclassified by former classifiers more often than the ones that were correctly classified; thus attempting to produce new classifiers that are likely to predict the right classifications of examples for which the current ensemble's performance is poor. Thus Boosting differs from bagging, as in the latter the training set generation does not relate to the performance of individual classifiers. Boosting, on the other hand, assigns weights to the training instances, the values of which are changed according to how well the associated training instance is learnt by the classifier. The weights for misclassified instances are increased, and vice versa; thus, re-sampling occurs based on how well the training samples are classified by the previous model. After several cycles, the prediction is performed by taking a weighted vote of the predictions of each classifier, with the weights being proportional to each classifier’s accuracy on its training set. The boosting approach to ensembles has been implemented in a variety of forms. Examples include ADABoost and MultiBoosting:

- **ADABoost** (Freund and Schapire, 1996) exploits one of two methods to construct training sets to feed to the classifiers in the generated series. The first approach is *boosting by sampling* in which examples are drawn with replacement with probability proportional to their weights. The second is *boosting by weighting* and it can be used with base learning algorithms that can accept a weighted training set directly. This latter approach has the clear advantage that each example is incorporated in the training set.

- **MultiBoosting** (Webb, 2000) is an extension to ADABoost, described above, for forming decision committees, and uses re-weighting for each training example.

The application of "*Arguing from Experience*" to classification problems can be viewed as an ensemble-like approach. Chapter 8 will discuss an experiment designed to investigate this aspect and evaluate the operation of the promoted model against the ensemble approaches mentioned above.

### 2.3.3. Classifying Noisy Data

As noted above, the model proposed in this thesis can be treated as a form of classification of a given case. The agents will pool their arguments for and against possible classifications from their own collections of past cases. However, this experience may be infected with noise, since some previous examples will have been classified wrongly, or wrongly recorded in the database, and so an evaluation of the robustness of the advocated model to noise is seen as essential. In the following an account of noise treatment in the field of data mining is discussed. To some greater or lesser extent input data used in any data mining task contains some degree of noise. Noise may infect the input data for variety of reasons. Noise may be introduced by mistake such as data errors during data capture (formatting errors). Alternatively the introduction of noise may be, intentional, such as for reasons of privacy preservation. Whatever the case the effectiveness of data mining tasks will be influenced by the presence of noise in the data. In the case of classifier generation noisy data results in classification inaccuracies, typically caused by the "*overfitting*" of the classifier to the (noisy) data. Forthcoming chapters will demonstrate that the advocated usage of "*Arguing from Experience*" to resolve disputes over classification problems also provides an approach to classification that is very noise tolerant.

#### *2.3.3.1. Types of Noise*

Zhu and Wu (2004) provide a detailed analysis of the different types of noise that may infect input datasets. In particular they distinguish between two information sources for measuring the quality of a dataset: (i) how well the *attributes* characterise the data records for the purposes of classification, and (ii) whether the *class label* for each record is correctly assigned. It is often assumed that the class labels are *correlated* with the attributes values, and that the interactions between the attributes are *weak* so that the classifier generation algorithms are likely to ignore these interactions and consider each attribute independently to induce the classifier. Zhu and Wu (2004) argue that with real-world data it is often the case that datasets contain some attributes that have little correlation with the class, or strong interactions may exist among attributes. Accordingly, the quality of a dataset is determined by two factors: (i) an *internal factor* indicates whether attributes and the class are properly selected; and (ii) and *external factor* indicates errors introduced into attributes and the class labels (systematically or artificially). Hickey (1996) identifies three major sources of noise: (i) insufficiency of the description for attributes or the class (or both); (ii) corruption of attribute values in the training examples; and (iii) erroneous classification of training examples. The first source is often ignored as it is difficult of determine when a description for the attribute and the class labels is sufficient and when it is not. For example Zhu and Wu (2004) identify noise as non-systematic errors in either attribute values or class information. Thus most distinguish noise into two categories:

- *Attribute noise*: Represented by errors in attribute values. One notable example to be discussed later is missing attribute values.
- *Class noise*: Caused by contradictory instances where the same instances appear more than once in the dataset but are assigned different class labels; or misclassifications whereby instances are labelled with wrong classes.

#### *2.3.3.2. A discussion of some of the data cleansing solutions*

Various techniques have been proposed to deal with the influence of noise on classifiers generation. The majority of these techniques deal with class noise (e.g. (Brodley and Friedl, 1999)). Others handle attribute noise (e.g. (Quinlan, 1989) and (Zhu and Wu, 2005)). However, these methods attempt to enhance the quality of the training data in order to improve the mining process by employing some pre-processing mechanisms to handle noisy instances before a classifier is formed. This approach involves cleansing the data by exploring the dataset for possible problems and then endeavouring to correct the errors. Although data cleansing is a very useful tool, in practice, it entails some major drawbacks. Mainly, eliminating "*bad data"* is not always feasible, due to the high cost of identifying such data. Also, eliminating whole records of "*bad data*" eradicates potentially useful information.

Other pre-processing techniques have been developed to correct noisy data prior to feeding it to the mining algorithm. One notable example of these techniques is Data *Imputation* (e.g. (Sarle, 1998)) which aims to fill in missing data entries to enhance the accuracy of subsequent pattern discovery process. Although empirical studies have shown evidence that pre-processing techniques improve the overall performance of the mining algorithm (e.g. (Brodley and Friedl, 1999)), new errors may still occur as a result of pre-processing the data, and correct data records may also be falsely pre-processed resulting in information loss. Forthcoming chapters will argue that "*Arguing from Experience*" can deal with different types of noise. The advocated approach will be shown to be of particular appeal in situations where data pre-processing is infeasible or costly.

### 2.3.4. Summary of Classification in the Field of KDD

This Section has presented a general overview of the problem of classifying unseen data and its treatment in the field of tabular data mining. The key points discussed in this section are summarised as follows:

- The process of "*Arguing from Experience*", as will be made clear in later chapters, enables a number of participants to jointly reason about a given case. This operation is seen to be akin to that of classifying new data instances on the basis of gathered examples. The application of the promoted model for the purposes of classification will provide means to test the underlying debates. Additionally, this application is argued to be indeed a beneficial approach competitive with other well-known classifiers.
- A general overview of the process of a number of classification algorithms was presented. These algorithms (Table 2.2) will be used to compare the

operation of "*Arguing from Experience"* in the context of classification scenarios. The ensemble approach to classification was also discussed.

- Real world data are almost always infected with different types of noise. The effect of noise on the operation of "*Arguing from Experience*" will be addressed in Chapters 5 and 8 and the approach will be shown to be noise tolerant, so enabling reasoning from noisy data without the need for data pre-processing.

# Chapter 3: A Model for Arguing from Experience

This chapter presents a model for "*Arguing from Experience*", which is intended to enable and automate inductive reasoning from past experience. The model allows participants to draw directly from past examples to find reasons for coming to a "*view*" on some current example, without the need to analyse this experience into rules and rule priorities. As noted previously, such reasoning can be found in informal everyday arguments where these techniques are common: "*Every time we do this: that happened*", "*All the Xs that we know of, have the features a, b and c*" or "*None of the Ys we have encountered have the feature d*". In the proposed model, when arguing from past experience, rather than drawing rules from a knowledge base, Association Rule Mining (ARM) techniques are used to discover associations between features of the case under consideration and a consequent "*view*" of this case proposed according to the previous experience. A "*view*" on a current example is expressed in terms of a categorisation (classification): the discovered associations provide support for a given example to be categorised as being of a certain class.

This form of argument, using ARM, has practical appeal because the construction of an effective knowledge base for a given domain usually involves a good deal of effort. The more complicated the considered domain, the more expensive and skilled the effort required to construct the knowledge base. The so called "*knowledge engineering bottleneck*" is an established, and well recognised, obstacle in the construction of knowledge based systems. The lack (or difficulty of construction) of knowledge bases suitable for incorporation to existing systems of argumentation is contrasted with the widespread availability of large datasets where each record in the dataset represents a particular case (a particular experience). In this chapter a mechanism to deploy, in an argumentation process, the extensive amount of experience provided by these datasets is presented. In the context of these arguments each participant has access to their own dataset. In effect these individual datasets reflect the collected personal experience of each participant. Persuasion occurs, between participants, because the individual datasets (experiences) are likely to differ from one participant to another. A participant may have encountered an untypical set of examples that the other participants have not had the chance to experience. Alternatively some participants may incorrectly generalise their experience to match the current case, and only by conversing with the other participants may they correct such erroneous generalisation. This form of argument differs from the typical belief based arguments (e.g. (Prakken, 2006)) where persuasion occurs through one participant telling the other(s) something previously unknown, either a fact or a rule.

This chapter represents "*Arguments from Experience"* by means of argumentation schemes. Section 3.1 gives a detailed account of the scheme used to support such arguments, the proposed scheme is inspired by a scheme for "*Argument from Analogy based on Classification*" proposed by Walton et al. (2008). This defeasible scheme is translated into a dialogue model in Section 3.2, where attention is drawn to the speech acts associated with "*Arguing from Experience*". In Section 3.3, starting from this theory of argumentation, details about the association rules accompanying each of the speech acts from the previous section are described together with how these rules are mined from the participants' past experience. Section 3.4 combines details from Sections 3.1, 3.2 and 3.3 into one formal framework. Finally, Section 3.5 gives a summary of the issues discussed in this chapter, these issues form the basis for two dialogue game protocols named PADUA (Protocol for Argumentation Dialogue Using Association rules) and PISA (Pooling Information from Several Agents). Both systems will be discussed in Chapters 4 and 6, respectively.

## 3.1. A Scheme for Arguing from Experience

The argumentation schemes most closely related to the proposed "*Arguing from Experience*" are those focusing on the notions of analogy and classification, especially inductive versions of these schemes concerning generalisation based upon previous observations. Chapter 2 discussed the representation of

"*Argument from Analogy*" using an argumentation scheme; and the relationship between the "*Argument from Analogy*" model and other types of arguments, mainly "*Argument from Verbal Classification*", was also emphasised. This section returns to one particular scheme from those discussed in Sub-section 2.1.2.2 , namely the "*Argument from Analogy based on Classification*" scheme, and explains how it relates to the form of argumentation advocated in this thesis, before introducing the "*Arguing from Experience based on Classification*" scheme.

Recall from Chapter 2 that Walton et al. (2008) argue that "*Argument from Analogy*" is based on "*Argument from Classification*". "*Argument from Analogy*" categorises two cases, the discussion case and an analogue case, under the same class based on their similarity under a particular point of view (the analogue case is identified and retrieved from a repository of cases on the basis of its similarity to the discussion case). "*Argument from Classification*" leads to the conclusion that one case has a determined property, because it may be classified as generally having that property. Walton et al. combined these two schemes in a new scheme, highlighting this similarity, which they called a scheme for "*Argument from Analogy based on Classification*" (AAC). The AAC argumentation scheme may be summarised as follows:

> *Given a discussion case and an analogue case:*
> *The analogue has feature set A.*
> *The case under discussion has feature set A.*
> *It is by virtue of feature set A that the analogue is properly classified as W.*
> *So, the case under discussion ought to be classified as W.*

The relation between "*Argument from Classification*" and "*Argument from Analogy*" is highlighted in the work of Walton et al. (2008) by the classical example of the famous debate between Hart (1958) and Fuller (1958) where a legal rule stating that: "*No vehicles are permitted in the park*" is used to demonstrate that even an apparently clear concept such as "*vehicle*" can be legally indeterminate. According to this rule a car is a vehicle whereas bicycle is not classified as such. Walton et al. (2008) raise the question about whether a skateboard can be classified as a car or as a bicycle. The argument for either case can be presented by instantiating the (AAC) scheme. For example, one can argue that a skateboard is not a vehicle:

> *The bicycle (analogue) has no engine and low risk factor*
> *The skateboard also has no engine and low risk factor.*
> *It is by virtue of these features (the absence of the engine and the low risk factors) that the bicycle is properly classified as not vehicle.*
> *So, the skateboard ought to be classified as not vehicle.*

The critical questions (Walton, 1996) associated with the AAC scheme are derived from the ones associated with the schemes for classification and argument from analogy[9]:

**AACQ1**: Are the analogue and the case under discussion similar, in the respects cited?

**AACQ2**: Are there important differences between the analogue and the case under discussion?

**AACQ3**: Is there some other case that is also similar to the case under discussion except that it is not classified as W?

**AACQ4**: Does the case under discussion definitely have features set A, or is there a room for doubt?

**AACQ5**: Can the classification be said to hold strongly, or is it one of those weak classifications that is subject to doubt?

It is worth noting that the AAC argumentation scheme is closely related to Case Based Reasoning (CBR), particularly as applied to the legal domain (as

[9] Questions AACQ1- AACQ3 are inherited from the argument from analogy scheme. The last question comes from the argument from classification scheme. AACQ4 is common between the two schemes.

discussed in Chapter 2). In the context of legal CBR systems such as Hypo (e.g. (Ashley, 1990)) and its progeny, AACQ2 corresponds to "*distinguishing"* a case, and AACQ3 to providing a "*counter example*." The AAC argumentation scheme is not suited to the problem of "*Arguing from Experience*", mainly because it suffers from two drawbacks with respect to this particular problem. The first is that in AAC the classification of the case under discussion is warranted by its resemblance to one case only. Referring to one case only does not cover the whole past experience[10]. The second is determining how similar two cases have to be before an inference can be seen as a reasonably strong argument from the source case to the target case. Humans rely on their skills in pattern matching, and some feel for what features are important, to make such inference. CBR systems apply different similarity measures to achieve some similar pattern matching. This could suggest an additional critical question for AAC:

**AACQ6**: Are the features in common of sufficient importance to allow the case to be seen as an analogue of the example?

The "*Argumentation for Experience*" scheme presented in this chapter addresses the two disadvantages identified above. The new scheme borrows some basic elements from the AAC scheme, but replaces the notion of similarity with the notions such as the support for, and confidence in, the association, as used in the context of association rule mining.

Recall from Chapter 2 that Association Rules (ARs) are probabilistic relationships expressed as rules of the form $A \rightarrow W$ which read as follows: "*if A is true then W is likely to be true*", or "*A is a reason to think W is true*" where *A* and *W* are disjoint subsets of some global set of attributes. In the context of "*Arguing from Experience*", ARs represent a means to draw arguments from individual experiences. Such arguments (as represented by the rules) can be read as follows:

[10] This similarity needs to be present in a significant number of examples in the classification context in order for the argument to be strong.

> *In my experience, typically things with features A are Ws: this case has those features, so it is a W.*

This argument can be treated using an argumentation scheme similar to AAC scheme. This new scheme will be referred to as the "*Argument from Experience based on Classification*" (AEC) scheme, and may be defined as follows:

> *Features set A is likely to be associated with classification W.*
> *The case under discussion has the features A.*
> *Therefore, the case under discussion ought to be classified as W.*

The critical questions associated with the proposed AEC scheme are now considered. The critical questions related to the previous scheme (AAC) can be easily translated to fit the proposed scheme. Let us start with question AACQ1. This question concerns the similarity between the two cases in arguments from analogy. However, the notion of similarity has been replaced with the notion of the association between the features in the case under discussion and some classification W, the question is therefore modified accordingly:

> **AECQ1**: Do all the features in the proposed classification W match the case?

Or in other words is the association from A to W valid in the case under discussion or does it imply new features that are missing from the current case?

AACQ2 aims to identify any differences between the analogue case and the current case. In the context of associations from experience this question can be re-stated as follows:

> **AECQ2**: Are there other features in the case under discussion that weaken the association?

AECQ2 concentrates on the same issue as AACQ2 which is the difference between the case under discussion and the propose association *A*→*W*. This difference is presented here by the additional features in the case under discussion that might undermine this association.

AACQ3 looks for other cases similar to the one under discussion supporting different outcomes. In the context of "*Arguing from Experience*" this can be translated to looking for other associations based on the case under discussion, but leading to different classification:

> **AECQ3**: Are there any features in the case under discussion associated with another class X?

Questions AACQ4 and AACQ5 relate to the "*Argument from Classification*" scheme. AACQ4 focuses on the existence of A in the case under discussion, while AACQ5 focuses on the strength of the classification. In the context of the new scheme, these questions can be avoided by employing "*measures of interest (MOIs)*" (discussed in Sub-section 2.7.1) in the same manner as they are applied in the context of ARM. MOIs determine the quality of the AR. By applying these measures one guarantees that the inference from the case features to the proposed classification is strong and sound. Question AACQ4 is also addressed in this manner, as ARs are mined with the case under discussion in mind, which ensures that all the features in the set *A* are also in the case. The *support/confidence* framework (e.g. (Agrawal et al., 1993)) is applied here to evaluate the likelihood (interestingness) of ARs used in the context of the dialogue. As stated in Chapter 2, in this framework *confidence* represents the likelihood value expressed as a percentage. This is calculated as *support (XY) ×100/support(X)* where the support of an itemset (or attribute set) is the number of records in the data set in which the itemset (attribute set) occurs. To limit the number of rules generated, only itemsets whose support is above a user specified "*support threshold"*, referred to as "*frequent itemsets*", are used to generate ARs. To further limit the number of associations only those rules whose confidence exceeds a user specified confidence threshold are accepted. Taking the *support/confidence* framework into consideration, the *"Argument from Experience based on Classification*" (AEC) scheme can be rewritten as follows (AEC2):

> *Features set A is associated with classification W with an acceptable confidence C.*
> *The case under discussion has the features A.*
> *So, the case under discussion ought to be classified as W.*

The Critical questions can then be rewritten as follows:

***AECQ1**: Do all the features in the proposed classification W match the case?*

***AECQ2**: Are there any other features in the case under discussion that weaken the confidence of the association rule A→W, such that it drops below the acceptable confidence C?*

***AECQ3**: Are there any features in the case under discussion associated with another class X with confidence higher than A→W?*

The acceptable confidence (*C*) element in the above critical questions is twofold: on one hand it represents the degree to which each participant *believes* that the case under discussion should classify as a given class (W). On the other hand it represents means to give weight to the associated arguments (arguments with higher confidence are considered *stronger* than arguments with lower confidence). The participants taking sides in the argument can determine the value of the acceptable confidence (*C*) prior to starting the argument. This value relates to the setup of the problem domain. For instance, in domains where the data comprises a large number of records the confidence value may be set to near perfect (100%). On the other hand, if the number of records is limited, a lower level of confidence may be more appropriate. The following example summarises this scheme along with its critical questions: upon coming across a chicken-like serpent with eight legs, one zoologist may say:

> *Creatures with reptile-like appearance which are less than twelve fingers in length are in my experience most likely to be snakes, and I am 80% confident of what I am saying.*
> *This creature has a reptile-like appearance and its length is less than twelve fingers*
> *Therefore this creature must be a snake.*

The following "*critical questions*" can be associated with this argument[11]:

- One zoologist may say: "*Although this creature has many snake-like features, it also has a chicken like bill. I have not heard of any snake that has a chicken bill, therefore my confidence in your argument is near zero!*" (AECQ2).
- Another zoologist who happens to have Pliny the Elder's experience may say to the other two zoologists: "*Both of you are wrong, this creature has not only chicken-appearance and snake features, but it also has eight legs. In my experience only the basilisk creature has all these features and therefore I am 100% confident that this creature is a basilisk.*" (AECQ3).

This example shows the importance of confidence in "*Arguing from Experience*": our experience will suggest that things with certain features (*A*) are often/ usually/ almost always/ without exception *Ws*. This is also why dialogues to enable experience to be pooled are important: one participant's experience will be based on a different sample from that of another. In extreme cases this may mean that one person has had no exposure at all to a certain class of exceptions: a zoologist who has never encountered or even heard of basilisks will not be able to classify these creatures correctly. The new argumentation scheme introduced in this chapter is intended to aid the process of reasoning from experience, as highlighted in the introductory chapter. The specific situation being covered is where one participant is attempting to persuade other participants that a case belongs to some class *W*, and the other participants are

[11] These critical questions are presented in a way anticipating how the corresponding speech acts are derived from the questions in the following section.

arguing against this position (because they think the case should be classified differently). Because such scenarios are seen as one of conflict the following section presents a dialogue model to facilitate resolution of such disputes.

## 3.2. The Dialogue Model

One can argue that inductive arguments are presumptive in nature and that the only way to avoid fallacious applications of these arguments is by successive refinement during a dialectical process. This is the underlying idea behind associating argument schemes with critical questions. "*Arguments from Experience*" are indeed inductive arguments, based on inductive associations between features from the case under discussion and the desired classification. These associations ought to go through a process of critique and refinement within the settings of a dialogue between two or more participants, each presenting a possible classification of the case under discussion. In the following subsections a dialogue model for "*Arguing from Experience*" is introduced. This model involves a number of software agents (entities)[13] participating in each dialogue, each with their own distinct dataset of records relating to some domain. The agents produce reasons for and against classifications by mining ARs, from their individual datasets, using ARM techniques of the form discussed in Chapter 2. The potential for difference of opinion to be resolved comes from the fact that experiences (as encapsulated in the datasets) differ, and so the set of examples available to the participants may ground different conclusions with respect to a new example. This style of dialogue is persuasive by virtue of contradictions among the possible classifications of cases in the given domain. In the following, the speech acts on which the dialogue model is built are introduced. A simple protocol connecting these speech acts with each others in a logical order is also sketched. A discussion is then given of the main aspects of the proposed dialogue model.

[13] These agents will sometimes be referred to as "*players*" or "*participants*".

### 3.2.1. The Speech Acts

This sub-section considers the speech acts required for the AEC scheme and indicates how they differ from those typical of the belief based persuasion dialogues identified by Prakken (2006) (an overview of which was given in Sub-section 2.1.3). Six different "*types*" of speech acts are identified for the operationalisation of the AEC scheme (and AEC2). The inspiration for these speech acts comes from the rich field of AI and Law. In particular reasoning with precedents, as has been modelled by the HYPO system (Ashley, 1990), and its progeny, discussed in Sub-section 2.1.4. What has emerged from this work is that there are three key high level types of speech acts:

- Citing a case
- Distinguishing a case
- Providing a Counter Example.

In the following discussion each of these high level speech acts is considered in turn, highlighting how each relates to the AEC scheme discussed in the previous section. In each case the particular speech acts required to operationalise the AEC scheme is identified. The speech acts are given numbers relating them to the protocol proposed in Section 3.4.

Citing a case involves identifying a previous case with a particular outcome which has features in common with the case under consideration. In the context of "*Arguing from Experience*" this translates to a direct application of the AEC scheme linking features from the example case to some classification W: *in my experience, typically things with these features are Ws: this has those features, so it is a W.* The features in common are thus presented as reasons for classifying the example as *C*, justified by the experience of previous examples with these features. Take this argument for example:

**The Swan Argument:**

*In my experience water birds which have red bills and mate for life are highly likely to be swans.*

> *This water bird has a red bill, and was observed to have the same companion for a long period.*
> *So this bird must be a swan.*

In the context of the promoted dialogue model, this speech act is referred to as "*Proposing a new rule*" (SA1) - the speech act by which a participant proposes a new AR justifying a classification of the case under discussion. Note that the content of the speech act here is an *argument* rather than a proposition. This is also true for the other speech acts in the promoted model, since the dialogue is seen as consisting of the presentation of arguments. Performing a speech act will thus involve instantiating a particular argumentation scheme appropriate to that speech act.

The second high level speech act type identified above was "*distinguish a case*". Distinguishing is one way of objecting to the above argument, by saying why the example being considered does not conform to this pattern. It often involves pointing to features present in the case which make it atypical, so that the "*typical*" conclusions do not follow. This type of speech acts is indeed an undercutter attack (Pollock, 1995) against the link between the premises and the conclusion of an argument. For example the feature may exhibit an exception:

> *Although typically things with these features are Ws, this is not so when this additional feature is present.*

This speech act is the direct application of the critical question AECQ2, as the additional features added when distinguishing weaken the association between the premises and the classification. For example, an adversary may distinguish the previous argument by saying:

> *"Although water birds which have red bills and mate for life are highly likely to be swans; this particular bird has black feathers, in my experience water birds with black feathers are not likely to be swans".*

This speech act will be referred to as "*Distinguishing a previous rule*" or "*Distinguishing*" for readability (SA2) - the speech act by which an agent points to additional features in the case under discussion that weakens the overall confidence in a previously proposed rule.

Another kind of distinction provides a means to counter the previous speech act by supplying a more typical case: while many things with these features are Ws, experience would support the classification more strongly if some additional feature were also present. For example, the plaintiff of the swan argument may response to the previous attack by saying:

> *"Although water birds which have black feathers are unlikely to be swans in the Northern hemisphere; this particular bird comes from Australia. In my experience water birds with black feathers that have all the other features we have been talking about and live in the southeast and southwest regions of Australia are* <u>*more*</u> *likely to be Cygnus Atratus".*

This distinction speech act is called "*increasing the confidence of a previous rule*" or "*Increase confidence*" for readability (SA5) - the speech act by which an agent points to additional features in the case under discussion that increase the overall confidence in a previously proposed rule.

A third form of distinction is to find a missing feature that suggests that the case is not typical:

> *While things with these features are typically Ws, Ws with these features normally have some additional feature, but this is not present in the current example.*

This speech act translates the critical question AECQ1 to fit the dialogue context. In the swan argument example, an adversary may say that:

> *"Although black swans from Australia have all the features advocated by the plaintiff; these birds also have a pale bar on the tips of their bills, which is not the case in this particular bird we are arguing about. Therefore this bird is unlikely to be a swan".*

This speech act is referred to as "*unwanted consequences*" (SA3). Where one participant has proposed a rule *A* →*W*. If the set *W* contains any features that are not present in the current case, then an adversary can point to these features as the unwanted consequences associated with this particular rule. Agents taking part in a dialogue of the sort discussed here may retract the unwanted consequences of their propositions by trying to get around these unwanted consequences. For instance, they may try to find another rule linking the precedents to the desired classification without associating this classification with any of the unwanted consequences. This speech act is referred to as "*withdraw unwanted consequences*" (SA6).

Thus three types of distinction have been identified with differing forces. The first (SA2) argues that the current example is an exception to the rule proposed. The second (SA5) argues that the confidence in the classification would be increased if some additional features were present. The third (SA3) argues that there are reasons to think the case untypical and so it may be an exception to the rule proposed. In all cases, the appropriate response is to try to refine the proposed set of reasons to meet the objections, for example to accommodate the exception.

The last high level type of speech act identified above was "*citing counter examples*". In the context of reasoning from cases in law this is typically citing one case whose result is different from the analogue case already cited at the beginning of the argument. In "*Arguing from Experience*", this is citing an association inference (rule) that matches the features from the case under discussion to a classification other than the one promoted by the plaintiff. Counter examples differ from distinctions in that they do not attempt to cast doubt on the reasons, but rather to suggest that there are better reasons for believing the contrary (i.e. rebutters (Pollock, 1995)). The objection here is something like:

> *While these features do typically suggest that the thing is a W, these other features typically suggest that it is rather an X.*

This speech act interprets the critical question AECQ3 and applies it as a direct attack in the dialogue context. For example, a response to the swan argument may be something like the following:

> **Counter Swan (Black Goose) Argument:**
>
> *In my experience water birds which have red bills and black feathers are highly likely to be black geese.*
> *This water bird has these features.*
> *Therefore this bird must be a black goose.*

Here the response is either to argue about the relative confidence in the competing reasons, or to attempt to distinguish the counter example. This speech act is referred to as "*counter rule*" (SA4) - the speech act by which agents use features from the case under discussion to propose an argument favouring their point of view. So far the basic speech acts that a dialogue supporting argument from experience will need to accommodate have been identified, the following section introduces how these speech acts relate to each other in the context of the dialogue. The realisation of these moves using ARM techniques is left to the next section (Section 3.3).

A total of six speech acts have been identified that collectively form the basic blocks in the building of the promoted dialogue model. These speech acts fall under three basic types: (i) stating a position, (ii) attacking a position and (iii) refining one's position, as follows:

- *Starting position*:
  - *Propose Rule* (SA1): allows generalisations of experience to be cited, by which a new association with a confidence higher than a certain threshold is proposed.
- *Attacking speech acts*: these speech acts attack a previous speech act either by identifying the weak points in the underlying association rule, or by proposing a counter rule:

  - *Distinguish* (SA2): allows the addition of some new premise(s) to a previously proposed rule, so that the confidence of the new rule is lower than the confidence of the original rule.
  - *Unwanted Consequences* (SA3): allows the inclusion of the features in the consequences (conclusions) of the rule under discussion that do not match the case under consideration
  - *Counter Rule* (SA4): used in a very similar manner as propose rule (SA1) to cite generalisations leading to a different classification.
- *Refining speech acts*: enable a rule to be refined to meet objections:
  - *Increase Confidence* (SA5): allows the addition of one or more premise(s) to a rule previously played to increase its confidence.
  - *Withdraw unwanted consequences* (SA6): excludes the unwanted consequences of a rule previously proposed, while maintaining a certain level of confidence.

For each of the above six *speech acts* a set of legal next speech acts (i.e. acts that can possibly follow each speech act) is defined. Figure 3.1 highlights the possible attacks/refining moves that could be put forward as a response to each of these speech acts. Note that the proposed speech acts contrast with those found in persuasion dialogues based on belief bases, a summary of which can be found in (Prakken, 2006). Additionally, despite having strong resemblance to the speech acts used in arguing on the basis of precedent examples in common law, especially the work carried out by Ashley (1990) and Aleven (1997). The promoted speech acts differ from case based reasoning in law as all of an individual agent's experience, represented by a dataset of previous examples, is used collectively, rather than identifying a single case as a "*precedent*". Unlike legal decisions, authority comes from the frequency of occurrence in the set of examples rather than endorsement of a particular decision by an appropriate court.

```
Receive Speech Act (R:A→Q)
Switch (Type of the Speech Act)
 Case:  Propose New (Counter) Rule
  Apply Unwanted Consequences OR Distinguish OR Propose Counter (New) Rule
 Case: Distinguish
  Apply Increase Confidence OR Propose Counter (New) Rule
 Case: Unwanted Consequences
  Apply Withdraw Unwanted Consequences OR Propose Counter (New) Rule
 Case: Increase Confidence
  Apply Unwanted Consequences OR Distinguish OR Propose Counter (New) Rule
 Case: Withdraw Unwanted Consequences
  Apply Unwanted Consequences OR Distinguish OR Propose Counter (New) Rule
```

**Figure 3.1. A pseudo code highlighting how agents in "*Arguing from Experience*" dialogues can respond to each of the six promoted speech acts.**

### 3.2.2. Discussion

It is clear from the above that "*Arguing from Experience*" is a persuasion dialogue that takes place between two or more conflicting parties, each trying to prove that the case under discussion should be classified in the way they think most suitable. Two sub-models can be instantiated from this model, according to their treatment of the burden of the proof in the dialogues they generate. In the first sub-model, the participants have a positive burden of proof. In this sub-model each of them will try to prove that the case under discussion classifies according to their own thesis. This model will be referred to as the "*Dispute model for Arguing from Experience*". The other sub-model – *dissents* ((Prakken et al., 2005) - is more flexible, the burden of the proof rests only with one party, and the dialogues will progress as follows. The participant with the positive burden of the proof starts the dialogue by proposing a rule supporting its thesis. The other participants will then try to attack this rule in an attempt to prove that it is wrong, without necessarily attempting to establish their own thesis. This model will be referred to as the "*Dissents model for Arguing from Experience*". The exact details of how these two sub-models work vary according to the number of agents taking part in the dialogue. This is because the nature of the dialogue game differs between two player dialogues and multi-player dialogues. This will be discussed further in Chapters 4 and 6, respectively.

Dialogues based on sharing experience are not limited to persuasion. Other variations are also possible. For instance, participants may not be so committed to a point of view, so that the dialogue takes on the characteristics of deliberation rather than persuasion. In the deliberation version, each participant will have an idea about the possible classification of the case under discussion, which they adopt "*for the sake of the argument*", rather than a firm thesis, and therefore they will be open to suggestions from other participants, and will not attempt to refute the suggestions that they find convincing. This deliberation flavour can be achieved by the means of dialogue strategies, applying certain types of strategies leads to a more flexible "*Arguing from Experience*", closer to deliberation than persuasion. However, if other types of strategies were applied, the resulting dialogues will have a strict dispute flavour. Chapters 4 and 7 will return to this point upon considering the issue of strategies in the two-party and multiparty dialogue games, respectively. The following section steps away from the argumentation theory to focus on the realisation of the model described here by the means of association rule mining.

## 3.3. Model Implementation using ARM

Having introduced the speech acts for "*Arguing from Experience*" dialogues, and described how these speech acts relate to each other, the realisation of these speech acts using Association Rule Mining (ARM) techniques is now considered. The idea is to mine ARs according to: (i) desired minimum confidence, (ii) a specified consequent and (iii) a set of candidate attributes for the antecedent (a subset of the attributes represented by the current case). The first condition guarantees that any rule used in the dialogue satisfies the acceptable confidence condition of the argumentation scheme (or in the case of distinguishing, reducing the confidence below the acceptable level). The second condition ensures that rules used in the arguments are relevant to the dialogue goal of resolving the conflict over a classification problem, by including a possible classification in the consequents of the rules. The last condition is added so that the arguments do not include any premises that do not match the case under discussion, avoiding the need for an analogue of AACQ4.

Most of the ARM techniques discussed in Section 2.2 generate the complete set of ARs represented in the case base. This may cast unwanted overheads on the dialogue process, as many of these rules will not be required in the context of the dialogue. Instead a "*just in time*" approach to ARM is applied, where ARs are mined dynamically as required. The dynamic mining process supports three different forms of dynamic request (queries):

1. Find a subset of rules that conform to a given set of constraints.
2. Distinguish a given rule by adding additional attributes.
3. Generalise a given rule by removing attributes.

As discussed previously, past experience is represented by a dataset of raw data. Each record in this dataset represents a previous example in the considered domain, the last attributes in each record denotes the classification associated with this example. Each participant, in the promoted model, has its own dataset representing its own experience that may differ from other participants' experience. In order to achieve the three requests mentioned above, a summarising structure of this dataset is required. In the work described in this thesis the T-tree structure discussed in Sub-section 2.2.2.3 is applied.

Recall from Chapter 2 that a T-tree (Coenen et al. 2004a, 2004b) is a "*reverse*" set enumeration tree data structure. Set enumeration trees impose an ordering on items (attributes) and then enumerate the itemsets according to this ordering. T-trees are "*reversed*" in the sense that nodes are organised using reverse lexicographic ordering. The reason behind this reverse ordering is that the T-tree differs from typical set enumeration trees in that the nodes at the same level at any sub-branch of the tree are organised into one dimensional arrays so that array indexes represent column numbers, hence the "*reverse*" version of the tree enables direct indexing based on the attribute (column) number. This reverse ordering is the reason why T-trees were chosen to represent the "*Experience*" in the "*Arguing from Experience*" model advocated here. This reverse data structure comprises itemsets rooted at a particular end itemset, thus all the itemsets involving a class attribute (*W*) are contained in one branch of the T-tree (other branches are required for calculating individual AR confidence values). This means that supporting any of the desired dynamic requests requires mining ARs from one branch only at a time. This reduces the cost of processing these requests, compared to other prefix tree structures such as FP-Trees (Han et al, 2000) which store the frequent itemsets in attribute order.

To further enhance the dynamic generation of ARs a set of algorithms that work directly on P-Trees (Coenen et al. 2004a, 2004b) were developed (the nature of the P-tree data structure was discussed in Sub-sub-section 2.2.2.3). These algorithms are able to mine ARs satisfying different values of confidence threshold, and therefore correspond to the definition of Aggarwal and Yu (1998) of on-line mining. The proposed algorithms are intended to facilitate querying the dataset for ARs comprising of a determined set of items (or attributes) and are based on gradually deriving "*mini*" T-trees from the P-tree representing the underlying database. Here “*mini*” means that the tree will contain only the nodes representing attributes from the instance (case) under discussion plus the classes’ attribute. Of note here, the nodes of the produced T-tree are not pruned according to some fixed support threshold, as in the original T-tree generation algorithm. Instead only the empty nodes that have null support are deleted from the generated T-tree (in addition to the nodes representing data items (attribute) which are not present in the current instance case). To satisfy these requirements, three dynamic AR retrieval algorithms have been developed to support the “*Arguing from Experience*” protocol:

- **Algorithm A**: Finds a rule that conforms to a given set of constraints.
- **Algorithm B**: Distinguishes a given rule by adding additional attributes.
- **Algorithm C**: Revises a given rule by removing attributes.

Algorithm A is intended to find a new rule (for speech acts involving proposing a new rule) given: (i) a current instance (I) (set of items/attributes), (ii) a desired class attribute (w) and (iii) a desired confidence threshold (*Conf*). The class attribute is included as an input parameter of this algorithm for the purposes of maintaining the focus of "*Arguing from Experience*" dialogues. As stated previously, these dialogues are intended to solve a conflict with respect to the correct classification of a given case, thus there is no point in mining a rule that does not have a class attribute in its antecedent as this rule will not fit in the promoted dialogue model, and it will lead to either prolonging the dialogue, and, ultimately, to incoherent dialogues. The algorithm attempts to minimise the number of attributes in the rule and operates by first generating candidate itemsets, using the input values, in a level-wise manner, starting with 2-itemsets (one attribute from the case and the class attribute). If no "*mini*" T-tree was previously generated then the algorithm generates the next level "*mini*" T-tree from the P-tree. Otherwise for every generated itemset ($S = A \cup w$), the "*mini*" T-tree is traversed for the node representing this itemset. If such node exists, the algorithm returns rules of the form ($P \rightarrow Q \cup w$) such that ($P \cup Q = A$ and $P \cap Q = \emptyset$), if not then the algorithm proceeds to the next level. When the algorithm reaches a level that is not supported by the "*mini*" T-tree, i.e. (K+1)-itemsets while the "*mini*" T-tree contains only the first K levels of nodes; a new level is added to the "*mini*" T-tree, and the traversal procedure continues on the newly added level. The algorithm returns the first rule that satisfies the given confidence threshold, otherwise the generation process continues until the entire T-tree has been created and processed. Figure 3.2 demonstrates the pseudo code for this algorithm. In the figure, the node (set S) function returns true if the node representing $S$ is part of the "*mini*" T-tree. The confidence (Rule r) function returns the confidence of the rule r. Generate- Mini-T-tree (mini-T-tree, P-Tree, Level) generates a T-Tree of the given level (if the input mini-T-tree =null) otherwise it adds the new level to the already existing "*mini*" T-tree.

Algorithm B is applied to distinguish an input rule $r = (P \rightarrow Q)$. The algorithm operates as follows. First it generates the candidate (K+1)-itemsets, by adding new attributes from the current instance (I), one at a time ($atrr_i$), to K-item sets presenting the input rule ($S_K = PUQ$). Then the algorithm search the current "*mini*" T-tree sub-branches for the node representing this itemset, if no T-tree was yet generated, the algorithm generates the (K+1)-level "*mini*" T-tree, where $K = |PUQ|$. If such node was found the algorithm shapes a new rule r` = ((P ∪

*atrr$_i$*) → Q). If the rule confidence is lower than the input rule confidence return the rule, otherwise traverse through the sub-tree whose root is the candidate itemset for a rule of the form r`` = ( P` → Q`) that satisfies the conditions listed in the algorithm. When the algorithm reaches a level that is not supported by the "*mini*" T-tree, a new level is added to the "*mini*" T-tree, and the traversal procedure continues on the newly added level.

```
Input: The current instance I, P-Tree PT, the class w,
confidence threshold Conf, max level L.
-----------------------------------------------------------
T-Tree MT = Generate-Mini-T-tree (null, PT, level = 2)
for every possible frequent 2-itemset $S_2$ =$attr_i$ ∪ w: $attr_i$ ∈
I do
 if (node($S_2$) ∈MT) then
  if ∃ AR r ($attr_i$→w))): confidence(r)≥Conf then return r
K =2
while (K<L)do
 T-Tree MT = Generate-Mini-T-tree (MT, PT, K)
 for every possible frequent (K+1)-itemset $S_{K+1}$ such that
 (S=A ∪ w: A ⊆ I and |A|=K)do
  if (node($S_{K+1}$) ∈MT) and ∃ AR r (P→Q∪w): P∪Q=A and P∩Q=∅
  and confidence(r) ≥Conf then return r
  else K++
```

**Figure 3.2. Pseudo Code for Algorithm A (Generating New Rules).**

Figure 3.3 demonstrates the pseudo code of this algorithm. In this figure, level (T-Tree MT) function returns the level of the current "*mini*" T-Tree. A variation of this algorithm (B`) is used to generate rules for the "*Increase Confidence*" speech act. This alternative algorithm proceeds in a similar manner to algorithm B, but instead of returning rules whose confidence is less than a given threshold, it returns rules with a confidence higher than, or equal to, the agreed acceptable level.

```
Input: Mini-T-tree MT, instance I, input AR r: P→Q, P-Tree PT, confidence threshold Conf, max level L.

if (MT=null) or (Level (MT)< $|P \cup Q|+1$) then
 MT = Generate-Mini-T-tree (null, PT, level = $|P \cup Q|+1$)
for every possible frequent (K+1)-itemset $S_{K+1}=attr_i \cup P \cup Q$: $attr_i \in I$ and $attr_i \notin P \cup Q$ do
 if (the node($S_{K+1}$) $\in$MT) and ($\exists$ AR r ($attr_i \cup P \rightarrow Q$))):
 confidence(r) $\leq$Conf) return r
K =K+2
while (K<L) do
 T-Tree MT = Generate-Mini-T-tree (MT, PT, K)
 for every possible frequent K-itemset $S_{K+1}$ : ($S_{K+1}=A \cup P \cup Q$: $A \subseteq I$ and $A \not\subset P \cup Q$ and $|A|=K-|P \cup Q|$)do
  if (the node($S_{K+1}$) $\in$MT) and $\exists$ AR r (P`→Q`): P`=P∪A` and
  Q`= Q∪A`` and A=A`∪A`` and A`∩A``=∅ and P∪Q=A and P∩Q=∅
  and confidence(r)$\leq$Conf then return r
  else K++
```

**Figure 3.3. Pseudo Code for Algorithm B (Distinguishing Input Rule).**

In order to withdraw some unwanted consequences (*X*) from an input rule ($r = P \rightarrow Q \cup X$), Algorithm C first tries to produce a rule (r` = P→Q). If such a rule satisfies the confidence threshold, then the algorithm returns this rule, otherwise, the candidate itemsets are generated and rules are produced and tested in a very similar manner to Algorithm B to produce the rule (r`` = P→ Q ∪ Y ) where (X ∩ Y = ∅). Figure 3.4 demonstrates the pseudo code of this algorithm. Players may apply Algorithm C, to defend rules they have previously proposed, or to attack rules put forward by other players. Algorithm C therefore takes the status of the player into consideration, so that if the player is defending its point of view the algorithm searches for rules whose confidence is equal or higher than the input rule. On the other hand, if the player is attacking the opponent's position, then the confidence of the returned rule must not be higher than the confidence of the input rule.

```
Input: Mini-T-tree MT, instance I, input AR r: P→Q∪X),
P-Tree PT, confidence threshold Conf, max level L,
defending status ds.
---------------------------------------------------------
if (MT=null) or (Level (MT)<|P∪Q∪X|+1)then
 T-Tree MT = Generate-Mini-T-tree (null, PT, level =
 |P∪Q∪X|+1)
if node(P∪Q) ∈MT then
 if (ds) then
  if ∃ AR r (P→Q))): confidence(r)≥Conf then return r
 else
  if ∃ AR r (P→Q))): confidence(r)≤Conf then return r
else
  if (ds)then
   apply algorithm B`(I, r`(P→Q), PT, Conf, L)
 else
  apply algorithm B (I, r`(P→Q), PT, Conf, L)
```

**Figure 3.4. Pseudo Code for Algorithm C (Withdraw Unwanted Consequences).**

## 3.4. General Framework for Arguing from Experience

A formal framework for "*Arguing from Experience*" is now introduced. This framework borrows various elements from the formal systems suggested by Amgoud et al. (2006) McBurney and Parsons (2002) and Prakken (2000, 2005) and employs these elements in the context of the dialogue model discussed earlier. It is worth noting that Ontañón and Plaza (2006) propose an argumentation framework for learning agents: this framework is similar to the one proposed here in taking the experience, in the form of past cases, of agents into consideration and focusing on the argument generation process. The protocol presented here differs in that the protocol they suggest applies learning algorithms techniques, while the protocol advocated below implements simpler ARM techniques to produce arguments. Also the protocol advocated by Ontañón and Plaza (2006) is designed for pairs of agents that collaborate to decide the joint solution of a given problem, while the one outlined here can be used by any number of agents.

McBurney and Parsons (2002) propose a three-level hierarchical formalism to support dialogues occurrences which consist of combinations of different types of dialogues. Their structure comprises three layers: The first is called the "*Topic Layer"*, and it represents the matters under discussion. The second is the "*Dialogue Layer"*, which encapsulates the component of different atomic types of dialogues. The third is the "*Control Layer"* which represents the selection of specific dialogue types and transition between these dialogues. Although "*Arguing from Experience*" is persuasive in nature, nevertheless the three-layer framework proposed by McBurney and Parsons (2002) provides a structural frame to combine the various elements of the promoted dialogue model. The "*Control Layer"* element of this three-layer framework is particularly important to the dialogue model proposed in this Chapter. Although this layer is originally intended to manage the transition between different types of dialogues, it will be exploited to achieve a different purpose in the context of "*Arguing from Experience*". For such dialogues, a control layer is necessary to facilitate the flow of the dialogue games so that they satisfy the requirements for multiparty dialogue games. The specification of this layer is as a facilitator is discussed in detail in Chapter 6. Additionally, this layer can be installed to control the transition, not between types of dialogues, but between different topics of discussion. This latter feature is fully explained in the next chapter, in the context of arguing over intermediate classifications.

Thus, the "*Argument from Experience*" dialogue model is configured using the three-layer structure discussed above as illustrated in Figure 3.5. The Topic Layer in this model is identified by three parameters considered as essential for defining topics in the context of "*Arguing from Experience*", denoted by $\tau$, as a tuple $\tau$ = <*I*, *D*, Conf, *ar*> (Wardeh et al., 2009a) where:

- $I = \{i_1, i_2,.., i_n\}$ is the set of attributes (or items in case of binary valued attributes). Each attribute (item) $i_k \in I$ has a set of possible values $V_i = \{v_{i1}, v_{i2}, v_{im}\}$.
- $D$ = the set of database records, each record $R \in D$ is a subset of the items in $I$. A record $R$ satisfies a set of items $X \subseteq I$ if and only if $X \subseteq R$.

- *Conf* = *"Confidence Threshold",* representing the lowest acceptable confidence, rules with confidence lower than this threshold are considered invalid arguments.
- *ar* = *"association rule"* written as *ar(P→Q, c)* such that:
  - $P \subseteq I$: the premises of the rule - P= {($i_{p1}$, $v_{x1}$), ($i_{p2}$, $v_{x2}$)… ($i_{pk}$, $v_{xk}$)} - such that for each tuple ($i_{ph}$, $v_{xj}$) $i_{ph} \in I$ and $v_{xj} \in V_i$.
  - $Q \subseteq I$: the rule's conclusion - *Q= {($i_{q1}$, $v_{y1}$), ($i_{q2}$, $v_{y2}$)… ($i_{ql}$, $v_{yl}$)}* - such that for each tuple ($i_{qh}$, $v_{qj}$) $i_{qh} \in I$ and $v_{qj} \in V_i$.
  - $P \cap Q = \varnothing$ (The empty set).
  - *c*: rule confidence, which means that *c%* of the transactions in *D* that contains *P* contains *Q* also (i.e. the conditional probability of *Q* given *P* as identified by Agrawal et al (1993)).

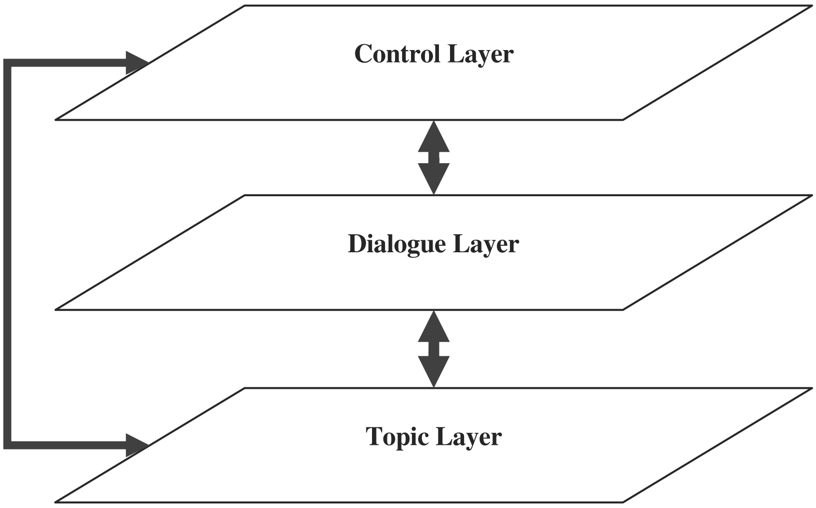


**Figure 3.5. The proposed three-layer structure.**

The Dialogue Layer presents the central layer in this structure. It includes the components of the dialogue games identified in Sub-section 2.1.3. In particular this layer takes into consideration each of the following components: commencement rules, locutions, combination rules and termination rules. Recall from Chapter 2 that commencement rules state the conditions under which the dialogue commences. Locutions indicate what speech acts are permitted. Combination rules define the dialogical contexts under which particular locutions are permitted or not. Finally, termination rules define the conditions for ending an ongoing dialogue. The Dialogue layer also encompasses the participants taking part in the dialogues. The rules for joining and leaving an ongoing dialogue are not, however, carried out within this layer, but are left to be handled by the Control Layer. Prakken identified each of these components for two-party persuasion dialogues in Prakken (2000, 2005). Amgoud et al (2006), on the other hand, present general settings for any dialogue protocol. For the purposes of constituting a formal setup for "*Arguing from Experience*" dialogues, a combination of these systems is considered as the elements of interest from each model are identified and joined, while the other components of these systems are kept aside.

Taking the above discussion into consideration the Dialogue Layer can now be identified as a tuple $\pi = < L_c, A, \varphi, DP, E, P, O >$ (Wardeh et al., 2009a) where:

- $L_c$ is the communication language for the "*Arguing from Experience*" dialogues. $L_c$ is denoted as tuple $L_c =<SA, M, DM, D>$ such that:
  - SA represents the Speech Acts discussed in Section 3.2.1 and is identified as SA = {propose rule, distinguish, unwanted consequences, counter rule, increase confidence, withdraw unwanted consequences}.
  - M denotes the set of all the possible moves. A move $m \in M$ is defined as a tuple m=<sa, content> such that:
    - sa$\in$SA is the move's speech act.
    - content is the content of this move:
      - If (sa ≠Unwanted Consequences): content = ar(p→Q, c).
      - If (sa = Unwanted Consequences): content = U⊂I (the set of unwanted consequences).
  - *DM* is the set of all dialogue moves such that each dialogue move $dm \in DM$ is defined as a tuple $dm=<a, H, m, t>$ where:
    - $a \in A$ is the agent that utters the move, given by $Speaker(dm) = a$.
    - $H \subseteq A$ denotes the set of agents to which the move is addressed, given by a function $Hearer(dm) = H$.
    - $m \in M$ is the move, given by a function $Move(dm) = m$.

  - $t \in DM$ is the target of the move i.e. the move which it replies to, given by a function *Target(dm)=t*. $t = \varnothing$ if *M* does not reply to any other move (initial move).
- *D* is the set of all finite sequences from $L_c$. For any dialogue $d = \{dm_1 \dots dm_n\}$, the speech act of the first move ($dm_1$) is a propose rule. The dialogue move $dm_n$ denotes the final move in the dialogue d, the winner of the dialogue is identified as $Speaker(dm_n)$. For any dialogue $d = \{dm_1 \dots dm_n\}$, the speech act of the first move ($dm_1$) is a propose rule. The current dialogue, denoted $d_{current}$, is the actual dialogue taking place between the set of participants taking part in every instantiation of the framework.

- $A=\{a_1,..a_n\}$**:** the set of agents participating in the dialogues identified by the communication language *Lc*. These agents are referred to as players. Each agent (player) $a \in A$ is defined as tuple $a = <name_a, C_a, \Sigma_a, CS_a, S>$ Where:
  - $name_a$: the agent (player) name.
  - $C_a$: the set of classes this player tries to prove that the discussed cases fall under. Each class $c \in C_a$ is a tuple <*name, value*> where name is an item $i \in I$, and v*alue* $\in$ *Vi* is the value this item the participant agent tries to prove it holds.
  - $\sum_a$: is a representation of the underlying Topic Layer. This representation enables participants to mine for the suitable ARs as needed. For example $\Sigma_a$ might be represented as the following tuple: $\Sigma_a = < T_a, R_a, Dr_a >$. Where: $T_a$ is the T-Tree representing the background database of the agent, $R_a$ is the set of ARs previously mined by this player (i.e. $R_a = \{ar: ar(P \rightarrow Q, conf)\}$), and $Dr_a$ is a function that maps between legal moves and their suitable rules: $Dr_a : T_a \times M \rightarrow R$, where R is the set of all possible ARs.
  - *S:* The Strategy function[14].
  - $CS_a$**:** is the player commitment store.

[14] This issue is discussed in details for two-party and multiparty arguing from experience dialogues in Chapter 4 and 7 respectively.

- $\varphi$*:* The instance argued about i.e. the dialogue subject. This instance is identified in the Topic Layer as tuple $\varphi = \{(i\varphi_1, v\varphi_1), (i\varphi_2, v\varphi_2)\ldots (i\varphi_k, v\varphi_k)\}$, such that for each tuple $(i\varphi_h, v\varphi_j)$ $i\varphi_h \in I$ and $v\varphi_j \in V$.
- *DP:* Is the *dialogue purpose*, defined as the resolution of conflicting opinions about the classification of the instance $\varphi$. This purpose is met when the dialogue is terminated, and is identified with the classification proposed by the winner of the dialogue game.
- *E:* is the set effect rules for $L_c$, specifying the effects of each move $dm <a, H, m, t> \in DM: a \in A$ on the commitments of the participants. These rules are specified in Table 3.1.
- *P* is a protocol for $L_c$ specifying the legal moves at each stage of a dialogue. P is formally defined as the function: $P: M \rightarrow 2^M$, where *M* is the set of dialogue moves. Thus the dialogue protocol P indentifies the combination rules for the dialogue games taking place under this framework. Table 3.2 summarises these rules and indicates where a new set of reasons is introduced to the discussion[15]. The given rules link each speech act with a set of possible next moves that are legal in the context of "*Arguing from Experience*" dialogues.
- *O* donates the outcome rules of the dialogues. These rules define for each dialogue *d* and instance $\varphi$ the winners and losers of *d* with respect to $\varphi$. The winners of *d* are identified as the participants whose goals match O(d, $\varphi$), and the losers are participants whose goals do not O(d, $\varphi$). However, the exact definition of the outcome of the dialogue and therefore the exact specification of the rules of such outcome, differ according to the number of participant taking part in "*Arguing from Experience*" dialogues. Therefore a separate discussion will be given with regard to outcome rules in the context of two-party games (Chapter 4) and multiparty games (Chapter 6).

[15] Any dialogue game protocol should take into consideration two essential issues: how to terminate the current game and the turn taking policy applied in the dialogues. However, termination or turn taking rules are not addressed here as these rules differ substantially between two-party and multiplayer games. An extensive account of these rules will be given in Chapter 4 and 6 respectively.

| Rule | Played move | Effects |
|---|---|---|
| **E1** | Propose rule , Counter rule, Increase Confidence ar(P→Q, conf). | $C_a = C_a \cup Q$.<br>$\forall h \in H$, $C_h = C_h$. |
| **E2** | Unwanted consequences (U). | $C_a = C_a$.<br>$\forall h \in H$, $C_h = C_h - U$. |
| **E3** | Distinguish, increase confidence. | $C_a = C_a$.<br>$\forall h \in H$: $C_h = C_h$. |
| **E4** | Withdraw unwanted consequences (P→Q', conf). | $C_a = C_a \cup Q'$.<br>$\forall h \in H$ $C_h = C_h$. |

**Table 3.1. The effect rules for the proposed protocol (Wardeh et al., 2009a).**

| Move (speech act) | Label | Next Move | New AR |
|---|---|---|---|
| 1 | Propose Rule | 3, 2, 4 | Yes |
| 2 | Distinguish | 3, 5, 1 | No |
| 3 | Unwanted Cons | 6, 1 | No |
| 4 | Counter Rule | 3, 2, 1 | Nested dialogue |
| 5 | Increase Conf | 3, 2, 4 | Yes |
| 6 | Withdraw Unwanted Cons | 3, 2, 4 | Yes |

**Table 3.2. Arguing from Experience dialogue legal moves (Wardeh et al., 2007a).**

## 3.5. Summary - Properties of Arguing from Experience

In this chapter the proposed argument scheme for "*Arguing from Experience*" has been described. This scheme was called the "*Argument from Experience based on Classification*" scheme and it is the derivation of a desired claim from the case under discussion by the means of ARs linking some features in the case to the claim. These rules are mined from a pool of past examples. These examples (experiences) provide the backing for the warrant in this scheme. Two versions of this scheme (AEC) and (AEC2) were described and the critical questions (CQs) associated with each of them were identified. The CQs associated with AEC2 were rewritten as speech acts to accompany the proposed dialogue model. This model was inspired by legal reasoning from precedents. However, the model differs from case based models, especially those used in legal case based reasoning, in that participants argue using generalisations of their experience (in the form of ARs) rather than relying on citing one case at a time. Details were also presented of the realisation of the speech acts associated with the promoted dialogue model using dynamic ARM requests. Finally, a formal framework was introduced summarising the key ideas of "*Arguing from Experience*" as applied in this thesis.

The argumentation model proposed in this chapter can be applied in situations where participants have not analysed their experiences into rules and rule priorities (knowledge base), but draw directly on past examples to find reasons for coming to a view on some current example. One classic example of such reasoning is found in common law, especially as practiced in the US, where arguments about a case are typically backed by precedents. This approach features several advantages:

- Such arguments are often found in practice. Many people do not develop a theory from their experience, but when confronted with a new problem recall past examples.
- It avoids the knowledge engineering bottleneck that occurs when belief bases must be constructed.
- There is no need to commit to a theory in advance of the discussion. The information can be deployed as best meets the need of the current situation.
- It allows agents to share experiences that may differ, one agent may have encountered types of case that another has not. This is why it important that each agent uses its own database.

The model proposed in this chapter can now be specified in terms of two-party and multiparty settings. These specifications will include the exact instantiation of the dialogue game protocol briefly mentioned in Section 3.4. The next chapter will articulate the two-party incarnation of this model. The multiparty incarnation is studied in detail in Chapters 6 and 7.

# Chapter 4: The PADUA Protocol

The previous chapter presented a theoretical model describing the foundation for a generic dialogue game protocol to facilitate "*Arguing from Experience*". This chapter articulates the advocated protocol for two-party scenarios. This manifestation is called PADUA (Protocol for "*Arguing from Experience*" Dialogues Using Association rules). The PADUA protocol will enable persuasive dialogues to be undertaken by two participants to resolve a classification problem. A proponent of a possible classification may state and justify their proposal in the form of the AEC2 scheme, and the opponent may attack this position according to the speech acts presented in Chapter 3. The result of dialogue games of this form will be the classification of the considered problem as proposed by the winning party. As a two-party "*Arguing from Experience*" protocol, PADUA necessarily has significant differences from the existing protocols designed to argue about knowledge represented as knowledge base of rules, which is the approach taken by the majority of existing dialogue systems. An excellent survey of these systems can be found in (Prakken, 2006). The resulting dialogues have a flavour akin to dialogues related to CBR in law.

The details of the PADUA protocol are described in Section 4.1. A description of a system that implements the PADUA protocol to mediate dialogues from experience between two software agents is also given. Section 4.2 provides a detailed discussion of the PADUA problem solving strategies, and how different dialogue flavours can be derived by applying different strategies. Section 4.3 gives a brief discussion of how accrual of arguments can be embodied in PADUA. Section 4.4 tackles the issue of *intermediate predicates*, particularly when the truth of such predicates cannot be functionally determined by some base level predicates. The proposed solution is accommodated in PADUA through the possibility of nested dialogues. Section 4.5 concludes with a summary of the key points.

## 4.1. PADUA from Theory to Application

This section explains how the theory presented in Chapter3 is applied to two-party scenarios by the means of the PADUA protocol. Following the discussion on the framework presented in Section 3.4 the PADUA protocol may be conceptualised in terms of three layers: a topic layer, a dialogue layer and a control layer. The dialogue game derived from the dialogue layer is between a proponent and an opponent of a classification of some case (C) in some domain (dataset) (*D*). The proponent claims that the case falls under some class ($c_1$), while the opponent opposes this claim, and tries to prove that case actually falls under some other class ($c_2 = \neg c_1$). The game participants are represented by software agents (entities). These agents are referred to as *players* in the context of PADUA games. Each player relies on its own experience to draw *"Arguments from Experience"* based on Association Rules (ARs). This experience is represented by a collection of raw data related to the problem domain (*D*). The representation of the players (agents) was given in Section 3.4. In PADUA, however, the set *A* comprises two agents: *A= {Proponent, Opponent},* where $c_1 \in C_{Proponent}$ *and* $c_2 \in C_{opponent}$. PADUA is therefore applicable in two-class domains. Another area where PADUA may be of use is when the proponent is trying to push forward some claim (classification) while the opponent aims to prove that this particular claim does not hold. Here the opponent tries to undermine the proponent's proposal. That is, proving that the case does not classify as ($c_1$) rather than proving it classifies as some other class ($c_2$). The topic layer is the same as explained in Section 3.4 while the specification of the control layer will be discussed in Section 4.4. Furthermore, the "*outcome rules"* (*O*) of PADUA dialogues define for each dialogue *d* and instance $\varphi$ the winners and losers of *d* with respect to $\varphi$. In PADUA, the winner of a given game is identified as the participant whose goal matches the output of the dialogue, and the loser is identified as the participant whose goal does not match this output. The outcome of the dialogue is defined as the class attribute of the association rule of the last move played in this dialogue. *O* consists of two functions $w_\varphi$ and $l_\varphi$. The first returns the winner of the game and the second returns the loser of the game (Wardeh et al., 2009a):

- $w_{\varphi}(d, a \in A) = true$ if $Ga = O(d, \varphi)$.
- $l_{\varphi}(d, a \in A) = true$ if $Ga \neq O(d, \varphi)$.
- O $(d \in D, \varphi) = o$: $o \in G_{pro} \cup G_{opp}$ and o $\in$ consequences of the content of the last move played in *d*.
- The two functions wφ and lφ satisfy the following conditions:
  - $w_{\varphi}(d, A) \cap l_{\varphi}(d, A) = \varnothing$.
  - $w_{\varphi}(d, A) = \varnothing$ *iff* $l_{\varphi}(d, A) = \varnothing$.
  - $w_{\varphi}(d, A)$ and $l_{\varphi}(d, A)$ are at most singletons.

The proponent starts the dialogue game by *proposing a new rule*[16] ($R_1$: $P \rightarrow Q$), to instantiate the AEC2 argument scheme. As discussed in Chapter 3, the premises (P) should match the case, while the conclusion (Q*)* supports the agent's position ($c_1 \in Q$). Once the initial position has been stated the opponent should place a legal move that can undermine the initial rule proposed by the proponent. As soon as the opponent plays its move the turn goes back to the proponent to defend its original position (using a legal move). The game proceeds in this manner until one player has no adequate reply. This player then loses the game, and the other wins. Taking this scenario into consideration the PADUA *Termination* and *Turn Taking* rules can now be identified:

- *Termination Rules*: The dialogue game terminates once one player fails to put forward a legal move. To guarantee that dialogue games will always terminate, PADUA forbids the players from proposing the same AR twice. Acknowledging this rule, the definition of legal moves is slightly different in PADUA than in the generic model discussed in Chapter 3. The new definitions are outlined in Table 4.1. This restriction is logically sound: there is no point of proposing a rule that has already been defeated, as it would be overcome in the same manner again and again. This will only lengthen the dialogue game (or result in an infinite game) without adding any real value to the dialogue itself. On the other hand, a distinguish move

[16] Rule here means an AR, so that P→Q should be read as "*P is a reason for Q*", rather than "*P materially implies Q*" or similar. However, premises and conclusions of these ARs will be spoken of: since if accepted the association will *become* a rule.

can be played more than once to undermine different rules (which is guaranteed by the previous condition). The above constraint differs from other forms of constraints proposed in the literature. For instance, Prakken and Sartor (1997) restrict their argument game such that the proponent is not allowed to repeat their moves while the opponent may do so.

| **Move** | **Speech act** | **Next Move** | **Condition** |
|---|---|---|---|
| **1** | Propose Rule | 3, 2, 4 | The player has not played the same move before. And the new AR is not part of any other move previously played. |
| **2** | Distinguish | 3, 5, 1 | |
| **3** | Unwanted Consequences | 6, 1 | |
| **4** | Counter Rule | 3, 2, 1 | The player has not played the same move before. And the new AR is not part of any other move previously played. |
| **5** | Increase Conf | 3, 2, 4 | The player has not played the same move before. And the new AR is not part of any other move previously played. |
| **6** | Withdraw Unwanted Cons | 3, 2, 4 | The player has not played the same move before. And the new AR is not part of any other move previously played. |

**Table 4.1. The legal Moves of the PADUA protocol.**

- *Turn taking Rules*: PADUA applies a simple turn taking policy, by which each player is allowed one move per turn. Thus, PADUA does not support arguments which premises are not in the case under discussion as such arguments involve more than one move per turn. For example, PADUA does not support the following type of arguments: "*The given case has feature x, feature x is associated with feature y which is not present in the current case. Feature y is associated with classification W therefore the case should be classified as W.*"

Note that the dialogue scenario discussed above is an instantiation of the sub-model: "*Dissents model for Arguing from Experience*", described in Sub-section 3.2.2, in which the burden of the proof rests with the proponent. The proponent starts the dialogue with a positive argument, and has to strengthen this position either by proposing new rules or by increasing the confidence of distinguished rules. Both moves concern proving that the case under discussion classifies as $c_1$. This is not necessarily the case for the opponent, which may go through an entire dialogue game undermining the proponent's propositions without proposing any positive argument to prove that the case classifies as ($c_2$). This is possible because PADUA is specified such that classifications proposed by the proponent and the opponent negate each other. Sub-section 3.2.2 identified another sub-model: "*Dispute model for Arguing from Experience*", in which all the participants have a positive burden of the proof. This sub-model can also be instantiated for PADUA. The opponent cannot win the game by simply undermining that proponent position (by playing distinguish or unwanted consequences moves only). Rather, the opponent should propose at least one rule suggesting that the case under discussion should classify as ($c_2$) (counter rule). The PADUA protocol can accommodate both these sub-models. On one hand it allows for "*disputes*" to take place between two players, each with its own positive burden of proof. On the other hand it allows for a "*weaker*" version of persuasion dialogues – "*dissents*", where the burden of proof rests with the proponent. In disputes, winning a game involves proposing arguments drawn from the AEC2 scheme. These arguments should be stronger than the ones proposed by the opposite side. As discussed in Chapter 2, winning in dissents dialogues involves undermining the other side's position to the point where the opposite side cannot defeat that position any more. This concludes the discussion about manifesting the theoretical model of Chapter 3 into two-party dialogues. In the following a detailed account is given of the implementation and the exploitation of PADUA to enable two-party dialogues over the classification of some case.

### 4.1.1. Implementation of the PADUA Dialogue Game Protocol

This section describes an implemented system which takes the form of a dialogue game and embodies the PADUA protocol articulated above. The objective of the implementation presented here is to provide a proof of concept for PADUA, and to enable empirical investigation of the efficacy of the protocol. This application will be used to generate a variety of example arguments used throughout the rest of this chapter. The implementation also represents a step towards the ultimate goal of allowing "*Arguing from Experience*" dialogues to take place between two software agents (players) over the classification of different cases derived from different domains. The system presented here will be studied thoroughly in the next chapter. A more detailed description of the implemented system, as well as the accompanying design documentation, can be found in Appendix A.

PADUA has been implemented in the form of a Java program: a brief description of how the system functions is also given. The software implements the protocol so that dialogues between two players can be undertaken with each player taking turns to propose and attack positions by uttering the speech acts specified in Section 4.2. The GUI interface enables the user to import a game dictionary, which is a brief schema describing the problem domain, a description of this schema can be found in Appendix A. The user can then choose the background dataset for each of the two players, and a case to argue about. The user has the option to change the *support/confidence* values for both players, and any other strategy parameters via a special window (these parameters are described in the following section). A dialogue game then takes place between the two players and the results of this game, together with the actual dialogue, are printed to a special tab screen. The underlying software comprises two major components: a "*Dialogue Game Facilitator"* and a "*Player Agent*", from which two players are instantiated. The "*Dialogue Game Facilitator"* provides a forum for the dialogue game to take place between the two "*Player Agents*" (namely the "*Proponent Agent*" and the "*Opponent Agent*") over the classification of the case chosen by the user. The "*Player Agent*" unit forms the base blueprint (class in object oriented terms) from which agents presenting one of the two players can be instantiated. Each "*Player Agent"* functions as an autonomous unit, and comprises:

- *Rule Mining Unit*: provides the players with means to mine ARs according to their needs. This unit translates the player's dataset (experience) into P- and T-tree structures and provides means to mine ARs from these structures.
- *Dialogue Unit*: is the basic unit in the "*Player Agent*" skeleton. It provides the vital functions needed to take part in the dialogue game mainly: a "*Play*" function that places moves according to the player strategy, the given case and the background dataset. The legality of each move is considered at this

stage such that the player is not allowed to place any move that does not follow the rules of PADUA.

The proponent starts the dialogue by proposing a new rule. If the proponent fails to propose a new rule, the "*Dialogue Game Facilitator*" checks the dialogue game style. In the case of "*disputes*", the facilitator gives the turn to the opponent. If the opponent also fails to generate any rules, the game terminates. In the case of "*dissents*" the game terminates with a failure. This failure can be avoided by setting up the values of the *support/confidence* to match the domain under consideration, so that mining an initial AR is always guaranteed. Of course such pre-determination of the values requires a heuristic study of the domain prior to the start of the game. This may not always be feasible. A better solution will be to allow the "*Player Agents*" to dynamically change the values of the *support/confidence* thresholds if they failed to mine rules that match the case at the initial stage of the dialogue game, or to allow the user to try again with different settings for these parameters.

Once the initial rule is proposed, a special repository called the "*Game History*" is updated with this move. "*Game History*" has a double functionality: firstly it keeps track of the moves placed by each participant, and makes sure that players are not reusing rules they have been proposed at a previous stage of the game. Secondly it provides a simple commitment store such that each player is committed to the consequences of its moves. The "*Dialogue Game Facilitator*" terminates the dialogue when the proponent fails to defend its position, or when the opponent fails to attack the proponent's position. Once the game is terminated the dialogue game moves, along with the resulting classification, are printed to the output screen of the GUI application. For the purposes of readability the dialogue presentation takes the following format:

```
Round R:
Player (Class) – The Speech Act:
Textual representation of the move.
```

Round R, indicates the round number of the dialogue game. Player, proponent or opponent, is followed by its advocated classification. This is followed by the speech act associated with the move the player has put forward in round R. Finally the textual representation of the move is a clear, easy to read and interpret, text describing the move the player has placed forward in round R, in terms of the association rule and associated confidence value.

### 4.1.2. PADUA Dialogue Example

This section provides a brief example demonstrating how the above implementation functions. This example also gives an insight to the style of the dialogues produced by the PADUA protocol. To illustrate the resulting dialogues, PADUA is applied to a fictional housing benefit scenario, where benefits are payable if certain conditions showing need for support for housing costs are satisfied. This scenario is intended to reflect a fictional benefit, Retired Persons Housing Allowance (RPHA), which is payable to a person who is of an age appropriate to retirement, whose housing costs exceed one fifth of their available income, and whose capital is inadequate to meet their housing costs. Such persons should also be resident in this country, or absent only by virtue of "*service to the nation*", and should have an established connection with the UK labour force. Whilst fictional, these conditions are very similar to those found in actual welfare benefit regulations. These legislative conditions need to be interpreted and applied by those adjudicating claims to benefits, typically using a set of guidelines (example interpretations are given by Bench-Capon (1991,1993)). The following desired interpretations were used:

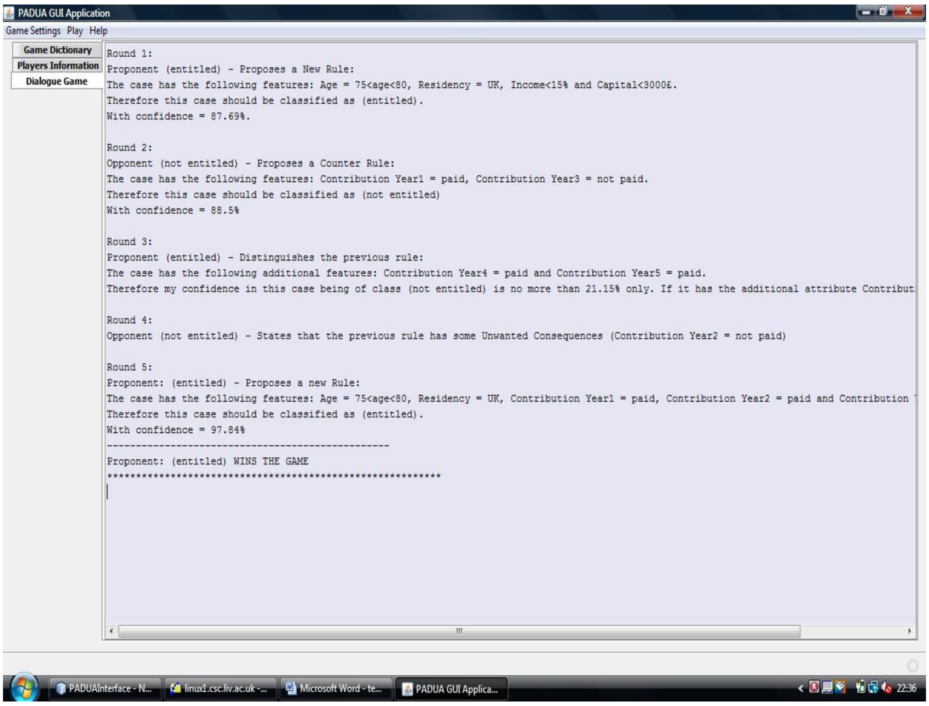
1. *Age condition*: "*Age appropriate to retirement*" is interpreted as pensionable age: 60+ for women and 65+ for men.
2. *Income condition*: "*Available income*" is interpreted as net disposable income, rather than gross income, and means that housing costs should exceed one fifth of candidates' available income to qualify for the benefit.
3. *Capital condition*: "*Capital is inadequate*" is interpreted as below the threshold or another benefit.
4. *Residence condition*: "*Resident in this country*" is interpreted as having a UK address.

5. *Residence exception*: "*Service to the Nation*" is interpreted as a member of the armed forces.
6. *Contribution condition*: "*Established connection with the UK labour force*" is interpreted as having paid contributions in 3 of the last 5 years.

The above conditions fall under a number of typical condition types: Conditions 2 and 3 represent necessary conditions over continuous values. Conditions 4 and 5 represent a restriction on the applicant's residency and an exception to this restriction. Condition 1 deals with variables depending on other variables and condition 6 is designed to test the cases in which it is sufficient for some *n* out of *m* attributes to be true (or have some predefined values) for the condition to be true. Examples of these sorts of conditions can be found in the actual legislation governing welfare benefits in the UK.

Let us now assume that there are two different offices providing RPHA services in the same region, each has a dataset of 12,000 benefit records. Each dataset was assigned to a PADUA player. Corresponding ARs were mined from these sets using a 70% confidence threshold and 1% support threshold for both players. In this example PADUA was applied to the case of male applicants aged around 80 years, a UK resident whose capital and income falls in the right range, and who has paid contributions in four out of the last five years (has not paid the contribution three years ago). Figure 4.1 shows the result of applying PADUA to this particular case. A more detailed account of how this example was produced can be found in Appendix A.

This example shows how the PADUA application can effectively construct meaningful dialogues explaining the reason behind assigning an advocated classification to each input case. No intervention, on the behalf of the user, is necessary beyond the input activities. The dialogue games between the assigned two parties will continue automatically until an agreement is reached. The user can then inspect the resulting dialogue. If the result of a dialogue does not satisfy the users' expectations then they could reapply PADUA using different input parameters. For instance, by changing the strategy of one of the players, or both strategies, or changing the values of the *support/confidence* thresholds. The advocated application is useable by two target audiences: the first includes those who are interested in examining the structure of the dialogues produced by the system. The second includes those who are concerned with the final results of these dialogues: the proposed classifications of the case under discussion, and the accuracy of these classifications. Having described the structure of dialogues generated by PADUA, the rest of this chapter provides a comprehensive analysis of this structure together with illuminating examples.



```
Round 1:
Proponent (entitled) - Proposes a New Rule:
The case has the following features: Age = 75<age<80, Residency = UK, Income<15% and Capital<3000£.
Therefore this case should be classified as (entitled).
With confidence = 87.69%.

Round 2:
Opponent (not entitled) - Proposes a Counter Rule:
The case has the following features: Contribution Year1 = paid, Contribution Year3 = not paid.
Therefore this case should be classified as (not entitled)
With confidence = 88.5%

Round 3:
Proponent (entitled) - Distinguishes the previous rule:
The case has the following additional features: Contribution Year4 = paid and Contribution Year5 = paid.
Therefore my confidence in this case being of class (not entitled) is no more than 21.15% only. If it has the additional attribute Contribut

Round 4:
Opponent (not entitled) - States that the previous rule has some Unwanted Consequences (Contribution Year2 = not paid)

Round 5:
Proponent: (entitled) - Proposes a new Rule:
The case has the following features: Age = 75<age<80, Residency = UK, Contribution Year1 = paid, Contribution Year2 = paid and Contribution
Therefore this case should be classified as (entitled).
With confidence = 97.84%
-------------------------------------------------
Proponent: (entitled) WINS THE GAME
**********************************************************
```

**Figure 4.1. Output screen showing the PADUA Dialogue Example.**

## 4.2. Strategy and Tactics in PADUA

The interaction of arguments in PADUA is viewed as a form of dialogue game that has aspects of both persuasion and deliberation dialogues. The balance as to which type dominates the game differs according to the dialogue strategies employed. Recall from Chapter 2 that "*formal dialogue games*" (e.g. (McBurney and Parsons, 2002)) are interactions between two or more players, where each player moves by making utterances, according to a defined set of rules known as a "*dialogue game protocol*", which gives the set of possible moves expected after a previous move. Choosing the best move among this set of possible moves is the "*strategy problem*". This section is concerned with the suggested solution for the "*strategy problem*" in "*Arguing from Experience*", embodied in PADUA, and demonstrates how the two participants taking part in PADUA games can be represented as cooperative or adversarial agents. How different strategies give rise to different flavours of dialogue is also discussed. Some of the dialogues illustrated in this section have the flavour of persuasion dialogues, others of deliberation dialogues. These two distinct types of dialogue, identified by Walton and Krabbe (1995), can be realised in the same protocol when different strategies are applied. Deliberation dialogues are marked by the participants attempting to reach the right decision rather than insisting on their own points of view: the more ready a player is to accept the arguments of its opponent when they seem reasonable, the more the game will take on the flavour of a deliberation.

In PADUA, each player must select the kind of move to be presented in the dialogue, and also the particular content of this move. The content of the move will depend on a variety of factors, all of which need to be considered by the player. Each combination of these factors leads to a different strategy. Firstly, the player should take into account the thesis of its arguments (the possible "*view*" or classification the player advocates), and the facts of the case under discussion. Then the player should consider the amount of data available in its repository, and the nature of this data (the domain from which this data is taken). The amount of information this player is willing to expose in one move is also important. Finally, the player's strategy should have space to consider the current state of the dialogue, and to forecast the future moves the adversary might make. In PADUA a player must select a single move to play in its turn. Moreover, every possible next move is associated with a set of possible rules that define the selection criteria for the move (desired confidence, premises and conclusion). Except for unwanted consequences, the other five speech acts included in PADUA introduce a new rule. However, the rules embodied in *Distinguishing* moves do not imply any conclusion. Instead they are intended to undercut the moves they attack. Proposing a counter rule leads to a switch in the players' roles, and thus changing the focus of the dialogue to this new rule.

Arguably, the notion of speech act and content selection in PADUA is best captured at different levels, as suggested by Moore (1993). Chapter 2 has drawn attention to how some argumentation systems have approached argument selection strategies. Here the three-layer structure suggested by Moore (1993) is adopted as a guideline for designing strategies suitable for the PADUA protocol. Recall that the structure Moore suggests has the following levels:

- **Level 1**: Maintaining the focus of the dispute.
- **Level 2**: Building its point of view or attacking the opponent's one.
- **Level 3**: Selecting an argument that fulfils the objectives set at the previous two levels.

The first two levels refer to the agent's strategy: the high level aims of the argumentation. The third level refers to the tactics: the means to achieve the aims fixed at the strategic levels. Moore's requirements form the basis of further research into agent argumentation strategies (e.g. (Oren et al, 2006), (Amgoud and Maudet, 2002) and (Yuan, 2004)). Amgoud and Maudet (2002) replace the first level of Moore's layered strategy with different profiles for the agents involved in the interaction. This approach is adopted here. In addition, another level is added to Moore's structure: level 0. This new level distinguishes PADUA games into two basic classes. In one class players attempt to win using as few steps as possible, so exposing the least amount of information to the opponent. Thus, the type and the content of each move are chosen so that the played move gives the opponent the least freedom to plan its next move. This mode is called: "*win*" mode. In the other class games are played to fully explore the characteristics of the underlying argumentation system and the dialogue game. Thus the type and the content of each move are chosen so that the played move will restrict the opponent's freedom to plan its next move, but this player will still have some space to counter attack, thus prolonging the dialogue game. This mode is called the "*dialogue*" mode. This layered strategy model is defined as follows (Wardeh et al., 2007b):

- **Level 0**: Defines the game mode: "*win*" mode or "*dialogue*" mode.
- **Level 1**: Defines the players (agents) profiles.

- **Level 2**: Defines the strategy mode: "*build"* mode or "*destroy"* mode. In the first, players aim to win the game by proposing new rules, thus building a strong argument. In the second, players try to win by "*destroying*" the adversary's argument. This is done by undermining these arguments either by distinguishing them or by pointing to their unwanted consequences.
- **Level 3**: Concerns choosing some appropriate arguments content depending on the tactics and heuristics suggested.

Amgoud and Parsons (2001) identify five classes of agents' profiles:

- *Agreeable Agent*: Accepts whenever possible.
- *Disagreeable Agent*: Only accepts when no reason not to.
- *Open-minded Agent*: Only challenges when necessary.
- *Argumentative Agent*: Challenges whenever possible.
- *Elephant Child Agent*: Questions whenever possible.

The last three profiles are suitable for rule-based argumentation system where "*challenge*" and "*question*" speech acts are available. For the purposes of PADUA, the first two profiles only will be considered (i.e. agreeable and disagreeable agents), as these attitudes are appropriate for the AEC scheme.

A Strategy function, which is called the *Play* function, can consequently be identified. PADUA players may use this function to select the moves to place next in dialogue games. For each player taking part in the PADUA dialogues, a∈ A, The function $Play_a$ is defined as follows (Wardeh et al., 2009a):

$$Play_a : M_{poss} \times R_{poss} \times D_{current} \times S_a \times Tactics_a \rightarrow M_{poss}$$

Where: $D_{current}$ is the current dialogue this player is taking part in (as identified in Section 3.4), thus $\mathrm{D}_{current}$ represents the set of moves played in the dialogue so far, and $M$ is the set of possible (legal) moves. $M_{poss} \subseteq M$ is the set of the possible moves this player can play. This set includes the possible legal moves that could be played as a response to the last move played in the game (as defined in Table 4.1). $R_{poss}$: is the set of legal rules that this agent can put forward in the dialogue. This set contains the rules that match each of the possible moves. $S_a$: is the

Strategy Matrix for this player, and has the form $S_a$ = [$gm_a$, $profile_a$, $sm_a$] where: $gm_a \in GM$: is the game mode, where $GM = \{win, dialogue\}$, $profile_a \in Profile$: is the player profile, where $Profile = \{agreeable, disagreeable\}$, and finally, $sm_a \in SM$: is the strategy mode, where $SM = \{build, destroy\}$. $Tactics_a$ is the tactics matrix including the move preference and the best move content tactics. These tactics are explained in detail in the following sub-section.

### 4.2.1. PADUA Tactics

A set of tactics to fulfil the strategic considerations discussed above can now be identified. These concern the best speech act to place next and, where applicable, the content of the move: the best AR to be used with the chosen speech act. Three different tactics will be considered: the first considers the best ordering of the legal moves. The second concerns the agent profile. The third is a combination of the previous two tactics.

**Tactic 1: Legal Moves Ordering**

This tactic identifies the order in which legal (possible) speech acts (moves) are considered when selecting the next move. In PADUA all games begin with *Propose Rule*: there are three possible responses to this, and these in turn have possible responses. The preference for these moves depends on whether the player is following a *build* or a *destroy* strategy. In a *destroy* strategy the player will attempt to discredit the rule proposed by its opponent, and hence will prefer moves such as unwanted consequences and distinguish. In contrast, when using a *build* strategy the player will prefer to propose its own rule, and will only attempt to discredit its adversary's rule if it has no better rule of its own to put forward. The preferred order for the two strategies is shown in Figure 4.2. Here the circles present the six promoted speech acts (moves) given the same numbers as Table 4.1. The links between the speech acts indicate the possible attack relations. The possible attacks on each speech act read from top to bottom, such that the one at the top is most desirable according to the particular strategy. For instance, if one player puts forward a new (counter) rule then if the other player has a build strategy, then it will to reply to this move by proposing a counter (new) rule corresponding to its advocated class. On the other hand, if this player has a destroy strategy, then it will attempt to distinguish the previous move first, and only if such attempt fails, the player will try to propose a counter (new) rule. Whether players are *agreeable* or *disagreeable* will have an influence on whether the agent would attempt to dispute the rule put forward by its rival, and the nature of the attack if one is made.

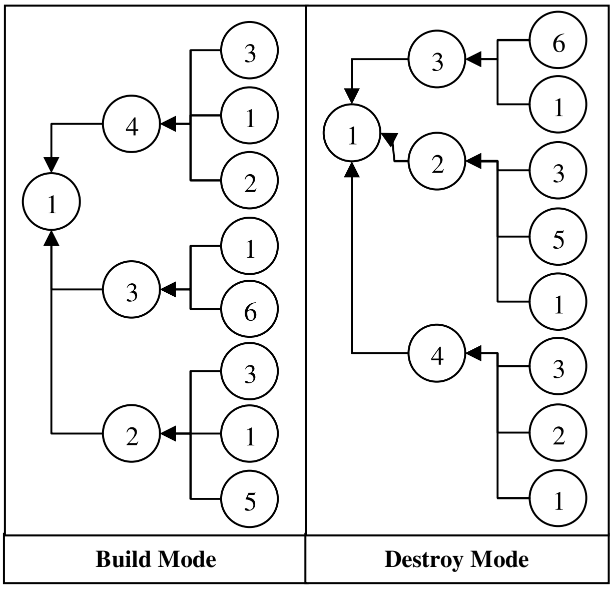


**Figure 4.2. Legal moves ordering for PADUA (Wardeh et al., 2007b).**

### Tactic 2: Agent Profile – Agreeable or Disagreeable

The Agent Profile tactic articulates how each profile affects a player's criteria for attacking its adversary. Agreeable players tend to be less aggressive. If an agreement with the other players is possible, then there is no need to challenge their propositions. Disagreeable players, on the other hand, will insist on challenging their rivals, even if their proposed rule would be acceptable according to their own data. Agreement will not be conceded as long as there is room to manoeuvre. In the following the two profiles used in PADUA are described.

*Agreeable Players (Agents):*

- An agreeable player *ap∈ A* accepts a played rule without attacking it if:
  - An exact match of this rule can be mined from the player's dataset ($\Sigma_{ap}$) with a higher or similar confidence.
  - A partial match of this rule can be mined from the player's dataset ($\Sigma_{ap}$). A rule $\mathrm{r}_{\mathrm{pm}} \in\Sigma_{ap}$ is considered a partial match of another rule $r\in\Sigma_{ap}$ if it has the same consequences of *r*, its set of premises is a superset of rule *r* premises such that all these premises match the case, and finally it has a higher or similar confidence.
- Otherwise the agreeable agent will attempt to attack the played move, according to its underlying strategy mode (build or destroy) using the legal moves preferences shown in Figure 4.2, and selecting a rule using the following content tactics:
  - *Confidence* of moves played by agreeable agent should be considerably lower/higher than the attacked rule, otherwise it agrees with its rival.
  - *Consequences* always contain a class attribute. The agent should make minimum changes to previous move consequences, which should contain as few attributes as possible.
  - *Premises* are always true of the case. The agent should make minimum changes to previous move consequences, which should contain as few attributes as possible.

*Disagreeable Players*

- A disagreeable agent accepts a played rule if and only if all possible attacks fail, and so does not even consider whether its data supports the rule. The choice of the attack (i.e. legal move) to be played depends on the preferences shown in Figure 4.1 and the choice of rule is in accordance with the following content tactics:
  - *Confidence* of moves played can be either *considerably* or *slightly* different from the last move. The choice of confidence depends on the player's game mode: whether it is *win* or *dialogue* mode.

- *Consequences* always contain a class attribute. The agent would attempt to use as few attributes as possible.
- *Premises* are always true of the case. The agent would attempt to use as few attributes as possible.

### Tactic 3: Best Move

The Best Move tactic is a combination of the above two categories of tactic. Table 4.2 brings together the considerations discussed above, and shows the best move relative to the agent's profile and game mode, for each of the six possible speech acts. For example in *win* mode an agent will want to propose a rule with high confidence, as one which the adversary is likely to be forced to accept, whereas in *dialogue* mode, where a more thorough exploration of the search space is sought, any acceptable rule can be used to stimulate discussion. The Best Move tactic thus advocates selecting the most appropriate move as illustrated in Table 4.2.

| **Best Move** | **Agreeable** | | **Disagreeable** | |
|---|---|---|---|---|
| | **Win mode** | **Dialogue mode** | **Win mode** | **Dialogue mode** |
| **Propose Rule** | High confidence | Average confidence | High confidence | Moderate confidence |
| | Fewest attributes | Fewest attributes | Moderate attributes | Fewest attributes |
| **Distinguish** | Lowest confidence | Moderate drop | Lowest confidence | Moderate drop |
| | Fewest attributes | Fewest attributes | Fewest attributes | Fewest attributes |
| **Unwanted Consequences** | If some consequences are not in or contradict the case | Only if some consequences contradict the case. | If some consequences are not in or contradict the case | |
| **Counter Rule** | Moderate confidence | High confidence | High confidence | Moderate confidence |
| | Fewest attributes | Fewest attributes | Moderate attributes | Fewest attributes |
| **Increase Confidence** | Highest confidence | Moderate increase | Highest confidence | Moderate increase |
| | Fewest attributes | Fewest attributes | Fewest attributes | Fewest attributes |
| **Withdraw Unwanted Consequences** | The preferable reply to unwanted consequences attack → selecting criteria is the same of the very last move that led to the unwanted consequences. | | | |

**Table 4.2. Best move content tactics (Wardeh et al., 2007b).**

### 4.2.2. Discussion of some Example Strategies

The different types of strategies applicable under the PADUA protocol are now discussed. These strategies will be illustrated using a number of example dialogues produced by the PADUA GUI application discussed in Section 4.1. These example dialogues are drawn from the same configuration as applied in PADUA Dialogue Example (Sub-section 4.1.2). This sub-section demonstrates that by changing the strategy the nature of the dialogue changes drastically. The strategy used in PADUA Dialogue Example was arbitrarily chosen such that both players apply *disagreeable* profiles, and a *win* game mode. The only difference was that the proponent applied a *destroy* strategy while the opponent applied a *build* one. In the following four examples this strategy was slightly changed, and each example will be discussed in terms of how these changes affect the dialogue game.

#### Strategy Example 1

Assume that both players apply *agreeable* instead of *disagreeable* agent profiles. Here the proponent starts the dialogue proposing the same rule as in PADUA Dialogue Example. But the opponent, instead of challenging this rule, simply agrees with the proponent:

```
Opponent (not entitled) - Agrees with the Proponent,
because the opponent was able to find an exact match of
the rule suggested by the Proponent: (Age = 75<age<80,
Residency = UK, Income<15% and 2000£<Capital<3000£ →
entitled). With confidence = 89.05%.
```

The consequent dialogue game is shorter than PADUA Dialogue Example as the opponent agreed with the proponent at a very early stage at the game. Such agreement may take place at a later stage in other dialogues, or may not take place at all. In this worst case it does not matter if the player applies an agreeable or a disagreeable profile. The result of this dialogue game is similar to the one in PADUA Dialogue Example, and the case under discussion is classified as entitled to housing benefit.

### Strategy Example 2

The strategy configuration of Strategy Example 1 can be changed such that both players apply *disagreeable* profiles, and a *win* game mode, and both apply a *build* strategy. The dialogue produced under this configuration is different from the ones produced in PADUA Dialogue Example and in Strategy Example 1. Here, the dialogue shares the first two steps with Example 1. But at the third round, the proponent, instead of distinguishing the opponent's position, proposes a new rule as follows:

```
Proponent: (entitled) - Proposes a new Rule: The case
has the following features: Age = age<80, Residency =
UK, contribution Y1 = paid, contribution Y2 = paid and
Contribution Y4 = paid. Therefore this case should be
classified as (entitled). With confidence = 97.84%.
```

The dialogue terminates at this stage. Of note here is that the proponent has proposed the same rule it proposed at the fifth round in PADUA Dialogue Example, in the third round of this dialogue. This is because this move matches the proponent strategy in this example, whereas in PADUA Dialogue Example the proponent, following a destroy strategy, was forced to wait until the fifth round to play this move after failing to distinguish the opponent moves.

### Strategy Example 3

The strategy configuration of PADUA Dialogue Example can be alternatively changed such that both players apply *disagreeable* profiles, and a *win* game mode. However the proponent can apply a *build* strategy while the opponent applies a *destroy* strategy. Once more, the proponent starts the dialogue in the same manner as the previous examples. However, at the next round the opponent, following their strategy, distinguishes this move instead of proposing a new rule, as follows:

> Opponent (not entitled) - Distinguishes the previous rule: The case has the following additional features: Gender = male and contribution Y3 = not paid. Therefore my confidence in this case being of class (entitled) is no more than **19.74%** only.

The proponent then replies to this attack by proposing the same winning rule from the previous examples. It is worth noting that the proponent has again played this rule earlier in the game because it matches its strategy.

### Strategy Example 4

All the examples displayed so far, assume that both sides apply a *win* game mode. However, both players can make use of *dialogue* mode strategies. In addition, the proponent can apply a *build* strategy while the opponent applies a *destroy* strategy. Using these strategies, the proponent opens the dialogue with proposing a new rule as follows:

> Proponent (entitled) - Proposes a New Rule: The case has the following features: Age = 75<age<80 and Residency=UK Therefore this case should be classified as (entitled). **With confidence = 70.51%.**

This move differs from the opening move the proponent used in the previous dialogues. This is because the proponent has looked for a rule whose confidence is not much higher than the acceptable level. At the next round the opponent replies to this move as follows:

> Opponent (not entitled) - Distinguishes the previous rule: The case has the following additional feature: Gender = male. Therefore my confidence in this case being of class (entitled) is no more than **42.34 %**only.

Here also the opponent distinguished the proponent proposition by adding as few additional features as possible to this proposition. Of note here, the drop in confidence is not as severe as has been the case in the previous examples. The dialogue continues in this manner until the eleventh round when the proponent finally proposes the winning rule from the previous examples. Note that this dialogue is much longer and explores more possibilities than the games where the players were attempting to force a quick win.

### 4.2.3. Discussion

It is clear from the above analysis, and the accompanying examples, that different strategies have different consequences on the structure of the dialogue game. Participants may choose to be cooperative or adversarial. Thus the resulting dialogues may have the flavour of persuasion or of deliberation. This is determined by the combination of strategies and tactics of both players. Dialogues between "*disagreeable"* players will have a persuasion flavour. If the opponent is applying a destroy strategy then the dialogue will be akin to "*dissents*". On the other hand, if both players apply build strategies then the result will be a "*dispute*" dialogue. A different dialogue type may be achieved by adopting "*agreeable*" profiles. When both participants are agreeable the result will be a deliberation dialogue. In this case players are not committed to proving a particular thesis. Rather they share the same objective - reaching a mutually acceptable classification, whether it is the one they are advocating or not. They, therefore, both aim to come up with the best possible classification for the case under discussion. A "*grey area*" exists in the middle between persuasion and deliberation dialogues. If one participant has an agreeable profile while the other has a disagreeable one, then the consequent dialogue will be that of persuasive flavour with a hint of deliberation, with one participant attacking all the time while the other player trying to avoid the attacks[17]. Table 4.3 provides a summary of the discussion included in this sub-section.

Game mode tactics do not have an immediate effect on the dialogue type. Rather, they offer a chance to explore the features of PADUA dialogues in more detail. "*Win*" and "*dialogue*" mode tactics lead, most of the time, to the same

[17] Although presented as distinct dialogue types in (Walton and Krabbe, 1995), other discussions such as (Walton, 2009) use examples in which the distinction is less clear cut, and even becomes blurred at times. Therefore, no difficulty is seen in regarding PADUA dialogues as persuasion dialogues (the existence of a winner, even where the competition is not fierce, gives them this form) with varying degrees of similarity to deliberation according to different strategies.

end result. The difference is that when both players apply win mode tactics, they will reach a resolution more swiftly than if they have applied "*dialogue*" mode tactics. Thus *dialogue* game mode tactics are important to seeking make the most of each other's different experiences in the course of the dialogue. *Win* game mode tactics, on the other hand, are more desirable in scenarios where the concern is not the dialogue itself, but rather the result of the dialogue. Another noteworthy point is the effect different strategies have on the end result of the dialogues governed by PADUA. This point was not covered by the examples discussed above. But in some cases, altering the strategies may lead to a significant divergence in the course of a dialogue game. Consequently, the output of this game could change from one possible "*view*" (classification) to another, for instance, if a player has an acceptable association for one classification but a better association supporting the contrary. This point will be explored in details in Chapter 5, where the effect of applying different strategies on the operation of PADUA is investigated.

| **Proponent's Strategy** | | **Opponent's Strategy** | | **Resulting Dialogue** |
|---|---|---|---|---|
| **Profile** | **Mode** | **Profile** | **Mode** | |
| Disagreeable | Build | Disagreeable | Build | **Disputes (persuasion)** |
| Disagreeable | Build | Disagreeable | Destroy | **Dissents (persuasion)** |
| Disagreeable | Destroy | Disagreeable | Build | |
| Disagreeable | Destroy | Disagreeable | Destroy | |
| Agreeable | ANY | Agreeable | ANY | **Deliberation** |
| Agreeable | ANY | Disagreeable | ANY | **Mixed Dialogue (persuasion with hint of deliberation).** |
| Disagreeable | ANY | Agreeable | ANY | |

**Table 4.3. A summary of the types of dialogues supported by PADUA.**

## 4.3. On Accrual of Arguments in PADUA

The tactics discussed above do not consider, when selecting moves to place in the dialogue game, the structure of the rules forming the content of these moves. Rather, they acknowledge the confidence of each rule and whether or not the rule matches the criteria for the underlying speech act. A new set of optional tactics will be presented in this section to attend to the particulars of the ARs associated with moves placed in PADUA's dialogue games. These tactics are integrated into the decision making process of each player as follows. Upon applying the best content tactic (tactic 3), the player will have the option to further filter the selected rules. Rules with more than one attribute in their antecedents or consequences are considered accruals of arguments. Therefore these rules should satisfy certain conditions if they are to be included in the best content tactic. Prakken (2005a) suggests three basic principles that accrual formalisations should consider. These principles will now be discussed in relation to the decision making procedure outlined above, before presenting how this procedure is translated into a set of tactics that can be applied along with the other tactics identified in the previous section.

The first principle (Principle 1) indicates that "*accruals are sometimes weaker than their elements*"**.** When combining multiple arguments together it is not always the case that the resulting arguments would be stronger than the individual sub-arguments. This is because there is a possibility that these sub-arguments are not independent. Principle1 applies also to the arguments instantiated from AEC2 as integrated into PADUA. For example, the following three rules may have been generated by a player in a PADUA dialogue game:

*$R_1$: X→Q with confidence $c_1$ = 50%.*
*$R_2$: Y→Q with confidence $c_2$ = 70%.*
*$R_3$: XY→Q with confidence $c_3$ = 20%.*

Note that although features X and Q appear together in 50% of the instances in which the set X appears, and features Y and Q appear together in 70% of the instances in which the set Y appears, yet Q appears only in 20% of the instances which contains both X and Y. Such scenario may take place in variety of domains. In the shopping basket example, we may observe that 50% of customers who bought chicken bought bread, and 70% of customers who bought cheese bought bread, yet only 20% of customers who bought both cheese and chicken bought bread.

The second principle (Principle 2) in Prakken's set states that: "*An accrual makes its elements inapplicable*". Thus, any larger accrual makes all its lesser versions irrelevant. This principle is relevant to the "*Arguing from Experience*" because accruals consider more "*information*" from the case under discussion than any of its individual sub-arguments. Therefore once an accrual is placed in a dialogue it makes all its sub-arguments illegal in the subsequent flow of this dialogue. For example, in a PADUA dialogue between two players P and O, assume that P can generate the following rules form its set of past examples:

$R: XY \rightarrow w$ *with confidence* $c = 60\%$

$R_1: X \rightarrow w$ *with confidence* $c_1 = 80\%$

$R_2: Y \rightarrow w$ *with confidence* $c_2 = 50\%$

While O may generate the following rule:

$R_4: Z \rightarrow \neg w$ *with confidence* $c` = 70\%$

If P opened the game with R as its initial argument, then O would attack this argument by the counter argument $R_4$. According to principle2 P would not be able to play any of the other ARs ($R_1$ or $R_2$) and thus loses the game to player O.

The last principle in Prakken's account (Principle 3) indicates that: "*Flawed reasons or arguments may not accrue*". Thus all the sub-arguments in an accrual should be valid arguments. In PADUA flawed arguments may be identified as ARs whose confidence is below the acceptable level. According to this definition the third principle is not binding when forming accruals of arguments in PADUA. Rather, players may apply this principle as a tactic to achieve "*stronger*" accruals, or to attack the "*weakest link*" in accruals proposed by the other side. The above three principles are embodied in PADUA by the means of two tactics that may be used with any of the strategies and tactics discussed in the previous section. The first tactic seeks an answer to whether a player is better off placing a single "*Argument from Experience*" or an accrual of these arguments. The second tactic concerns the last principle (Principle 3).

### Accrual Tactic 1: Playing a single argument or an accrual of related arguments:

A player may consider playing an accrual of "*Arguments from Experience*", if an AR satisfying the following two conditions could be mined from the set of this player past experience. The first condition concerns the speech act for which the rule is mined:

- The content of "*Propose Rule*" or "*Counter Rule*" speech acts can be either a single argument or an accrual.
- The content of "*Increase Confidence*" speech acts is an accrual of arguments, as new attributes (arguments) are added to the initial argument to increase its confidence.
- The content of "*Distinguish*" speech act is a counter accrual in which more arguments are gathered against a certain conclusion.

The second condition examines Principle 1 and Principle 2 identified above as follows:

- Propose Rule or Counter Rule: Let $R: z \rightarrow Q$ (with confidence $C_R$) be a single argument, let $R`: Z \rightarrow Q$ (with confidence $C_{R`}$) be an accrual of arguments. If $\exists z \in Z$ such that $C_{R`} > C_R$ then play $z \rightarrow Q$.
- Increase Confidence: Let $R: X \rightarrow Q$ with confidence $C_R$ is the previously played argument, let $R`: XY \rightarrow Q$ with confidence $C_R`$ is the output of the increase confidence step such that $C_R > C_{R`}$. Now the player puts $XY \rightarrow Q$ forward if and only if there is no $y \in Y$ such that the confidence of $(y \rightarrow Q) > CR`$, otherwise play $y \rightarrow Q$.

### Accrual Tactic 2: The weakest link attack

This attack is related to Principle 3 regarding flawed arguments. In PADUA some conflicts may take place if the same rule (argument) was mined by both sides with different confidence values. This means that an argument which is totally valid from one player's perspective can be considered flawed from the perspective of the other. Thus, a player can attack its rivals moves based on its own confidence in these moves. This attack is formally explained as follows. Let $P$ and $O$ be two players. If P played an accrual $X_1 \ldots X_n \rightarrow Q$ (with confidence c), then if O could formulate an argument of the form $X_i \rightarrow Q$ (with confidence $c_i$) such that $Xi \in X$, and $c_i$ is lower than the minimum confidence (the argument is invalid) then $O$ can play this argument (rule) as a weak link attack. This tactic, while interesting, requires additional speech act to be incorporated in the dialogue model presented in Section 3.2. Therefore, the implemented PADUA application does not take this attack into consideration. However, seen as an appealing extension of the promoted dialogue model, Chapter 9 will return to this attack when suggesting direction for future work.

## 4.4. Arguing about Intermediate Concepts (Classifications)

One classic example of the PADUA reasoning and application model is found in common law, especially as practiced in the US, where arguments about a case are typically backed by precedents. Even where decisions on past cases are encapsulated in a rule, the *ratio decendi*, the particular facts are still considered and play crucial roles in the argument. An important topic of discussion in recent work on reasoning with legal precedent is the significance of intermediate concepts (e.g. (Ashley and Brüninghaus, 2003), (Atkinson and Bench-Capon, 2005) and (Lindahl and Odelstad, 2005)). Lindahl and Odelstad (2005) make an important distinction between intermediate predicates which are functionally determined by some base level predicates, and those for which there is no simple truth functional relationship: where there are a number of considerations, but no way of combining these to form necessary or sufficient conditions. For this latter kind of intermediate predicates, it may be necessary to first agree their application before deciding the main question. This point is analogous to the difficulty in classifying examples of *XOR* using a single layer perceptron proposed by Minsky and Papert (1969). No simple classification rule for *XOR* over two variables can be produced using only the truth functions of the inputs. Rather the intermediate classifications "*and*" and "*or*" must be produced and then final classification conducted in terms of these. So too, with law: some features used in classifying cases are not simple facts of the case, but rather classifications of the applicability of intermediate concepts on the basis of a subset of the facts of the case. "*Arguing from Experience*" dialogues must therefore be able to accommodate a degree of nesting, where first the satisfaction of intermediate concepts is agreed, and then used in the main debate. This concept of intermediate concepts (classifications) is accommodated in PADUA through the possibility of "*nested"* dialogues. PADUA allows for dialogues to be nested so that a number of secondary dialogues may take place to solve disputes over some intermediate classifications, before arguing over the main issue (classification) can take place. Note that this form of "*nesting*" differs from the one identified in (Walton and Krabbe, 1995), by which resolving certain conflicts may require leaving the persuasion dialogue to enter a dialogue of a different type. As in PADUA, the players leave the main dialogue to enter a secondary dialogue of the same type to debate intermediate concepts. To realise this view of nested dialogues, a *Control Layer* was incorporated into the PADUA system. This *Control Layer* is intended to manage the arrangements of the main and secondary dialogues. This layer also facilitates the communication among the participants of every dialogue, to cover the cases in which some players are engaged only in some "*nested*" dialogues, and not in all of them. The implementation of PADUA Control Layer has been kept as simple as possible, mainly because dialogues taking place in the PADUA system are of a persuasive nature, and follow the "*Arguing from Experience*" dialogue model discussed in the previous chapter.

The formalisation of the PADUA control layer is defined in the terms of the following components:

- The set of agents (players) $A = \{Proponent, Opponent\}$ as identified in Section 4.1. Such that the set of classes each player tries to prove true contains the main class and any intermediate classes argued about in the nested dialogues.
- *Gs*: set of PADUA secondary dialogue games.
- *gm*: the PADUA main dialogue game.

- *start*: a function that begins a certain PADUA dialogue game, start($gs \in Gs$) begins a secondary dialogue game, while *start*(*gm*) begins the main dialogue.

Note that the PADUA control layer requires some degree of domain analysis to identify and organise the intermediate predicates, so as to form what is termed in IBP by Ashley and Brüninghaus (2003) a "*logical model*" of the domain. This analysis is at a high level and, as in IBP, does not require the consideration of individual cases. Once identified, this "*logical model*" can be used by the control layer of PADUA to set the agenda for the dialogue.

A short example is now provided to clarify the above definitions and to illustrate the improvements gained when applying nested dialogues to the housing benefit welfare domain discussed in Section 4.1.3. Recall that the major problem with benefits such as the above is that they are often adjudicated by a number of different offices and exhibit a high error rate due to various misunderstandings of the legalisation and how it should be interpreted. This yields large data sets which contain a significant number of misclassifications, the nature of which varies from office to office. To test how PADUA can cope with this situation, artificial RPHA benefits datasets (each comprises 12,000 records) were generated to mimic different systematic misapplications of the rules. For example that one does not consider the exceptions to the residency condition (i.e. only UK residents are considered valid candidates for the benefits), while another interprets the "*established connection with the UK labour force*" as having paid contributions in 3 of the last 6 years rather than 5. The purpose of this test was to find out whether the proposed dialogue game helps in correctly classifying examples and henceforth correctly interprets them, even when the two agents are depending on (completely or partially) wrongly classified examples. This could provide a way to facilitate the sharing of best practice between offices. Each dataset was assigned to a PADUA player, corresponding ARs were mined from these sets (as necessary) using a 70% confidence threshold for both players, and PADUA was applied to different sets of examples each of which focuses on an exception of one of the six conditions mentioned above.

Unfortunately when *n* out of *m* attributes are needed to decide whether a condition is satisfied or not, like the contribution years in our example, it is not always the case that the classification process will run correctly. More reliable results can be achieved by applying an intermediate nested dialogue over the contribution years factor, which gives as a result the status of the contribution condition (true or false) before a main dialogue takes place over the eligibility of the applicant. For example, take the case of a male applicant that satisfies all the conditions except for the contribution condition as he paid only the contribution fees of the third, fourth and the sixth years, and apply the "one-dialogue" PADUA to this case between the same proponent and opponent as in the last example (also applying the same strategies and tactics), the proponent fails to correctly classify the candidate status even after a very exhaustive 30 step dialogue in which each contribution year is considered as independent factors, as can be shown by some of the rules played in the dialogue[18]:

> Proponent (not entitled) – Proposes a New Rule (**R1**): The case has the following feature: Contribution Y5 = not paid. Therefore this case should be classified as (not entitled). **With confidence = 73.14%.**

> Proponent (not entitled) – Proposes a New Rule (**R23**): The case has the following features: Gender=male, and Contribution Y2= not paid. Therefore this case should be classified as (not entitled). **With confidence = 87.69%.**

> Proponent (not entitled) – Proposes a New Rule (**R29**): The case has the following features: residence=UK, Contribution Y1= not paid, Contribution Y2= not paid and Contribution Y5= not paid. Therefore this case should be classified as (not entitled). **With confidence = 95.31%. If it has the additional attribute Age>=65 years.**

None of these can gain acceptance from the dataset used by the opponent. The opponent can play a rule such as:

[18] This example was previously publish in (Wardeh et al., 2008b, 2009a)

```
Opponent (entitled) - Proposes a New Rule (R30): The
case has the following features: Age>=65, Residence=UK,
Contribution Y3= paid, Income<20% and
2000£<Capital<3000£. Therefore this case should be
classified as (entitled). With confidence = 96.82%.
```

The latter rule is in fact the final move in the dialogue, as the proponent fails to defeat it using any of the valid attacks. This shows the inability of the proponent to force acceptance of any of its proposed rules, which means that a mistake is made. Table 4.4 shows how, by applying two dialogues (nested and main) to the same case using the same individual datasets, so that the contributions issue can be settled separately, the proponent becomes able to win the game: by winning the nested dialogue over contribution years first, then applying the result of that dialogue to the main dialogue. Of course, to apply this method, first the intermediate concepts, which require this special treatment, must be identified, and so at least some of the sort of analysis found in the works of Aleven (1997) and Ashley (1990) must be performed. This further strengthens the argument made by Governatori and Stranieri (2001) and in other works such as (Atkinson and Bench-Capon, 2005) which stress the crucial role of intermediate concepts.

## 4.5. Summary

This chapter has taken the theory of "*Arguing from Experience*" articulated in Chapter 3 and transformed this into a dialogue game protocol called the *PADUA protocol*. This protocol enables dialogue games between two players, each representing one side of a conflict over the classification of some case in some domain. The result of dialogue games under PADUA is a proposed classification of the case under discussion. Details were given of how this protocol was realised as the *PADUA GUI Application*. This implemented system was used to investigate the nature of the resulting dialogues. This chapter also discussed PADUA strategies and tactics. A four layer strategy model based on that proposed by Moore (1993) was presented. This chapter ended by considering the issue of accrual of arguments in the PADUA protocol, and the problem of intermediate classifications. In the case of the latter it was shown how the performance of PADUA can be enhanced by allowing for nested dialogues to take place over intermediate precedents.

| Nested Dialogue | Main Dialogue |
| --- | --- |
| Round 1: Proponent (Contr Not Paid) - Proposes a New Rule: The case has the following features: contr year1 = not paid and contr year5 = not paid. Therefore this case should be classified as (not paid). With confidence = 74.71%.<br>Round 2: Opponent (Contr Paid) - Distinguishes the previous rule: The case has the following additional feature: contr year3 = paid. Therefore my confidence in this case being of class (not paid) is no more than 30.00% only.<br>Round 3: Proponent (Contr Not Paid) - Increases the confidence of a previous rule: The case has the additional feature: contr year1 = not paid. Therefore this case should be classified as (not paid). With confidence = 100.00%.<br>Round 4: Opponent (Contr Paid) - Distinguishes the previous rule: The case has the following additional feature: contr year6 = paid. Therefore my confidence in this case being of class (not paid) is no more than 30.00% only.<br>Round 5: Proponent (Contr Not Paid) - Increases the confidence of a previous rule: The case has the additional feature: contr year4 = not paid. Therefore this case should be classified as (not paid). With confidence = 100%. | Round 1: Proponent (not entitled) - Proposes a New Rule: The case has the following feature: Contribution NOT PAID. Therefore this case should be classified as (not entitled). With confidence = 94.00%. If it has the additional attribute age>65.<br>Round 2: Opponent (entitled) - Distinguishes the previous rule: The case has the following additional features: gender = male and 2500£<capital<3000£. Therefore my confidence in this case being of class (not entitled) is no more than 18.84% only.<br>Round 3: Proponent (not entitled) - States that the previous rule has some Unwanted Consequences (2500<capital<3000). |
| The opponent fails to counter the proponent attack, and the game ends in favour of the proponent. | The opponent fails to counter the proponent attack, and the game ends in favour of the proponent. |

**Table 4.4. The results of applying the nested dialogue (Wardeh et al., 2009a, 2008b).**

The next chapter will examine the issues addressed in this chapter by means of a sequence of empirical experiments designed to provide evidence regarding the operation of PADUA and the nature of the consequent dialogues. The results obtained from these experiments will affirm that PADUA can successfully predict class values for cases from different two-class domains, with accuracies comparable to other conventional classifiers. The included experiments will also be used to evaluate the various features of the PADUA protocol, such as strategies and nested dialogues.

# Chapter 5: Empirical Observations (1) - Analysis of the Features of the PADUA Protocol

The previous chapter provided a description of the PADUA protocol and the style of dialogues it produces. This chapter complements the discussion of the previous chapter with a report of a set of empirical experiments designed to evaluate the operation of PADUA in terms of the accuracy of the dialogues produced. The analysis presented here will provide evidence that PADUA can successfully facilitate two-party "*Arguing from Experience*". Also it will be shown that PADUA can effectively classify cases from two-class datasets, by means of argumentation, while at the same time producing an explanation (in the form of a dialogue transcript) as to how each case was classified. The results of the reported experiments will show that PADUA is particularly applicable to domains in which there are large volumes of data available and where it would prove unrealistic to hand craft a knowledge base. PADUA can thus complement rule based protocols, since its performance is actually enhanced by large volumes of data; whereas, for example, the work of Chorley and Bench-Capon (2005), which used dialogue to generate rule based theory, can only be applied to comparatively small datasets. Also, the work suggested by Governatori and Stranieri (2001) to generate defeasible and strict rules using ARM techniques is limited to small datasets. Other restrictions are also forced on the datasets used in this work such as that they should have no missing values and that all values are correctly recorded. PADUA on the other hand is applicable to misinterpreted or noisy data, as will be emphasised throughout this chapter.

Most of the experiments discussed here were carried out using the same setup. Section 5.1 provides details of this setup. Sections 5.2 to 5.6 discuss the results of the various experiments implemented to examine the various aspects of PADUA. Section 5.7 concludes with a summary. Four categories of experiments were defined:

1. **The operation of PADUA** as means to facilitate two-party "*Arguing from Experience*" and the nature of the resulting dialogues. Section 5.2 provides analysis of a number of experiments intended to assess this operation.
2. **The operation of PADUA as classifier** was seen as important by-product of the promoted model. Section 5.3 provides a comparative analysis of the application of PADUA to a number of classification problems.
3. **PADUA robustness to noise**, seen as important if PADUA is to be applied in real-world domains was thoroughly evaluated. Different types of noise were investigated: (i) random class noise and missing attributes. Section 5.4 provides an analysis of a number of experiments designed to address these types of noise. (ii) "*Systematic errors*" that occur because of some underlying misconception or misunderstanding rather than data loss or miscommunication. Section 5.5 provides analysis of applying PADUA with datasets infected with these errors.
4. **The effects of applying different strategies** on the characteristics of the underlying dialogues. Section 5.6 provides a summary of experiments intended to evaluate the different possible strategies in PADUA.

## 5.1. Experimental Design

This section describes the background to the evaluation described in this chapter. The section is divided into four sub-sections: (i) review of the data sets used, (ii) review of the comparator classifiers, (iii) review of the evaluation measures used, and (iv) description of the advocated methodology.

### 5.1.1. Datasets

A number of real-world and artificial datasets were used for the evaluation. The latter were utilised because they provided fertile settings to examine the particular features of PADUA. The datasets used are summarised below, Table 5.1 provides an overview of the features of these datasets:

1. **Three two-class real world datasets**, all were drawn from the UCI repository (Blake and Merz, 1998). For the purposes of testing PADUA a discretised version of each of these sets was used. The discretised datasets were obtained by anonymous download from (Coenen, 2003).
2. **Two Artificial datasets generators** were implemented for the purposes of testing PADUA. Each generator produced datasets expressing different fictional benefit scenario:
   - A Retired Persons Housing Allowance (RPHA) discussed in Chapter 4.
   - A welfare benefit scenario originally developed by Bench-Capon (1993), and had been used in several experiments such as the one conducted in (Bench-Capon and Coenen, 2000), (Mozina et al, 2005) and (Johnston and Governatori, 2003).

| Domain | Exs# | Atts# | Missing | Classes | Best published accuracy (UCI) |
|---|---|---|---|---|---|
| **Mushroom** | 8124 | 22 | 1.4% | 51.8% edible, 48.2% poisonous | 100% (Kim and Park, 2004) SVM. |
| **Congressional Voting Records** | 435 | 16 | 10.7% | 45.2% democrats, 54.8% republican | 89% (Gionis et al., 2007). Cluster aggregation. |
| **Pima (Diabetes)** | 768 | 8 | 0 | 65.1% no, 34.9% yes. | 76.6% (Eggermont et al., ) Bagged C4.5. |
| **Housing Benefits** | 2400 | 11 | 0 | 50% entitled<br>50% not entitled | |
| | 24000 | | | | |
| **Welfare benefit** | 2400 | 13 | 0 | | |
| | 24000 | | | | |

**Table 5.1. Datasets used with PADUA.** *The columns are, in order: name of the domain; number of examples; number of attributes; the percentage of missing value and the classes' distribution***.** *Last column shows the best published accuracy according to the UCI Machine Learning repository (Blake and Merz, 1998).*

The second scenario concerns a fictional welfare benefit paid to pensioners to compensate expenses for visiting a spouse in hospital. The benefit is payable if six conditions are satisfied. These conditions resemble the ones associated with the RPHA benefits. However, RPHA employs a more flexible contribution and residency conditions: in the welfare benefit the applicant should have paid contributions in four out of the last five relevant contribution years (instead of three). Also the applicant should be resident in UK and not absent from it. An additional condition features in the Welfare benefit scenario, such that: "*If the patient is an in-patient the hospital should be within a certain distance: if an out-patient, beyond that distance*." The wide range of conditions covered by both scenarios is one of the reasons why they were selected to evaluate PADUA. Two datasets following each scenario were generated for the purposes of testing PADUA and examining its features, such that one of the two is much larger (in terms of number of records) than the other. Thus the four sets provided a means to examine the performance of PADUA with large and medium sized datasets. Both scenarios are applied in terms of two classes: "*Entitled*" to the benefit or "*Not Entitled*". The datasets were produced such that they contain equal numbers of cases falling under each class.

### 5.1.2. Comparator Classifiers

The performance of PADUA, as a classifier, was compared to a total of eight alternative classification algorithms so as to cover a wide range of paradigms[19]:

- *Decision Trees*: Two variations of the C4.5 algorithm (Quinlan, 1993) were applied. The distinction between the two of them is in the *splitting criteria*:
  - Random Decision Tree (RDT) selects the most frequent item.
  - Information Gain Decision Tree (IGDT) applies the Information gain measure as suggested by Quinlan (1987).
- *CARM Algorithms*: These apply similar techniques to the underlying rule mining mechanism embodied by PADUA. Comparing PADUA to these algorithms was therefore seen appropriate. The following were used:
  - CBA - Classification Based on Associations (Liu et al, 1998).
  - CMAR - Classification based on Multiple ARs (Li et al, 2001).
  - TFPC - Total From Partial Classification (e.g. (Coenen et al, 2005)).
- *Inductive learning algorithms* for generating CARs:
  - CPAR - Classification based on Predictive ARs (Yin and Han, 2003).

[19] For full details of these classifiers the reader can refer to Chapter 2.

- PRM - Predictive Rule Mining (Yin and Han, 2003).
- FOIL - First Order Inductive Learner (Quinlan and Cameron-Jones, 1993) as implemented in (Coenen 2004a).

### 5.1.3. Evaluation Measurements

Four measurements were applied to assess the operation of PADUA:

- *Classification accuracy* calculated as the number of cases PADUA (and the other identified classifiers) correctly predicted when applied to each of the datasets discussed above, where "*Ten-fold Cross Validation*" (TCV) tests were applied to calculate this accuracy.
- *Average Accuracy* across all the included datasets was also calculated (for PADUA and the other classifiers) in some of the included experiments. The significance of this measure is debatable but it has often been used (e.g. (Quinlan, 1993) (Clark and Boswell, 1991)).
- *The length of the underlying dialogues* calculated as the average number of rounds PADUA requires to come to a decision with respect to cases from each of the identified datasets. This measurement provides evidence to the soundness of PADUA dialogues.
- *The McNemar's test* was used to uncover the *significant differences* in the operation of PADUA and the included classifiers, or when applying PADUA using different settings. The use of this test was recommended in a number of papers (e.g. (Salzberg, 1997) and (Aleven, 2003)).

The McNemar's test is essentially a "*sign test*" designed to explore the hypothesis that one classifier is *significantly* better than another, by comparing the number of cases on which one classifier does better against those on which the other classifier does better[20]. In this chapter, the results of applying PADUA with 100 cases were compared to the results from one classifier at a time using the same cases. The difference in the operation of the two approaches was

[20] McNemar($\chi^2$) = $\frac{(|number\ of\ cases\ in\ which\ PADUA\ failed - number\ of\ cases\ the\ other\ classifier\ failed| - 1)^2}{|number\ of\ cases\ in\ which\ PADUA\ failed + number\ of\ cases\ the\ other\ classifier\ failed|}$

considered significant if the P-value associated with the test was less than 0.05[21]. The *lower* the *P-value*, the *more* "*significant*" the differences are between the two classifiers.

### 5.1.4. Methodology

The evaluation methodology used for the experimentations described in this chapter was as follows. First each dataset was divided equally among two PADUA players, such that each one got a random half of the dataset under consideration. Then a number of PADUA dialogue games were conducted between these two players to classify a number of cases. The results were then interpreted according to the nature of the test. Note that the code used in these experiments is available for anonymous download from the author's webpage: **http://www.csc.liv.ac.uk/~maya/PADUA_App.html.** Table 5.2 provides a summary of the PADUA parameters used in the experiments included in this chapter and their values.

| **Experiment** | **Strategy** | **Support** | **Conf** |
|---|---|---|---|
| Evaluating the Operation of PADUA. (5.2.1, 5.2.2 and 5.2.3) | **Proponent**: disagreeable build strategy in win mode.<br>**opponent**: disagreeable build strategy in win mode<br>. | 1% | 50% |
| Evaluating the Operation of PADUA as a Classifier. (5.3.1, 5.3.2 and 5.2.3) | | | |
| Assessment of PADUA's Robustness to Noise (5.4.1 and 5.4.2) | | | |
| Applying PADUA to Misinterpreted Data (5.5.1, and 5.5.2) | | | |
| Assessment of the Role of Strategy in PADUA (5.6) | All Possible Stratgies. | | |

**Table 5.2. PADUA input parameters.**

[21] This value indicates the probability of PADUA producing results (at least) as good as the other classifier, assuming that the null hypothesis is true.

Where any of the identified comparator classifiers, which all use a single dataset, was applied, they operated on the union of the two datasets (the original sets). Where applicable the same *support/confidence* values were used.

## 5.2. Evaluating the Operation of PADUA

This section provides an empirical analysis of the process of "*Arguing from Experience*", embodied in PADUA. The included experiments made use of the seven datasets (itemised above), which were chosen because they represented a diverse set of past experiences, providing a range of coverage suitable for experimenting with PADUA. The study comprised three experiments:

- The first experiment assessed PADUA dialogues by means of the accuracy of the resulting classification. A high accuracy indicated that the underlying argumentation process can be used as means to enable joint reasoning from past experience amongst two participants.
- The second experiment evaluated the improvement in the operation of PADUA when nested dialogues are applied.
- The third experiment examined the average length of PADUA dialogues with respect to the individual application domain (dataset).

### 5.2.1. The Accuracy of the Underlying Dialogues

The first experiment involved applying a number of TCV tests. This was achieved by running PADUA ten times leaving out one tenth (in order) of the available data. In each run this 10% was applied as a test set, and the accuracy of the run was calculated as the number of correctly classified cases from the training set divided by the size of the set. Figure 5.1 shows the average accuracy obtained for each dataset. Note that PADUA obtained above 90% accuracy in all cases. This high accuracy indicates that "*Arguing from Experience*" was productively utilised in PADUA, between two parties, to come to a classification of cases in each of the included domains. The two parties mined arguments, as needed, from their own datasets, and efficiently placed them in the underlying dialogue games.

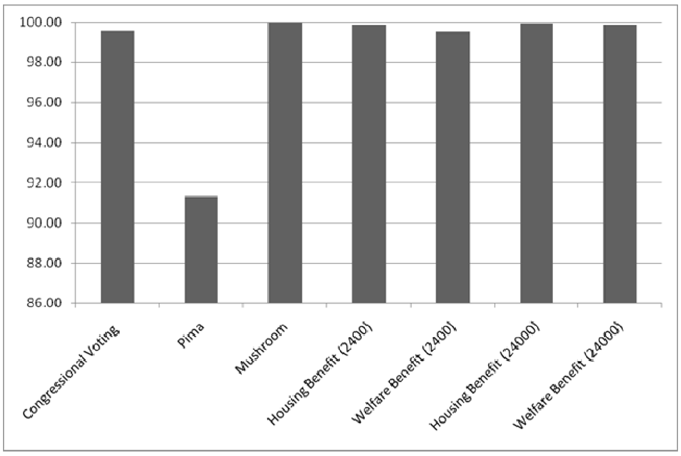


**Figure 5.1. Accuracy of PADUA TCV tests.**

### 5.2.2. Assessment of Nested Dialogues

The previous chapter explained how PADUA handles intermediate classifications by applying nested dialogues to resolve these classifications, then applying the results of these dialogues in the dialogue over the main classification problem. For the purposes of evaluating if the performance of PADUA benefits from applying nested dialogues, these dialogues were applied over the issue of contribution years in the RPHA domain. Here, both players had to engage, first, in a nested dialogue to determine whether the candidate under consideration has paid their contribution in at least three out of the last five years, the result of which was then used in the main dialogue. Thus the attributes representing the contribution condition were expressed in the main dialogue as a *Boolean* value: paid enough contributions or not. A TCV test was performed to measure the average accuracy obtained when nested dialogues were applied using the Housing Benefits (2400) dataset. The result was then compared with those obtained from the TCV test discussed in Sub-section 5.2.1. The obtained results showed an increase in PADUA accuracy from 99.87% when no nested dialogues were performed to 99.96% with nested dialogues. These results suggest that applying nested dialogues could improve the operation of PADUA.

### 5.2.3. Discussion about the Length of the Dialogues

The analysis reported thus far has shown that PADUA can successfully facilitate "*Arguing from Experience*" between two players. A detailed investigation of the underlying dialogues can now be discussed. This discussion is intended to confirm the mechanism by which PADUA aids the process of "*Arguing from Experience*", and to provide further evidence of the soundness of the dialogues conducted by PADUA. This sub-section provides an analysis of the average length of the dialogues measured by the average number of rounds PADUA requires to reach decision about cases in given datasets. If dialogue games finish too quickly, or if they take a large number of rounds to resolve each case, then PADUA will be rather inapplicable to solve real world scenarios. Fortunately this is not the case as exemplified in Figure 5.2 which shows the average number of rounds for each of the domains considered in the TCV tests of Sub-section 5.2.1. For instance, with the congressional voting data set the average length of dialogues was 12.9 rounds; with a standard deviation equal to 5.89 (the longest dialogue recorded took 19 rounds to complete). Note that the longest dialogues take place in the Mushroom dataset. This is because records in this domain comprise a large number of attributes providing players with ample means to attack and counter attack each others, thus prolonging the dialogues.

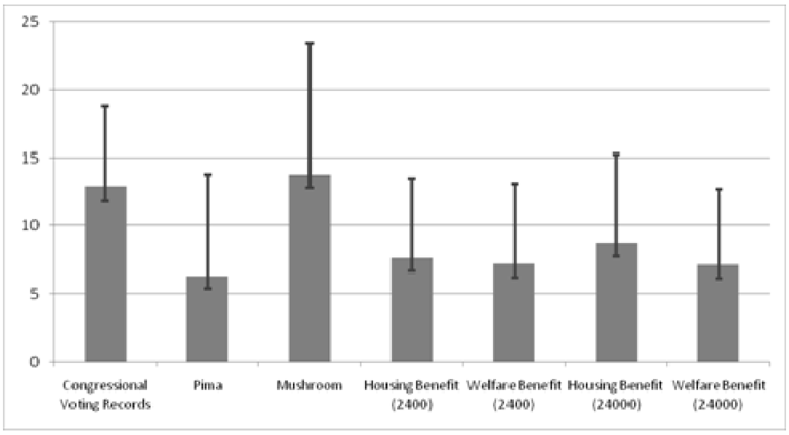


**Figure 5.2. The average number of PADUA rounds per domain.** *Error bars represent the standard deviation.*

Another issue of note in this discussion is the number of rounds consumed in nested dialogues (where applied) and their consequences on the overall dialogue length. If nesting dialogues considerably prolong the overall dialogues then such a procedure might not be desirable. To examine this point, the average lengths (and the standard deviation (SD)) of the nested and main dialogues produced in Sub-section 5.2.2 were calculated. The results of these calculations reveal that the average length of both dialogues combined is no worse than the average length observed when no nesting was applied. Without any nesting PADUA required 7.68 rounds on average to come to a conclusion about the classification of case from Housing Benefits (2400) dataset. With nested dialogues, the number of rounds PADUA needed to complete nested dialogues over the issue of contribution payments was on average 5.18 rounds (SD= 4.676), and then PADUA finished the main dialogue in 2.204 rounds (SD= 0.88). Thus, PADUA required on average 7.38 rounds to finish both dialogues. This indicates that applying nested dialogues slightly shortens the length of the overall dialogues.

## 5.3. Assessment of PADUA as a Classifier

This section provides evidence of the application of PADUA as a classifier in two-class domains. Two experiments were undertaken to provide a thorough analysis of this application:

- The first experiment compared the accuracy of the classifications obtained from PADUA from each identified dataset with those obtained from the other identified classifiers.
- The second experiment examined the difference in behaviour between PADUA and each of the applied classifiers.

### 5.3.1. Comparison with other classifiers

One of the most distinctive features to emerge from the analysis given in the previous section was the possibility of exploiting PADUA to solve binary classification problems. In order to establish PADUA as a worthy classifier, its average accuracy, in each of the itemised datasets (as calculated in Sub-section 5.2.1), was compared to the accuracies obtained from applying TCV tests using the identified classifiers. Figure 5.3 illustrates the results obtained from these TCV tests. The displayed results suggest that PADUA can be exploited as a classifier with two-class datasets. Also, PADUA outperformed the other classifiers in three cases (Pima, Housing Benefit (2400) and Welfare Benefit (2400)), and achieved the second best accuracy in all the other domains. These observations merit further discussion. RDT outperformed all the other classifiers in the Congressional Voting Records domain mainly because of the nature of these records. All the 16 attributes in this dataset have a binary value (yes/no) which provides an ideal format for the application of decision trees algorithms. Note that the performance of PADUA improved when moderately large datasets were used. Very small datasets do not provide enough experience to back the arguments advanced by PADUA players in the context of each dialogue game. Very large datasets, on the other hand, may increase the processing time required to build the P- and T-trees data structure employed by each player.

The average accuracy across all the domains included in the previous test was also calculated. Figure 5.4 illustrates these results. Note that average performance of PADUA (98.57%) is better than the other classifiers (e.g. RDT 97.48% and FOIL 95.3%) because PADUA has performed consistently with the seven included datasets. The results reported thus far encourage the application of PADUA as a classifier utilising dialogue games between two players, each representing one possible classification in some two-class domain, to classify cases from this domain. The reason why PADUA does well as a classifier lies in its underlying dialectical process, in which the two possible classification of each case are debated, by the two players, each defending its own thesis, until they come to a decision of which classification suits the given case.

Note that the TCV run time for PADUA with each dataset was as follows (in seconds): Housing Benefits 2400 (94.06), Housing Benefits 24000 (169.53), welfare 2400 (112.66), welfare 24000 (212.16), Congressional Voting (20.13), Mushrooms (226.76) and Pima (8.95). This is of course considerably longer than the runtime recorded for the other included classifiers which could be measured in milliseconds, rather than seconds.

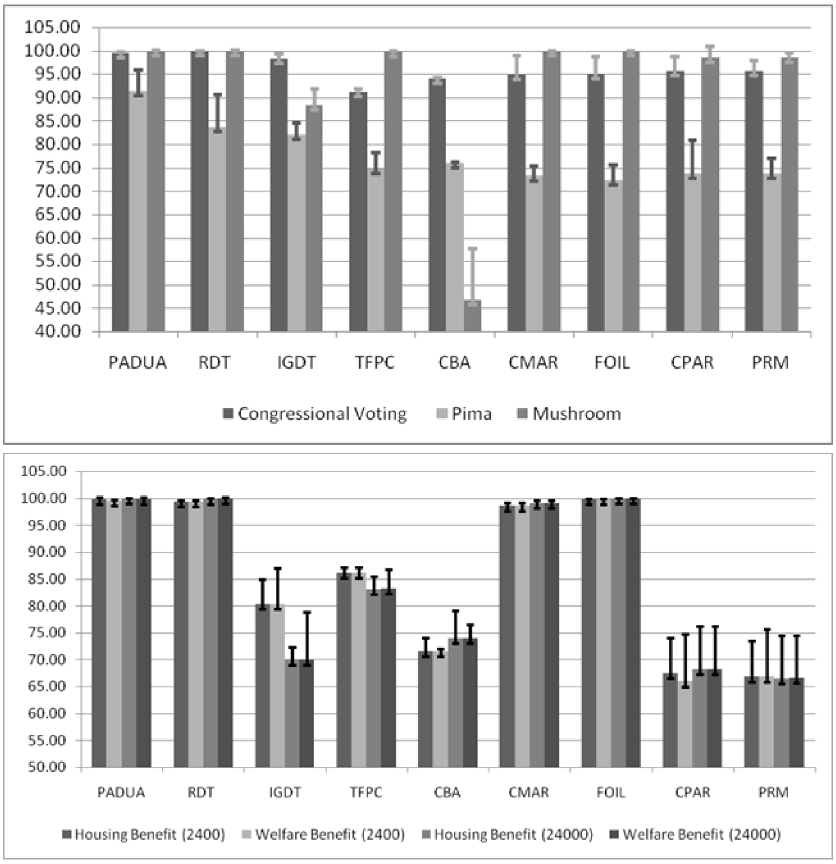


**Figure 5.3. Accuracy of the TCV tests for the two-class domains.** *Error bars represent the standard deviation for each classifier.*

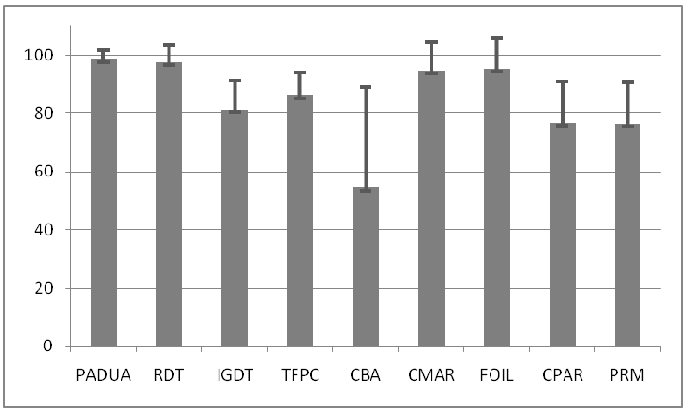


**Figure 5.4. Average accuracy across all two-class domains.** *Error bars represent the standard deviation for each classifier.*

## 5.3.2. Analysis of Applying McNemar's Test

The McNemar's test was applied with each of the seven datasets to explore the hypothesis that PADUA is significantly better than any of the other included classifiers, and to examine the differences in behaviour between PADUA and each of the other classifiers in turn. For each McNemar's test 100 cases were randomly drawn from each dataset to provide the basis for comparison. Once these cases were chosen, the rest of the cases in each dataset were split into two halves, each half providing the background experience for one PADUA player. Moreover, for PADUA, two runs were performed per each dataset; one in which the agent with the first half of the dataset was the proponent, and one in which the agent with the first half of the dataset was the opponent. The results were then compared with the results obtained using the eight identified classifiers with the union of the two datasets. Table 5.3 shows the *P-value* associated with each McNemar's test. In this table, PA2 (PADUA2) refers to the case in which PADUA was applied with the second half of the original dataset assigned to the proponent. Also, as part of the McNemar testing detailed information as to which cases were misclassified by one or both of the classifiers under consideration was also generated. Figure 5.5 illustrates these results comparing PADUA with the decision trees classifiers (the closest competitors), and with PADUA2.

| **Domain** | **PA2** | **RDT** | **IGDT** | **TFPC** | **CBA** | **CMAR** | **FOIL** | **CPAR** | **PRM** |
|---|---|---|---|---|---|---|---|---|---|
| **Congressional Voting** | 0.48 | 0.48 | 0.48 | **0.0015** | 0.22 | 0.48 | 0.68 | 0.48 | 0.48 |
| **Pima** | 1 | 0.58 | 0.72 | **0.0059** | 0.021 | **0.0035** | **0.0017** | **0.0021** | **0.0021** |
| **Mushroom** | 0.48 | 0.48 | **0.0007** | 1.00 | **<0.0001** | 0.4795 | 0.48 | 0.48 | 0.48 |
| **Housing Benefit (2400)** | 1.00 | 0.68 | **0.0003** | **0.0033** | **<0.0001** | **0.0012** | 0.25 | **<0.0001** | **<0.0001** |
| **Welfare Benefit (2400)** | 1.00 | 0.37 | **0.0012** | **<0.0001** | **<0.0001** | **<0.0001** | 0.07 | **<0.0001** | **<0.0001** |
| **Housing (24000)** | 0.48 | 0.48 | **<0.0001** | **0.0012** | **<0.0001** | **<0.0001** | 0.48 | **<0.0001** | **<0.0001** |
| **Welfare (24000)** | 0.48 | 0.61 | 0.08 | **<0.0001** | **<0.0001** | **<0.0001** | 0.25 | **<0.0001** | **<0.0001** |

**Table 5.3. The P-value associated with McNemar's Tests.** *<0.0001 indicates that this value is less than 0.0001. Values in bold indicate significant differences***.**

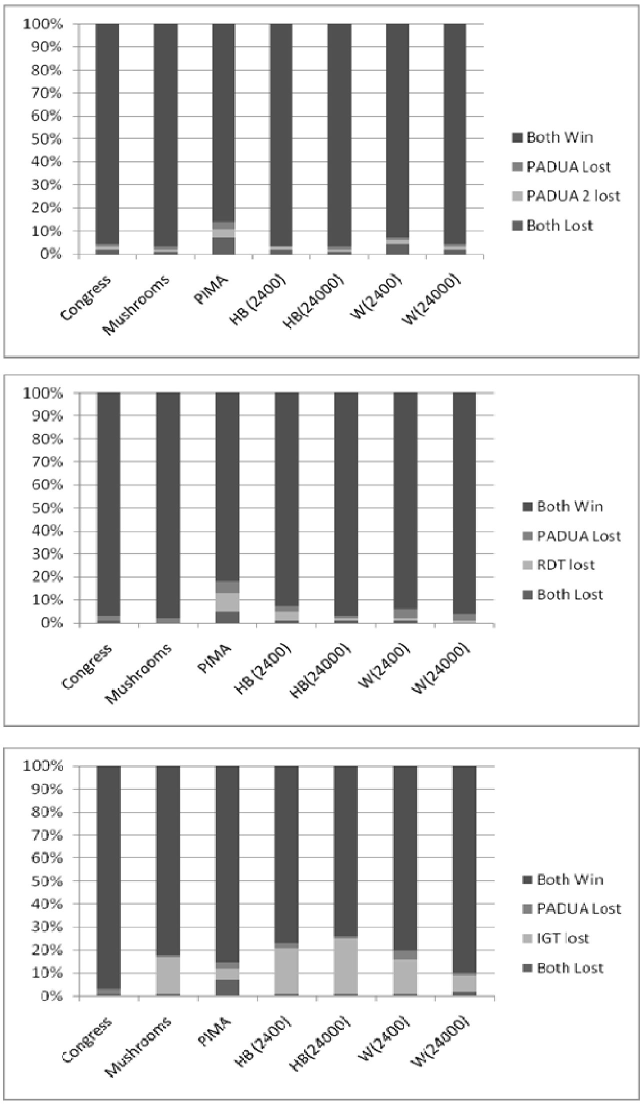


**Figure 5.5. Comparing PADUA with PADUA2 and the Decision Trees classifiers included McNemar's Test.**

Interestingly, McNemar's test reveals that there are not any significance differences between the performance of PADUA and the performance of RDT, or PADUA2, or that of PADUA and FOIL or CMAR in the domains where these classifiers performed better than PADUA. Also the results suggest that PADUA is significantly better than the other classifiers with most of the domains. Moreover, even though both applications of PADUA succeeded with similar accuracy over all the domains, yet the mistakes made by each application were different. Therefore the joint application of PADUA and

PADUA2 significantly increases the overall accuracy of applying PADUA as a classifier. It is also worth noting that the same applies for both RDT and FOIL. For example, if both PADUA and RDT were applied to classify cases from the Pima dataset, then the accuracy of classification will increase from 90% to 95%. In the case of RPHA domain the accuracy will increase to 99% when combining these two methods together. This suggests that PADUA could profitably be used with a decision tree method or a covering method in combination.

The investigation, reported thus far, establishes that PADUA provides a classification mechanism competitive with other classification systems in the absence of noise. Since, however, we can never be sure that the data will be perfect, a study of the operation of PADUA with datasets infected with noise was seen as necessary. The following sections provide an extensive account of PADUA's ability to handle different types of noise.

## 5.4. Assessment of PADUA's Robustness to Noise

The ability to handle noisy data is seen as important because it must be recognised that real-world data will often contain wrongly classified examples, representing misconceptions and mistakes. In certain domains, such as welfare benefits, it is estimated that 30% or more of previous examples may have been wrongly classified (Mozina et al, 2005). In the following, different types of noise are introduced to the different domains discussed in Section 5.1. The reported results will show that PADUA can cope more readily than the other classifiers when given noisy input data.

### 5.4.1. The Effects of Random Class Noise

This sub-section provides an assessment of the robustness of PADUA with respect to random class noise. The model used to introduce noise was the same as that reported in (Mozina et al, 2005): for N% noise in a dataset of (I) instance, ((N/100)*I) instances were randomly selected and the class label changed to some other randomly selected value (with equal probability) from the set of available classes. The experiment reported below comprises three parts. The first two inspect the consequences of introducing random noise to the welfare benefit and housing benefit artificial datasets. The last part reinforces the results of the first two parts by applying noise to the real world datasets.

The reason behind using the welfare benefit dataset to examine the effects of random class noise was twofold. Firstly, this dataset was handcrafted without any missing attributes or any unintended noise. Thus, it provided an ideal background to study the effect of intentional noise. Secondly, this dataset had been used previously in the context of examining the effect of noise on other systems (e.g. (Mozina et al, 2005)). Therefore, by applying PADUA to this domain, the horizon of comparison was extended to include machine learners and rule induction algorithms such as CN2 (Clark and Niblett, 1989), and ABCN2 (Mozina et al, 2005) which an argument based variation of CN2, without the need to worry about the implementation of these algorithms. The welfare dataset, used in this test, comprised of 2400 records such that half were classified as "*entitled*" (to benefit) and the other half as "*not entitled*". The noise levels applied to this dataset were: 2%, 5%, 10%, 20% and 40%. For each noise level, a random 70% of the rows in the dataset were used as the training set and the rest (30%) as the test set. Noise was then applied to the training set only and not to the test sets. The training set used for each of the noise levels, was split into two equal subsets, one given to the proponent and the other to the opponent in PADUA. The two players argued to classify the 720 cases in the test set. Table 5.4 shows the affect of adding noise to the Welfare dataset on the accuracy of each classifier. As expected the accuracy of all the classifiers drops as the noise level increases. When using clean data (no noise) RDT outperformed all the other classifiers, with PADUA producing acceptable results. However, as the noise level increased PADUA was observed to be more tolerant to noise: its accuracy dropped only 2.78% even when the noise level was increased to 40%, while the accuracy of RDT dropped 3.61%. The other classifiers suffered even more severe drops in their accuracy levels (FOIL's accuracy dropped 10.28% with 40% noise). These results indicate that PADUA is more tolerant to noise than the included classifiers. Note that the results for

CN2 and ABCN are taken from (Mozina et al., 2005), while the others were produced as part of the experiment.

| N | PADUA | RDT | IGDT | TFPC | CBA | CMAR | FOIL | CPAR | PRM | CN2 | ABCN2 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 99.86 | 100 | 92.50 | 98.47 | 99.17 | 96.81 | 99.72 | 67.08 | 66.67 | 99.47 | 99.76 |
| **2** | 99.86 | 98.6 | 88.19 | 98.33 | 100 | 98.75 | 100 | 65.36 | 65.36 | 97.78 | 98.42 |
| **5** | 99.31 | 99.6 | 93.33 | 99.86 | 98.75 | 98.1 | 94.17 | 65.36 | 65.36 | 96.36 | 96.96 |
| **10** | 98.47 | 98.3 | 92.78 | 97.08 | 91.94 | 97.19 | 93.19 | 64.44 | 64.44 | 93.51 | 94.69 |
| **20** | 97.78 | 97.3 | 90.97 | 98.75 | 86.94 | 97.33 | 88.89 | 61.67 | 63.61 | 88.69 | 92.00 |
| **40** | 97.08 | 96.4 | 90.44 | 96.25 | 94.03 | 96.80 | 89.44 | 58.06 | 57.92 | 83.26 | 85.03 |

**Table 5.4. Accuracy versus Noise (PADUA – Welfare Dataset)[22].** ***N= noise percentage (%) in the training dataset.***

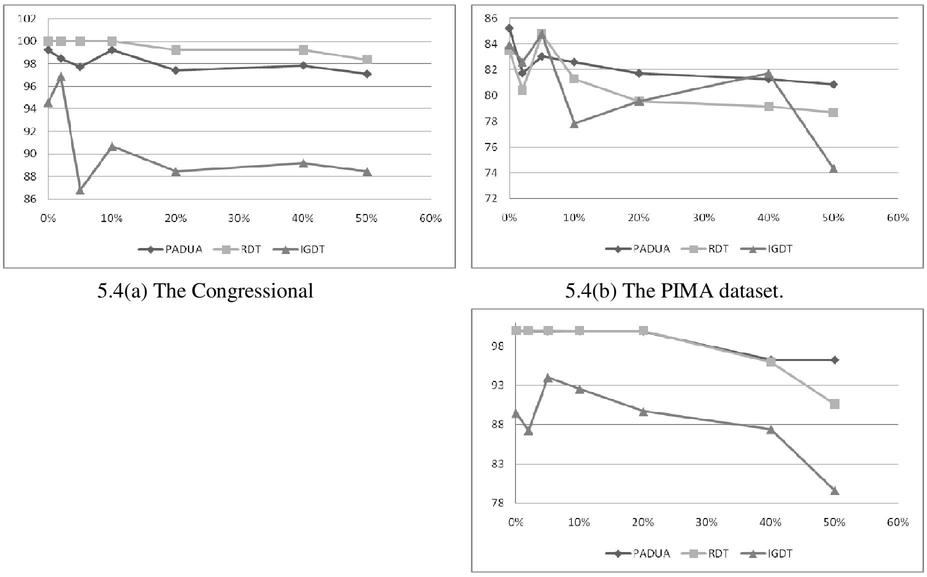
The effect of random class noise on PADUA was further evaluated by applying it to the Housing Benefit (2400) dataset (Section 5.1) configured in terms of two classes: *entitled* and *not entitled*. For the evaluation the 2400 records were again generated and distributed evenly over the two classes. The *not entitled* cases were generated such that they fail to meet one and only one condition of the five conditions listed above. This dataset was then randomly split into 70% training set and 30% testing set and noise was then applied to the training set in the same manner as in the previous evaluation. However, in this case an extra noise level of 50% was added to the experiment. Again the training dataset used for each noise level was split equally between two PADUA players and they were allowed to "*argue*" to classify the 720 cases in the test set. Table 5.5 shows the results of this experiment. Here it can be noticed that FOIL was the best classifier when using correct data (unlike the previous experiment), but again it can be observed that as the accuracy of all the classifiers drops with the increase in noise level in the data. PADUA is again more tolerant of noise that the other classifiers. The accuracy of PADUA drops 5.83% as the noise level is increased from 0% to 50% whereas the accuracy of FOIL (which worked well with clean data) drops 21.81% and the accuracy of RDT drops 10.97%.

[22] The CN2 and ABCN2 results are those given in (Mozina et al., 2005).

| Noise % | PADUA | RDT | IGDT | TFPC | CBA | CMAR | FOIL | CPAR | PRM |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 99.86 | 99.72 | 77.00 | 98.33 | 97.36 | 99.31 | 100 | 64.03 | 66.81 |
| **2** | 99.72 | 97.78 | 76.25 | 98.61 | 99.86 | 98.01 | 96.67 | 63.75 | 64.72 |
| **5** | 99.58 | 98.89 | 64.31 | 96.53 | 97.50 | 98.61 | 94.44 | 65.28 | 65.14 |
| **10** | 98.61 | 98.75 | 73.61 | 93.61 | 91.11 | 95.69 | 87.08 | 63.61 | 64.92 |
| **20** | 96.81 | 98.19 | 73.06 | 93.89 | 96.25 | 96.50 | 86.39 | 62.28 | 64.58 |
| **40** | 96.11 | 92.22 | 64.44 | 83.06 | 92.08 | 92.92 | 86.11 | 60.97 | 61.25 |
| **50** | 94.03 | 88.75 | 62.22 | 54.72 | 84.17 | 85.31 | 78.19 | 59.58 | 61.81 |

**Table 5.5. Accuracy versus Noise (PADUA – Housing Benefit Dataset).**

The tests described above were desirable mainly because it was feasible to acquire a full understanding of the artificial datasets included in these tests. But relying only on artificial datasets is not enough to demonstrate how PADUA tolerates noise. Therefore PADUA was applied to a set of real world datasets (Table 5.1). Here the operation of PADUA was compared with the same classifiers as used before, but only the comparison with decision trees classifiers is reported here, because decision trees were found to be the closest "*competitors*" to PADUA when using real world datasets. The results of this evaluation are illustrated in Figure 5.6, in which the horizontal axis represents the noise level and the vertical represents the accuracy. These results show a similar pattern to the benefits experiments. The accuracy of almost all the classes dropped when the noise percentage was increased. The only case in which PADUA performed worse than RDT, with high levels of noise, was when the Congressional Voting Records dataset was used. The reason is that this dataset is very small (435 rows), which means that each player has only 152 cases from which they could mine their arguments (ARs). This is rather a small size when a high level of confidence is used. In addition, as noted above, this dataset comprises only binary valued attributes and thus lends itself to the decision tree classification paradigm. The average accuracy across all the five domains included in the previous discussion was calculated. The results show that the accuracy of RDT classifications (96.638%) is higher than any other classifier when no noise is introduced. But once random noise is introduced to the datasets, PADUA emerges as the classifier with the best average accuracy, starting with slightly better performance (0.586%) than RDT and with the difference in accuracy between PADUA (92.06%) and RDT (89.11%) reaching 2.95% when 40% of the data is noisy.



5.4(a) The Congressional

5.4(b) The PIMA dataset.

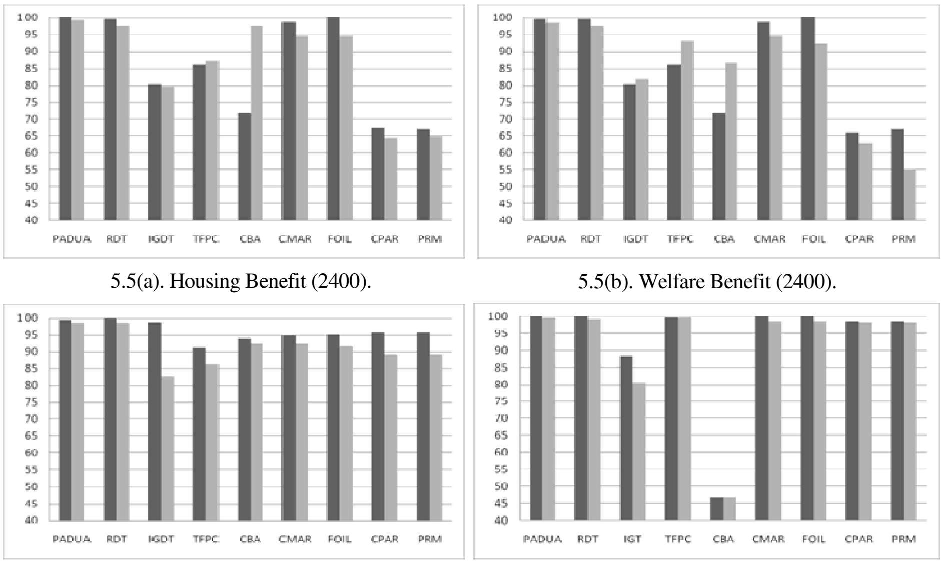
5.4(c) The Mushroom dataset.

**Figure 5.6. The effect of applying noise on: PADUA, RDT and GDT.**

In summary, PADUA's robustness to noise was emphasised by detailed experiments using two artificial welfare datasets, and summary results for three real datasets. The results obtained indicate that PADUA's is comparable to, or better than, other classification approaches. The particular advantage that PADUA offers is that it operates very successfully in noisy environments, outperforming competitor classification systems. The ability to handle noisy datasets is of significant importance in the many domains where sufficient data can only be obtained at the cost of including misclassified records.

### 5.4.2. Missing Attributes and the Operation of PADUA

PADUA was shown to exhibit the ability to handle random class noise. Here, another type of noise is closely examined: the effect of the absence of some attributes, other than the class attributes, on the operation of PADUA. The reported experiment was configured as follows: First, each included dataset was split into two halves. A random attribute was then omitted from 50% of the records in each half. The two halves were then assigned to the two PADUA players, and a TCV test was performed. The performance of PADUA was then compared to the results of applying the other classifiers to the union of the two halves. Figure 5.7 illustrates the results obtained from omitting two random attributes from three of the datasets used in the previous sub-section[23].



5.5(a). Housing Benefit (2400).

5.5(b). Welfare Benefit (2400).

5.5(c). Congressional Voting Records.

5.5(d). Mushrooms.

**Figure 5.7. Results of omitting two random attributes.**

In the above figure, dark bars in each diagram represent the accuracy obtained when all attributes are present in the dataset and the light ones the accuracy obtained from the datasets with missing attributes. Note that although the accuracy of almost all the classifiers drops when two random attributes are omitted from 50% of the data, PADUA is more resistant to this type of noise than the other classifiers. In the case of the Housing Benefit dataset the accuracy of PADUA dropped 0.794% when two random attributes were omitted (the attribute representing if a payment was made toward the contribution two years ago from the 50% of the proponent dataset, and the gender attribute from 50% of the opponent dataset). The accuracy of RDT, the closest competitor classifier, had dropped by 1.88%. The same pattern was repeated with the other three

[23] Pima and Mushroom datasets were not included in this test because the attributes of these sets were discretised in a way that makes omitting one or two of them insignificant.

datasets. These results emphasised the conclusion of the previous sub-section that PADUA is tolerant to high levels of noise.

## 5.5. Applying PADUA to Misinterpreted Data

This section discusses a different account of noise, other than that examined in the previous section. Here, the focus is on errors that are not random. Rather they emerge from different interpretations of similar cases. These errors are referred to as "*systematic errors*". Such errors are most significant in the context of assessment of claims to benefits, due to the high error rate encountered with this assessment. Groothius and Svensson (2000) drew attention to this point in connection with the Netherlands General Assistance Act, and reported experiments which suggested that an error rate of more than 20% was typical. The problem is international: the US National Bureau of Economic Research reports[24] that the multistage process for determining eligibility for Social Security Disability Insurance (DI) benefits causes a variation in the award rates across the states. Similar observations are made of the UK. An official UK Publication produced by the Committee of Public Accounts[25] states that the error rate for Disability Living Allowance, for example, is as high as 50%. The same publication also notes that: "*There are also regional differences in decision making practices that may lead to payments to people who are not eligible for benefits*". There is thus a significant problem regarding the process of awarding benefits, indicating that current procedures are unable to provide a satisfactory service. One important feature of errors encountered when assessing benefits, is that they are not random; regional differences in decision making practices arise from the complexity of the rules and regulations because the misunderstandings and misinterpretations differ from office to office. Thus, one office will tend to decide one class of case wrongly, while a different office will get this right, but fail on another class of cases.

[24] From Web Page: http://www.nber.org/aginghealth/winter04/w10219.html.

[25] Getting it right: Improving Decision-Making and Appeals in Social Security Benefits. Committee of Public Accounts. London: TSO, 2004 (House of Commons papers, session 2003/04; HC406).

One way of resolving disagreement in assessment, such as the ones described above, is to have a *moderation* dialogue, in which the parties in disagreement may argue for their positions with one another. Thus, they can come to recognise strengths and weaknesses that they have overlooked or under weighted and so converge on agreed decisions. This section describes how PADUA can be exploited as means to facilitating this process for the Retired Persons Housing Allowance (RPHA) decisions made in different offices. The results reported below demonstrate that by applying PADUA the misclassifications in the database can be reduced to less than 10%.

### 5.5.1. Argument Based Moderation of Benefit Assessment

Below the kinds of dialogues produced by PADUA when the disagreement between its two players is a result of one party misinterpreting the input data is illustrated experimentally. For this purpose, PADUA was applied to the fictional RPHA scenario, as described in Section 5.1. Let us suppose that the RPHA benefit is assessed in two different offices, covering different regional areas, and each producing errors through a different misinterpretation. Three experiments were performed:

- **Experiment SE1**: Examined the extent to which classification would be improved by moderation using PADUA. This was done using a TCV test. A number of other classifiers were also applied to the data to provide a comparison.
- **Experiment SE2:** involved applying a McNemar's test to show the significance of the differences between PADUA and the other classifiers.
- **Experiment SE3:** provided a more detailed analysis of the performance of PADUA to discover some interesting properties of the moderation process.

In order to perform the above experiments, two sets of (RPHA) Housing Benefits data were generated[26]. Each record comprised thirteen fields, the

[26] Note that the structure of these datasets is different from the ones outlined previously, because such structure enables a more detailed analysis of the performance of PADUA.

information relevant to the above experiments being surrounded by other features which should be irrelevant to the determination of the case. Both contained 500 cases which should be awarded benefit, and 500 cases which should be denied benefit. Cases can fail on any one of five conditions, and the failing cases were evenly divided across them. One dataset (DS1) was completed by the addition of 500 cases which should fail on the age condition, but which in fact awarded benefit to men over 60, and the other (DS2) with 500 cases which should have failed the residence condition, but which interpreted the exception too widely, allowing benefit to members of the *Merchant Navy* and the *Diplomatic Service*.

In SE1, the baseline was the number of correct cases in the dataset: namely the 66.7% accuracy which had been achieved by the original decision makers (Bench-Capon, 1991). Eight other classifiers were used, operating on the union of the two datasets (TFPC, CBA, CMAR, RDT, IGDT, FOIL, PRM and CPAR). The TCV tests were conducted in the same manner as the previously reported experiments. For PADUA, two runs were performed, one in which the agent with DS1 was the proponent (argued for award of benefit), and one in which the agent with DS2 was the proponent. Figure 5.8 illustrates the results.

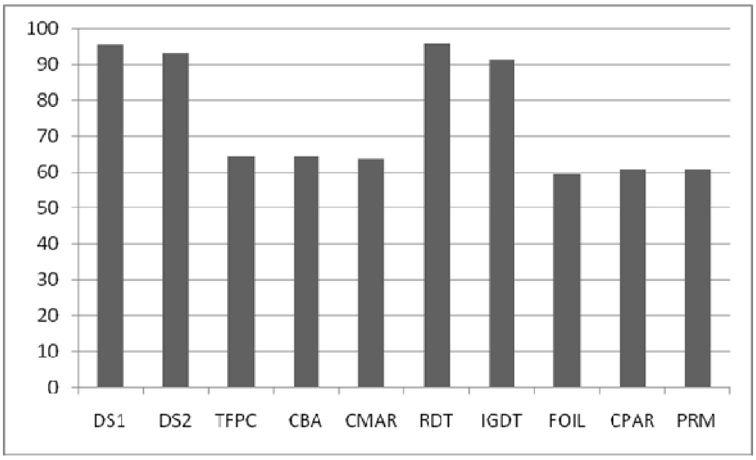


**Figure 5.8. Results of TCV tests using data with systematic errors.**

From this figure it is evident that the three CARM classifiers perform less well than the baseline. In contrast, PADUA, and the decision tree based classifiers perform significantly better, attaining above 90% accuracy in all the ten trials (on average 95.51% with DS1 and 92.95% with DS2). While the decision tree classifiers perform rather consistently throughout the ten trials (95.88% on average), there is more variation in PADUA, especially for DS2, suggesting that its performance is more sensitive to the exact sample available to the agents. This point will be considered in more detail in the discussion below.

Overall, the level of PADUA performance was encouraging. For comparison with other AI and Law systems, Bench-Capon (1993) reported an accuracy of 98%, but that was based on training set of correctly decided cases. Ashley and Brüninghaus (2003) reported a success rate of 91.4% for IBP, and Chorley and Bench-Capon (2005) a success rate of between 91% and 93% for AGATHA, both applied to noise free examples of US Trade Secret Law. It seems therefore, compared with this previous work that the level of accuracy attained by PADUA was towards the top end of what can be expected from successful AI and Law systems. McNemar's tests were also performed to explore whether PADUA was better or worse than any of the other classifiers used in the previous experiment. As might be expected from the results shown in Figure 5.8, PADUA DS1 and DS2 were significantly better than the three CARM classifiers and IGDT, but not significantly better or worse than RDT. For these tests, PADUA operated on a set of newly generated cases (500 positive, 500 negative as before and 250 wrongly decided, appropriate to each database)[27]. This data was then used as a test set for the other classifiers, the original data supplying the training set. As part of the test detailed information was generated as to which cases were misclassified by one or both of the classifiers under consideration. The results for DS1 and DS1 showed that the performance of PADUA (using both DS1 and DS2) was significantly better comparing to any classifier other than the two decision tree classifiers (RDT and IGDT). The comparison between PADUA DS1 and DS2 are presented in Tables 5.6(a) and (b).

[27] Again, the layout of the datasets used with the McNemar's test is different from the ones described in Section 5.1, for the same reason as footnote 26.

| Test Cases | DS2 | TFPC | CMAR | CBA | RDT | IGDT | FOIL | CPAR / PRM |
|---|---|---|---|---|---|---|---|---|
| Both Failed | 5 | 8 | 8 | 145 | 7 | 10 | 5 | 5 |
| PADUA Failed | 146 | 139 | 139 | 2 | 140 | 137 | 146 | 146 |
| Other Failed | 142 | 318 | 364 | 461 | 62 | 129 | 150 | 294 |
| Both Succeeded | 1207 | 1035 | 989 | 892 | 1291 | 1224 | 1199 | 1055 |

**Table 5.6(a). Comparison with DS1 (Wardeh et al., 2009a, 2008b).**

| Test Cases | DS1 | TFPC | CMAR | CBA | RDT | IGDT | FOIL | CPAR | PRM |
|---|---|---|---|---|---|---|---|---|---|
| Both Failed | 5 | 48 | 92 | 55 | 10 | 22 | 4 | 8 | 6 |
| PADUA Failed | 142 | 103 | 59 | 96 | 141 | 129 | 143 | 139 | 141 |
| Other Failed | 146 | 214 | 514 | 66 | 31 | 119 | 152 | 308 | 319 |
| Both Succeeded | 1207 | 1136 | 835 | 1283 | 1318 | 1230 | 1201 | 1053 | 1034 |

**Table 5.6(b). Comparison with DS2 (Wardeh et al., 2009a, 2008b).**

With respect the above tables, it is interesting to note that although both classifiers succeed[28] only on 86.07% of cases for RDT and DS1, 81.60% of cases for DS1 and IGDT and 82.00% of cases for DS2 and IGDT; the mistakes are very different. Less that 0.5% of the cases are misclassified both by DS1 and RDT and only 1.47% by the worst combination, DS2 and IDGT. This suggests that PADUA and a decision tree method could profitably be used in combination. If cases where there was agreement were believed to be correct, and DS1 and RDT were used, for example; and referred cases of disagreement to an expert for decision error rates could be reduced to below 0.5% (0.47%), at the cost running RDT first, then applying PADUA using the cases that RDT has failed to classify (4.13% of the 1500 cases, when using PADUA with DS1). Thus, by focusing the cases for expert checking, the error rate could be reduced with very little additional expert intervention (only the cases misclassified by both PADUA and RDT). Moreover, DS1 and DS2 only both misclassify one case in three hundred (0.33%), although they are both successful in only 80.47% of the cases. Using PADUA alone, but having each case argued for by both agents, therefore, could reduce the error rate to 0.33%. However, it would require executing PADUA with (say) DS1 then rechecking around 10.07% of the cases using DS2.

[28] *Both classifiers* here, refers to the results of the McNemar's test were both classifiers succeeded in correctly classifying the input cases.

In the remainder of this sub-section the TCV trials for PADUA will be considered in more detail. The detailed results are shown in Tables 5.7(a) and (b). In Table 5.7(b), Pro1 refers to the accuracy of the classifications (in % percentage) achieved by applying PADUA where the proponent uses DS1. Pro2 refers to the accuracy of the classifications (in % percentage) achieved by applying PADUA where the proponent uses DS2.

| **Test** | **Positive** | | **Negative Age** | | **Negative Income** | | **Negative Capital** | | **Negative Residency** | | **Negative Contribution Years** | | **All Female Exception** | | **All UK Exception** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **DS** | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 |
| **1** | 6 | 98 | 92 | 100 | 100 | 91 | 96 | 98 | 96 | 98 | 96 | 98 | 88 | 100 | 93 | 72 |
| **2** | 100 | 94 | 92 | 93 | 100 | 95 | 96 | 97 | 96 | 97 | 96 | 97 | 89 | 94 | 95 | 74 |
| **3** | 99 | 98 | 90 | 100 | 100 | 95 | 97 | 93 | 97 | 93 | 97 | 93 | 90 | 100 | 91 | 68 |
| **4** | 100 | 98 | 94 | 100 | 100 | 94 | 97 | 94 | 97 | 94 | 97 | 94 | 85 | 100 | 94 | 68 |
| **5** | 98 | 96 | 94 | 93 | 100 | 93 | 96 | 95 | 96 | 95 | 96 | 95 | 89 | 76 | 99 | 68 |
| **6** | 99 | 98 | 95 | 100 | 99 | 91 | 98 | 93 | 98 | 93 | 98 | 93 | 88 | 100 | 99 | 78 |
| **7** | 98 | 96 | 94 | 100 | 99 | 93 | 96 | 95 | 96 | 95 | 96 | 95 | 92 | 100 | 100 | 79 |
| **8** | 98 | 94 | 95 | 98 | 100 | 91 | 97 | 97 | 97 | 97 | 97 | 97 | 89 | 100 | 98 | 72 |
| **9** | 99 | 96 | 94 | 100 | 99 | 95 | 97 | 94 | 97 | 94 | 97 | 94 | 82 | 100 | 88 | 78 |
| **10** | 97 | 98 | 92 | 100 | 99 | 94 | 96 | 95 | 96 | 95 | 96 | 95 | 82 | 100 | 97 | 74 |

**Table 5.7(a). Detailed TCV tests results for the systematic errors experiment (Wardeh et al., 2009a, 2008b).**

| **Trial** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| **Pro(DS1)** | 95.1 | 95.5 | 95.1 | 95.5 | 96 | 96.8 | 96.4 | 96.3 | 94.1 | 94.4 |
| **Pro(DS2)** | 94.4 | 92.6 | 92.5 | 92.8 | 88.9 | 93.3 | 94.1 | 93.3 | 93.9 | 93.9 |

**Table 5.7 (b). Summary results for the systematic errors experiment (Wardeh et al., 2009a, 2008b).**

From the above table it can be noted that:

- The overall performance is rather consistent, with only trial 5 for DS2 showing a significantly worse performance that the rest. Within the detailed breakdown by types of case, however, there is rather more variation.
- Although PADUA succeeds in classifying more cases correctly, some errors are introduced. Rarely does it succeed in classifying 100% of cases in the data sets correctly. This is because the high number of misclassified cases in the dataset impairs the ability to form correct rules. In particular, the

negative age condition becomes harder for DS1, which misunderstands the exception to that condition.

- It matters who is the proponent. For example when DS1 is arguing for benefit for the misclassified age cases, it can defend itself quite a lot of the time. On the other hand when DS2 is proposing that the benefit be given wrongly in these cases, it almost invariably fails. This is readily explicable because DS2 cannot find any good reasons from its own dataset to award benefit in these cases. This effect is not observed, however, in the case of trial 5, when DS2 performs unusually badly on this factor. One assumes that this is explained by a lack of correctly classified men between 60 and 65 in the particular selection of data used by DS2 in that trial. A similar effect can be observed when cases with misclassified residency are argued for: misclassifications are more likely to be accepted when DS2, which believes them, is the proponent.

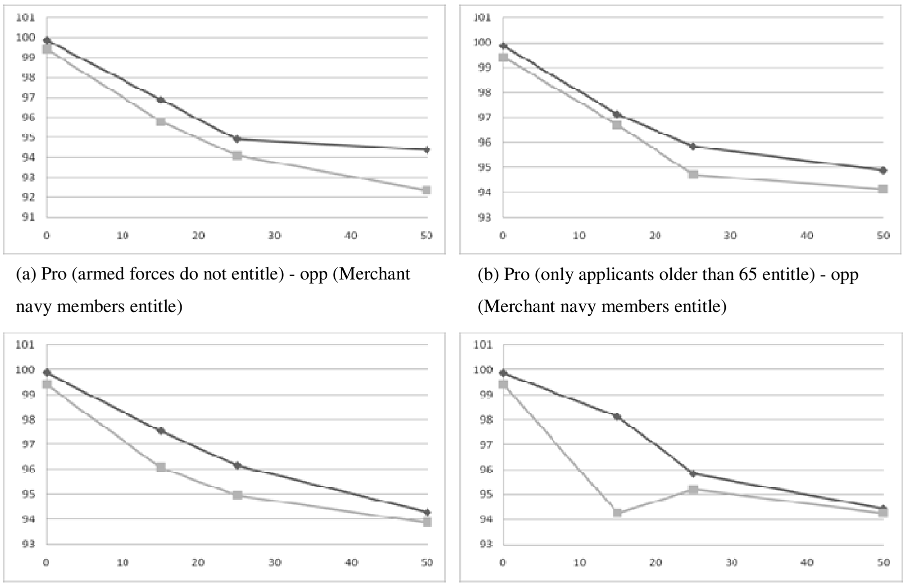
The above discussion demonstrates that PADUA provides an approach to the problem of systematic errors as discussed above. The experimental results reported in this section show that PADUA dialogues result in reducing the misclassifications in datasets containing such errors, from 33% in the original data to less that 10%, a performance superior to other association rule classifiers and comparable with decision tree classifiers. This performance also matches previously reported AI and Law systems, even where they have used only correctly decided cases for training. Moreover the results unveil that the cases which remain misclassified differ according to which agent acts as proponent and which as opponent. By running the cases with first one agent as proponent and then the second as proponent, the number of cases misclassified on both runs can be reduced to 0.3%, although there is disagreement (correct classifications obtained from one run but not both) in 19.2% of the cases, by running PADUA with DS1 first then executing a second run with DS2 with only the cases misclassified in the first run. This particular point could provide an effective way of identifying cases for expert checking, which would improve significantly on the current practice of checking a random sample. Alternatively

PADUA could also be effective when used in conjunction with a decision tree classifier.

### 5.5.2. Further Discussion

In the foregoing it has been established that PADUA is a useful tool in reducing errors in datasets containing misinterpreted records, which were referred to as "*systematic errors*". However, the previous sub-section examined only two possible systematic errors. A certain percentage of the proponent's records were misinterpreted such that the gender\age condition was expanded so that all men and women above 60 were entitled to benefit. Also, a certain percentage of the opponent's records were misinterpreted such that the residence condition was overlooked. But the number of possible misinterpretations is actually much larger. Besides, the percentage of erroneous records in the dataset of each player is also of importance. A detailed analysis was performed to uncover when the number of misinterpreted records becomes larger than what PADUA can handle. An experiment was executed to investigate all the possible combination of systematic errors, such that only one error was produced in one dataset. The percentage of these errors was then set to 15%, 25% and 50%. A TCV test was then performed for each level of errors, for each combination, for each of PADUA and the other classifiers, which were executed on the union of the proponent's and the opponent's datasets. The results obtained from these tests show that on average, when only 15% of the data is erroneous, RDT outperforms PADUA[29] with 0.24%. But when the level of systematic errors is increased to 25% PADUA outperforms RDT with 0.035%, and the gap between the two classifiers increases to 0.39%. Figure 5.9 illustrates the most significant differences in performance between PADUA and RDT obtained from the experience outlined above. The dark line represents PADUA and the light one represents RDT. Pro refers to the exception in the proponent's dataset and Opp to the one in the opponent's dataset.

[29] Only PADUA and RDT were mentioned here because both of them perform considerably better than the other classifiers. The third best classifier is GDT with accuracy ranges from 85% (errors level =15%) to 75% (error level = 50%).



(a) Pro (armed forces do not entitle) - opp (Merchant navy members entitle)

(b) Pro (only applicants older than 65 entitle) - opp (Merchant navy members entitle)

(c) Pro (only applicants older than 65 entitle) - opp (armed forces do not entitle)

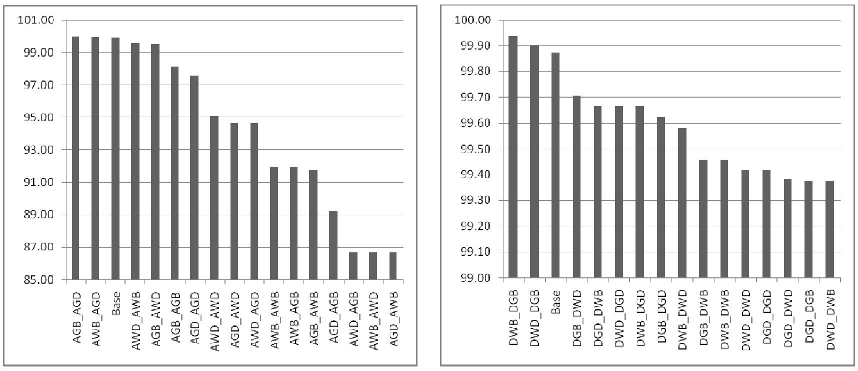
(d) Pro (it suffices to pay contribution in two /last five years) - opp (only applicants older than 65 entitle)

**Figure 5.9. The performance of PADUA and RDT with different levels of systematic noise.**

## 5.6. Assessment of the Role of Strategy in PADUA

The role of players' strategies in connection with the operation of PADUA, with respect to the results recorded in Section 5.2, will now be examined. Chapter 4 provided a detailed account of how different dialogue types could be derived by applying different strategies. Here, all the possible strategy allocations are considered in order to supplement the previous discussion with instructive heuristics as to which allocations produce the highest or lowest accuracy, and which ones lead to the longest or shortest dialogues. Players could use these heuristics to make a decision about which strategy is best applied in certain situation, as will be discussed in Chapter 9. The heuristics reported in this section could be used to improve the overall performance of PADUA. A detailed assessment of strategies in PADUA is given below. This account takes two points into consideration: (i) whether applying different strategies contributes to significantly better or worse classifications, and (ii) the effect of strategy on the length and overall accuracy of PADUA dialogues.

To address the first point, a number of TCV tests were performed to compare the results obtained in Section 5.2 to those acquired using different strategy allocations. The strategy allocation applied in Section 5.2, which is referred to as the *base* allocation, assumed that the two PADUA players apply the same strategy, namely the disagreeable build strategy in dialogue game mode. This allocation proved useful, as discussed thus far. Nevertheless, a thorough assessment of the performance of the PADUA protocol should consider other possible strategy allocations. Due to the large number of such allocations a graphical representation of the results was thought more suitable than textual one. In the figures and tables reported in this sub-section, the name of the allocation is given as two sets of three letters. The first represents the proponent's strategy and the second the opponents. Each letter represents a strategy parameter: A=Agreeable, D=Disagreeable, W=Win mode, G=Game mode, B=Build and D=Destroy strategy. Figure 5.10 illustrates how PADUA's accuracy changes when applying different strategies. The vertical bars in both diagrams represent the accuracy of one allocation; the name of which is given under the bar as two sets of three letters. The first represents the proponent's strategy and the second the opponents. The reported results suggest that the highest accuracy (99.94%) was obtained when both players applied a disagreeable dialogue game mode strategy such that the proponent built its proposition while the opponent attempted to destroy the proponent's propositions.



(a) "*Agreeable Strategies*"

(b) "*Disagreeable Strategies*"

**Figure 5.10. Results of applying PADUA with different strategies.**

Similarly high accuracy (99.93%) was obtained when the two players applied the previous allocation but maintain agreeable profiles. The worst possible strategy allocation yielded a mere 86.67% accuracy level. In this allocation the proponent applied agreeable destroy strategy in *dialogue* mode, while the opponent applied agreeable build strategy in *win* mode. Thus the accuracy dropped 13.26% from the best to the worst agreeable allocations. Note that both sides in each strategy allocation embodied in Figure 5.10 were given the same profile (either both sides are agreeable or both are disagreeable). Recall from the previous chapter that when two players apply an agreeable profile the resulting dialogues would be closer to deliberation rather than persuasion dialogues obtained when both players employ a disagreeable profile. A "*grey area*" can be identified between persuasion and deliberation dialogues where each side applies a different profile. Such strategies yielded average accuracy around 95%. For example, where the proponent applied a disagreeable profile along with a destroy strategy in *dialogue* game mode and the opponents applied the opposite allocation, i.e. agreeable *win* game mode build strategy, the overall accuracy was 95.63%. Where each player applied the other's strategy, the obtained accuracy level was 95.64%.

The TCV tests also investigated which allocations produced the longest dialogues and which produced the shortest. Figure 5.11 illustrates the average number of rounds dialogues take when applying different strategy allocations. The vertical bars represent the average number of rounds dialogues take when applying one allocation. As expected the longest dialogues were obtained when both players employ disagreeable profiles (the average number of rounds when two parties are agreeable is 2.31 rounds) while the shortest dialogues were produced when two players are agreeable (the average number of rounds when two parties are disagreeable is 8.02 rounds).

The McNemar's test was preformed to identify the strategy allocations that produce significantly better or worse classifications than the ones obtained using the base strategy. The results of this test demonstrated that only five strategy allocations yielded significantly different performance when compared with the base allocation, two of which performed in a very different manner compared with the base allocation. Table 5.8 lists the results of these allocations.

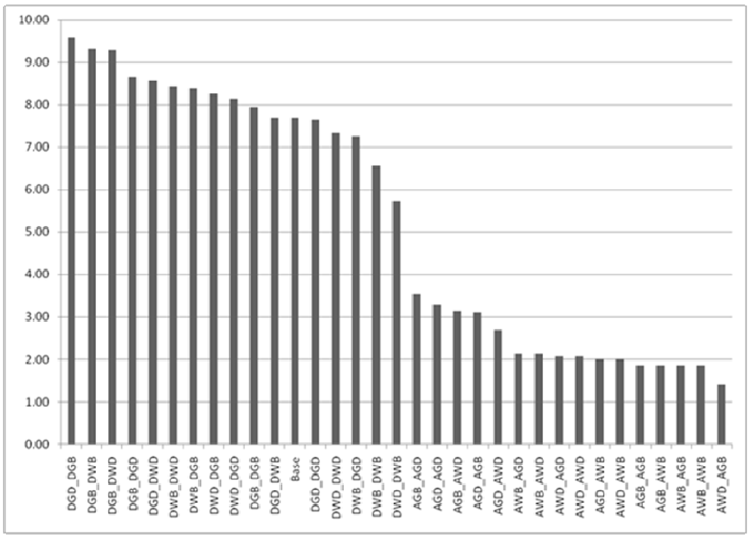


**Figure 5.11. The effect of strategy on length of PADUA dialogues..**

| Strategy | Both Lost | Strategy Lost | Base Lost | Both Win | McNemar | P-value | Significant |
|---|---|---|---|---|---|---|---|
| **AGB_AGD/ AWB_AWD** | 2 | 9 | 1 | 88 | 8.10 | 0.0269 | YES |
| **DGB_DWD** | 0 | 2 | 11 | 87 | 7.69 | 0.0265 | YES |
| **AWB_AGB** | 1 | 13 | 2 | 84 | 9.60 | 0.0098 | VERY |
| **AWB_AGD** | 1 | 13 | 2 | 84 | 9.60 | 0.0098 | VERY |

**Table 5.8. Detailed results of the significantly different strategy allocations.**

## 5.7. Summary

This chapter provided evidence that PADUA can facilitate "*Arguing from Experience*" dialogues between two players in a variety of domains. An analysis was undertaken by means of empirical experiments intended, mainly, to demonstrate that reliable dialogues can be conducted using PADUA. This was emphasised by the high accuracy (above 90%) obtained using PADUA in the reported experiments. This accuracy indicates that PADUA can be used, successfully, to resolve conflicts between two parties over cases from some domain, by means of "*Arguing from Experience*". PADUA was also shown to exhibit a high resistance to different types of noise. Also, this chapter provided a detailed account of some of the elements of PADUA such as strategy and nesting. Nested dialogues were shown to improve PADUA's ability to resolve conflicts over the issue of classifying cases from the RPHA domain. An extensive experiment including all the possible strategy allocations has shown that applying different strategies for move selection gives rise to dialogues with different characteristics. The results reported throughout this chapter also suggest that PADUA can be profitably exploited as means to classifications. This was investigated by comparing PADUA to other well-known classifiers. PADUA was shown to be competitive with the included classifiers. Moreover, unlike other classifiers, PADUA enjoys the following desirable features:

- It does not require a training phase: classifying fresh cases could be achieved without any previous preparations.
- It is noise tolerant: it can cope with high levels of noise in the datasets without failing to classify correct cases.
- It provides the user with a set of parameters (e.g. support\confidence values, and the strategy configurations). By changing some of which the user can modify the course of PADUA dialogues to better fit the underlying datasets.

However, the PADUA implementation examined in this chapter is not complete and there is room for some improvements. Some of these possible upgrades will be discussed in the final chapter of this thesis. This concludes the analysis of the PADUA protocol. The next chapter will present an argumentation system called PISA (Pooling Information from Several Agents) which allows any number of software agents to engage in "*Arguing from Experience*" dialogues. PISA will address the issues related to multi-party dialogues, and will apply a unique treatment to these issues in ways appropriate to "*Arguing from Experience*".

# Chapter 6: Multiparty Arguing from Experience - The PISA Framework

Chapter 3 provided a theory to enable "*Argument from Experience*", by real time association rule mining, conducted by agents to find reasons to support their viewpoints and critique the arguments of the other parties in a dialogue, the aim of which was to come to a decision in relation to a case from some domain. A foundation for a generic dialogue game protocol to facilitate dialogues involving such arguments from experience was also presented. This chapter describes the *PISA* (*Pooling Information from Several Agents*) *Framework,* intended to allow any number of participants (presented by software agents) to engage in "*Arguing from Experience*" dialogues. This is particularly beneficial when there are more than two possible "*views"* (classifications, opinions, etc), since each possible "*view"* can then have its own champion.

Multiparty dialogues, of this style, raise a number of significant issues, necessitating appropriate design choices. To date research into persuasive argumentation dialogues have largely been confined to scenarios with two agents. Very few previous examples of dialogue with several agents can be found in the literature. Section 6.1 gives a discussion of these dialogue systems along with the main issues regarding multiparty dialogues in general as identified in (Dignum and Vreeswijk, 2004) and (Traum, 2004). Section 6.2 discusses how these issues were addressed in the PISA Framework. Sections 6.3 and 6.4 describe the basic elements of PISA, namely the Argumentation Tree data structure, which presents the arguments exchanged in PISA and the attack relations amongst them, and a "*chairperson*" agent which function is the facilitate the progress of the dialogue. Section 6.5 gives a description of an implementation of PISA and Section 6.6 concludes with a summary.

## 6.1. Issues in Multiparty Arguing from Experience

The focus of argumentation dialogues in works on AI has largely been limited to two-party dialogues. Typically these dialogues have been adversarial (see (Prakken, 2008, 2006) for surveys of some of these systems). In practice, however, such dialogues can take place with more than two participants. For some (classification) problems there may be a set of possible answers, and it is desirable to allow each possibility to have its own advocate, or group of advocates, so that each possible answer can be given fair consideration. By allowing several parties to all take part in the dialogue they can pool their experience, increasing the chances of the correct solution being reached. Moreover, where a group of agents advocates a position their joint pooling of experience will further tend, when the case merits it, to a correct solution. Examples are numerous. In debates within coalition governments, such as in the Nordic or Benelux countries, representatives of different national parties may engage with governmental policies from various angles, with each trying to push forward the agenda of their own party, decreasing the bias that results from a single perspective. In medicine, several diseases may share similar symptoms. Hence doctors may disagree with one with the other and only by consulting and arguing with each other can the right diagnosis emerge. In academic assessment and peer review it is normal to allow several people to debate the correct outcome with each other, often discovering strengths or weaknesses that one of them had missed. These real life examples not only exemplify the frequency and importance of multiparty argumentation dialogues, but also highlight several issues that must be taken into consideration when trying to design multiparty dialogue systems in general and argumentation ones in particular.

There are several different models for multiparty dialogues: formal meetings, informal meetings, bulletin boards, seminars and brainstorming sessions and so on. Some of the most important issues arising from this variety of models are discussed in (Dignum and Vreeswijk, 2004). For each of these issues, choices must be made to yield a particular flavour of dialogue. Note that these choices are not between something right and something wrong, but between something appropriate or inappropriate for a particular purpose. Thus, the issues must be resolved in the light of a particular goal that the dialogue is intended to serve. Some work has been done regarding n-person argumentation games. Pham et al. (2008) represent a defeasible logic approach to model such games. The approach advocated here addresses situations requiring agents to settle on a common goal despite the fact that their agendas may contain conflicting goals, if the individual preferences are not sufficient to solve the conflict. This group of agents applies the majority rule to identify the "*most common*" claim. This approach is argued to simplify the complexity of n-person argumentation games into two–group games: one supports the major claim, the other opposes it. This sort of game, while interesting, does not cover situations where there is no "*major claim*", but rather each participant has its own claim, and the situation requires consideration of each of these claims as legitimate stand-alone claims. In such cases it is not possible to categorise the players into those supporting the major claim and those opposing it, and so a more sophisticated system that considers the point of view of each individual agent is required.

The main concern regarding scenarios such as the one described above is how to allow for an indefinite number of participants to engage in the debate without jeopardising the generic "*Argument from Experience*" protocol proposed in Chapter3, and at the same time allowing each participant to defend its own thesis. There are a number of issues of relevance in any multiparty dialogue. The following summary is based on the discussions of these issues given in (Dignum and Vreeswijk, 2004) and (Traum, 2004):

- *System openness*: Multiparty "*Arguing from Experience*" can either be closed or open. A *closed dialogue* starts with *N* participants and continues with the same *N* participants until it terminates. Agents are not allowed to join the dialogue once it has started, and those involved in the dialogues cannot leave the dialogue while it is in progress. *Open dialogues* are the opposite: agents are free to join or leave them at any time. Both systems have their *pros and cons*. On one hand closed systems are less complicated to realise, on the other open systems allow for more flexibility.

- *Roles of the participant*: PADUA allows for two possible roles only: *the proponent* and *the opponent* of some claim. However, for the purposes of facilitating multiparty "*Arguing from Experience*", participants, in the underlying dialogues, have to play more variant roles. There may be *several proponents* and *several opponents* of a possible classification of the case under discussion. Alternatively, each of the agents involved in the dialogue may be the champion of their own theory. Moreover, some participants could take a neutral stand - they would have no opinion regarding the debate. A mediator agent aiding the dialogue between other parties is a good example of neutral roles. Linguistically speaking, in the two-party dialogues one (and only one) participant may speak per turn (the *speaker*) while the other listens (the *listener* or the *hearer*, e.g. (Goffman, 1981)). In multiparty dialogues there can be more than one *hearer* per turn. Moreover, arguably there can be more than one *speaker* per turn, since in real life people may start talking at the same time, interrupt or compete with each other for attention. The roles the agents play influence their behaviour. For example, neutral agents favour deliberation dialogues, because they are not committed to any point of view.
- *The chairperson:* Dignum and Vreeswijk (2004) distinguish another category of roles, which they refer to as social roles within the dialogue as exemplified by the role of the chairperson. The chairperson influences the turn taking policy applied within the dialogue, and may have the authority to determine when parties can join and leave the dialogue. The chairperson may also have the power to terminate a dialogue either unilaterally, or through some predetermined protocol.
- *A clear addressing policy*: This policy should tie in with that of linguistic roles. The main question, when designing such policy, is how moves are addressed. Participants can choose whether to address a move to a specific recipient or to several (specified) recipients or just broadcast the message to all other participants. Also, if a number of participants are allowed to exchange moves that other participants are not allowed to "*hear*", then one might argue that messages (moves) that are only *heard* by a subgroup of participants are a separate dialogue between that subgroup. Following the

theory presented in Chapter 3, the promoted addressing policy can take one of two forms, either: *public broadcasting* where all the participants in a dialogue may listen to what the speaker(s) is saying, or *targeted broadcasting* to one or more of the participants, but not all of them.

- *Turn taking*: A further essential policy to determine who is allowed to "*speak*", about what topic, at what time, for how long and in what manner (e.g. (Allwood, 1995)). Also, for persuasion dialogues the decision as to whether all participants are given permission to utter a speech act when they want, or if they have to wait for their designated turns, can have a significant influence on the final outcome of the dialogue.
- *Termination*: In multiparty dialogues termination happens either when all the participants have reached a decision. Alternatively, the participants may fail to reach an agreement. Therefore a mechanism should be applied to terminate these dialogues, rather than allowing the participants to argue indefinitely. In these cases there should also be a mechanism to determine the winner of the game or to accept that ties can take place.

The issues discussed above must be addressed in any system for multiparty dialogues. There are no right or wrong answers: the questions must be resolved appropriately for the particular context. The addressing of these issues required significant developments and improvements to the promoted model for "*Arguing from Experience*" as realized by PADUA described in the foregoing chapters[30]. This presented a significant challenge, because:

- PADUA is a closed system with exactly two players.
- PADIA allows its players very restricted set of roles, by virtue of the simplicity inherent in two player dialogue games.
- PADUA *lacks a powerful control structure*, as it does not require one.

It was therefore essential to resolve the above issues before instantiating the generic protocol introduced in Chapter 3 for multiparty "*Arguing from Experience*" dialogues. The adjustments to be introduced to the structural

[30] These issues, however, are not significant in two-party dialogues.

framework from Chapter 3 should not affect the basic protocol structure: the legal moves, the speech acts and the rule mining, nor the generic framework identified in Section 3.4. Rather the adjustments should complement these elements with a control layer that makes it possible to organise the participants and their turns within the dialogue games, and to identify the termination conditions for those games. Therefore, the type of dialogue intended, in terms of the issues addressed in this section, had first to be identified. This multiparty version will be referred to as PISA (*Pooling Information from Several Agents*).

## 6.2. The PISA Framework

This section provides a detailed account of the *PISA Framework* to enable multiparty "*Arguing from Experience*" dialogues. PISA concerns dialogues where there is a range of options for classification, and each of the participants is the advocate of one of these options. Where there are more agents than opinions, the agents will act in groups[31], one for each opinion. Additionally, there will be one agent, the *chairperson*, who will not be the advocate of any position, but rather manage and facilitate communication between the clashing advocates. This style of dialogue thus determines the roles of its parties: a chairperson, and, for every option, at least one player acting as its advocate. Each group will act as one entity in the ongoing dialogue, thus masking the internal decision making process that is taking place between different players sharing the same objective. The distinction between players and groups will be drawn in Chapter 7. Until then and for the purposes of readability, both individual players and groups of players will be referred to, throughout this chapter, as participants; where necessary a distinction is made between players and groups. Each participant is the defender of its thesis, and an opponent of the rest of the participants. PISA, however, allows its participants to temporarily defend each other where appropriate for strategic reasons[32].

[31] This notion of "*group*" will be discussed in detail in the following chapter.

[32] The issues of strategy will be discussed in length in the next chapter.

The dialogue will be open, in a sense that participants may enter or leave when they wish. For turn taking, a structure with rounds is adopted, rather than a linear structure where a given agent is selected as the next speaker (e.g. the turn taking protocol in (Bel-Enguix and López, 2006) where the current speaker chooses who will speak next). In each round, any participant who can make a legal move may do so. The chairperson then updates a central argument structure, which will be termed the "*Argumentation Tree*", and another round occurs. The central argument structure acts as a co-argumentation artifact as proposed by Oliva et al (2008b). In every round there is a number of *speakers* (participants contributing in that round) and a number of *addressees* (participants which positions are under attack). The rest of the parties (which did not participate and are not attacked in the given round) need to be aware of the developments in the dialogue and are thus assumed to be *passive listeners* (*overhearers*, e.g. (Goffman, 1981)).

There is no limitation on the number of parties that can participate in any round. However, to simplify the game, each participant is limited to one move per round. This turn taking policy gives the participants a rich context to explore strategy issues. It also simplifies the game, allowing participants to skip a (predetermined) number of rounds for strategic reasons. This structure allows participants to place their attacks/counter attacks as soon as seen appropriate, without the need to wait for their turns to contribute. This is not perhaps the most usual structure for human meetings, but it can be found in some board games such as Diplomacy[33]. It is suggested that the structure is particularly appropriate in order to achieve fairness in the situations where every advocate is playing for themselves, and has to regard every other advocate as an opponent (even though they may form temporary coalitions against a particular opponent as in Diplomacy). For addressing, every move after the first move, attacks a move of some other participant and so that particular participant can be regarded as the addressee of that move, and the others as listeners. The game terminates when no participant makes a contribution for two rounds (to ensure that they

[33] A description of this game (very popular in the UK in the 70s and still played) can be found http://en.wikipedia.org/wiki/Diplomacy_(game).

have really finished and not withheld a move for tactical reasons) or after some limiting number of rounds have been played, and thus the termination of the game is guaranteed. The model is essentially that of a facilitated discussion, with the chairperson acting as the facilitator. The realisation of this model and the choices summarised above are considered in the following sub-sections.

### 6.2.1. The Structure of the Control Layer

The study of communication among a number of agents is not new to Multi Agent Systems (MAS). Blackboard systems (e.g. (Hayes and Roth, 1985)) are probably the most generic form of multiparty communication. In blackboard-based coordination, interactions occur by means of shared data "*spaces*" used by agents as common repositories to store and retrieve messages. The most significant advantage of this coordination model is that messages can be left on blackboards without needing to know, neither where the corresponding receivers are nor when they will read the messages. The drawback of this model is that the sender and the receiver have to agree on a common message name to interact. On the other hand, in Tuple spaces-based coordination (e.g. Linda (Ahuja et al., 1986) and (Carriero and Gelernter, 1994)) information, in the shared data spaces (tuple spaces), is organised in tuples. These Tuples can be asserted to and retracted from the shared knowledge base asynchronously by a number of agents. The tuple space can also be used as a global space for agents to reside and interact (e.g. (Omicini and Denti, 2001) and (Bergenti, and Ricci, 2002)). It provides the social environment for the agents. It is also the communication media for the agents to interact with each other. In argumentation, tuple centres (programmable tuple spaces), were used to play the role of agent mediator in (Oliva et al., 2008a) and co-argumentation artifact in (Oliva et al., 2008b). In these works, coordination rules were expressed in terms of tuples of an event driven language over the multi-set of tuples (Doutre et al., 2005) (this latter work presents an implementation of information-seeking dialog based on tuple centre architecture). Also, the tuple centres embody computational capacity, which enables it to issue specific programmable reactions that can influence the interacting agents.

In PISA, the control layer from Chapter 3 is specified in terms of multiparty "*Arguing from Experience*". The suggested structure is illustrated in Figure 6.1 as a "*meeting room"* in which participants can be "*seated*". This structure is equipped with a Tuple-space like structure as central means for communication between the different agents taking part in PISA dialogues.

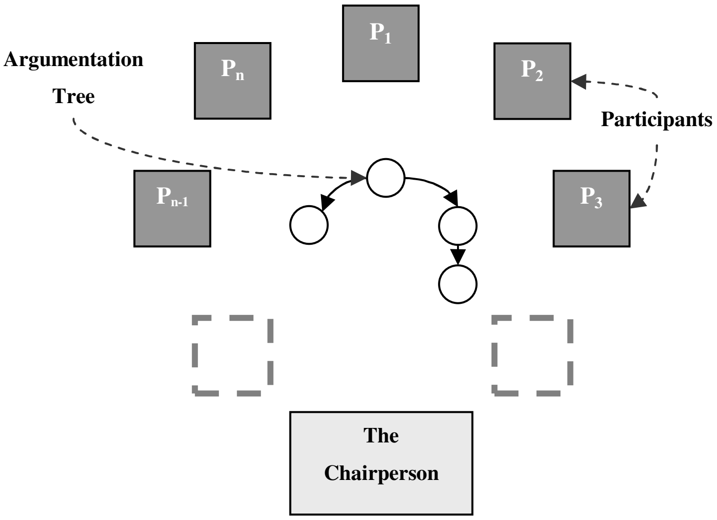


**Figure 6.1. The suggest structure for control layer in PISA (Wardeh et al., 2009b).**

The "*meetings*" taking place within the suggested structure are guided by a dedicated agent, referred to as the "*chairperson*". This agent is responsible for several tasks including: monitoring the participants, controlling the turn taking procedure and enforcing the protocol rules. There is no distinction between the participants other than their opinions ("*views*") regarding the case under discussion. When a new game commences the chairperson randomly chooses one agent ($a \in A$) to start the dialogue. In the meeting room scenario this participant, referred to as $P_1$ is given the first seat at the meeting table; the rest of the participants are seated randomly around the table and given according names ($P_2$ … $P_n$). Then $P_1$ proposes a new rule and pastes it on a black board: this is called the first argumentation round ($R_1$). The game continues in the same manner, and in each of the subsequent rounds all the participants who can and wish to attack any of the previously played arguments are allowed to place their arguments (moves) on the black board. The suggested facilitated discussion scenario enjoys the following advantages:

- *It increases the flexibility of the overall operation of PISA:* By assigning the majority of protocol surveillance to the chairperson the system gains great flexibility with regard to the participating agents. For instance the system can be switched between closed and open by applying a few limited changes to the chairperson, while the rest of the participants remain unaffected.
- *It is a very simple structure*: There is no complicated turn taking procedure involving a choice of the next participant, allowing the internal implementation of the participants to be kept as simple as possible.
- *It provides a fair dialogue environment*: The organisational configuration of the dialogue is neutralised by restricting the control tasks to the chairperson which is not allowed to take sides in the dialogue.

### 6.2.2. Turn Taking and Termination Policies

There is no strict turn taking procedure in PISA: participants who can make legal moves in any round are allowed to participate in this round. Participants will follow the generic algorithm highlighted in Figure 3.1 when proposing rules or responding to moves played by other participants. Also, PISA does not enforce any specific rules on the upper limit of the number of participants that may take part in each round. However, PISA inherits from PADUA the restriction stating that each participant should be limited to one move per round, mainly to simplify the dialogue, and to constrain the growth of the blackboard and the repetition of moves within reasonable bounds. Also, the advocated turn taking policy grants the participants the freedom to apply their strategies in the way they consider fitting to win the game. Enforcing a stricter turn taking policy may have implications on the flow of the underlying dialogues and their consequent results. For example, with tokenised turn taking, one participant might lose a dialogue game just because it was not given the opportunity to present its case at a favourable time. Also, in sequential turn taking policies a participant may not be able to defend itself against some attacks in its own turn, but might be able to do so after some other participants have placed their moves. In general the turn taking technique adopted here enjoys the following advantages:

- It allows participants to skip a limited number of consecutive rounds without taking part in them (for strategic reasons). However, if participants do not contribute for a pre-defined number of rounds they will be discarded from the dialogue.
- It allows participants to place their attacks/counter attacks as soon as they see appropriate with the need to wait for permission to contribute to the ongoing dialogue.
- It presents a solution for situations where participants have to wait to gain permission, presented by the means of special token, before they are allowed to contribute. Also, the advocated turn taking policy gives all the participants an equal chance to win the game.

The chairperson terminates an ongoing PISA game when two rounds have passed and the argument state has not changed: i.e. none of the participants has contributed to the game. The chairperson waits for two rounds to accommodate the cases where some participants have chosen to skip some rounds for strategic reasons. After one round has passed without moves, the second round is considered a last chance for participants (which skipped) to contribute should they wish to prevent the game from ending. This termination policy is called "*legal termination*". However, there are also cases in which the game should be exceptionally terminated ("*exceptional termination"*). The chairperson also terminates the game if only one participant remains active after all the other participants have withdrawn (in which case the "*surviving*" participant wins). Also if the game has taken more than a predefined number of rounds (say n) the chairperson ends the game, assuming that if the parties could not reach an agreement in *n* rounds, where *n* is sufficiently large, then they will never agree. In this case no one wins the game. After terminating the game the chairperson has to determine which agent has won this game, the rules for identifying the winners in PISA are specified in Section 6.3.

### 6.2.3. Roles of the Participants

PISA requires considerable attention to roles, and more importantly the way these roles change from round to round. The main distinction in participants' roles is between *attackers* and *defenders* and between *speakers* and *listeners*:

- *Attacker(s) vs. defender(s):* While participants are most certainly defenders of their advocated *views*, they can take different positions regarding other participants' proposals. Each can decide whether to attack or defend other participants' arguments. Enabling participants to defend the arguments of other participants (supposedly, and in the long term, their opponents) may be of strategic importance within the game. Chapter 7 will return to this point and discuss when defending certain opponents may be desirable.
- *Speaker(s) vs. listener(s) (addressee(s))*: In the first round of PISA there is only one speaker ($P_1$) while the rest of the participants are addressees (the chairperson may be considered as an auditor). In all the subsequent rounds there are *s speakers* where *s* is the number of the participants participating in the given round (and $s <= m$, the number of participants). Once the speakers are done with their moves the addressees of the round are defined as the participants whose arguments were attacked in this round and the rest of the participants (i.e. those who have not participated nor have been attacked in the given round) are considered (passive) listeners.

## 6.3. The Argumentation Tree

The notion of an "*Argumentation Tree*" is used, in PISA, to describe the central data structure representing the arguments exchanged in a dialogue, and the attack relations between those arguments. This tree acts as a mediating artifact for the dialogue as described in (Olivia et al., 2008). The tree structure differs from other argumentation structures used in the literature (e.g. (Hunter, 2006) and (Prakken, 2006)) as it consists of arguments presented by more than one participant. It uses four "*colours*" to mark the status of the arguments played. so far, and two types of links: *explicit* links representing direct attacks, and *implicit* links representing indirect attacks. The issue of addressing is solved via the direct links. A move is addressed to the participant that played the argument attacked by this move except for the first move in the game which is addressed to all the other participants. This type of addressing is a "*public broadcast*" as all the participants can "*listen*" to what "*speakers*" are saying by consulting the tree. Another form of broadcasting, a variation of targeted broadcasting, applied within "*groups"* of players, will be discussed in the following chapter.

The proposed "*Argumentation Tree"* data structure consists of nodes, links and the *Green Confidence*. *Nodes* represent the speech acts made in the game. Recall that "*Arguing from Experience*" has six speech acts corresponding to the AEC2 argumentation scheme and the critical questions associated with this scheme. These speech acts were identified in Section 3.2 and given numbers for readability. These numbers were as follows: Propose New Rule =1, Distinguish = 2, Unwanted Consequence = 3, (Propose) Counter Rule = 4, Increase Confidence = 5 and Withdraw Unwanted Consequences = 6. Each node has a colour: green, blue, red or purple, representing the status of this node (and hence the argument presented by it). It also has a separate field representing: (i) the speaker of the dialogue move; (ii) the confidence of the move; (iii) the round in which the move was played and (iv) an array representing the attacks against this node[34]. *Links* represent the explicit attack relationships between nodes. The *Green Confidence* is a global value associated with the tree representing the highest confidence of the undefeated green node(s).

Nodes are either green or blue when introduced, depending on whether they propose a new association rule (1,4,5,6) or only attempt to undermine an existing one (2, 3). Red nodes are those directly under attack and purple nodes are those indirectly attacked. Nodes change their colour according to Table 6.1.

[34] Note that the first two fields in the node structure represent the move in a compressed manner: any other information related to this move (such as the actual rule that have been played) can be tracked using the "*History Log*" structure (Section 6.4.2).

The first node played in a PISA game is referred to as the "*root*" of the dialogue's "*Argumentation Tree*".

| **Colour** | **Meaning** | **Shifts to** |
|---|---|---|
| **Green** | 1, 4, 5 or 6 move node, undefeated in the given round. | **To red**: If attacked by at least one undefeated node.<br>**To purple** If indirectly attacked by an undefeated green node with higher confidence. |
| **Red** | The node is defeated in the given round. | **To green**: If all attacks against it are successfully defeated and the original node colour was green.<br>**To blue**: If all attacks against it are successfully defeated and the original node colour was blue. |
| **Blue** | 2 or 3 move node undefeated in the given round. | **To red**: If attacked by at least one undefeated node. |
| **Purple** | 1, 4, 5 or 6 move node indirectly attacked by a higher confidence green node, played by a different participant. | **To green**: If all attacks against it are successfully defeated, and if the move(s) indirectly attacking this node was defeated.<br>**To red**: If attacked by at least one undefeated node. |

**Table 6.1. The colours used in the Argumentation Tree (Wardeh et al., 2009b).**

When participant *Pi* plays some dialogue move (*dm*), it must satisfy a number of conditions in order to be added as a node to the Argumentation Tree, otherwise it will be rejected. The conditions of acceptance are as follows:

- *dm* is added to the tree if and only if it changes the colouring of the tree. In consequence participants are not allowed to attack, for instance, red nodes (defeated moves), as these attacks will not change the colouring of the red node, nor that of the branch of the Argumentation Tree in which the red node is located. Note, however, that participants may attack purple nodes, as direct attacks against purple nodes will change their colouring to red.
- *dm* explicitly attack the move it is associated with (parent node).
- A participant can put forward one move only per round (deciding which rule to play is strategy issue).
- Moves 1, 4, 5 and 6 implicitly attack all other 1, 4, 5 and 6 moves played by other participants which have content with lower confidence.
- Moves 2 and 3 affect only the nodes they directly attack.
- Participants should not play moves that weaken their position, such that another participant would take the lead. This condition holds when a

participant tries to attack blue node that was originally made to attack an argument proposed by other participants, unless this move changes the colouring of that argument to purple.

Once a game has terminated, the chairperson consults the Argumentation Tree to determine the winner. The winner should satisfy one of the following rules:

- **Rule1:** If all the green nodes belong to the same participant, that participant is the winner. This condition is realised only when no other participant has played an undefeated move with higher or similar confidence.
- **Rule2:** If there are no green nodes, and all the blue nodes were played by the same participant, that participant wins.

### 6.3.1. Winner Announcement

It is not always the case that the dialogue games conducted within PISA result in a clear winner. There are two scenarios:

- Upon the termination of the game, there may be two or more green nodes with the same confidence, each belonging to a different participant. This situation may occur if the confidence value of these nodes are the highest (indirectly attacking all the other potentially green nodes), or if all the other nodes with higher confidence values are defeated.
- The Argumentation Tree may not contain any green moves at the end of the game. For instance, because all the green moves have been defeated in the course of the game. Additionally, all the (undefeated) blue nodes were played by a number of different participants.

The first case is considered a *"strong tie situation"*, as the participants have actually proposed their opinions within the game. One possible solution is to enforce a new game involving the tying parties only and see how this game develops. However, there is no guarantee that this game will not also lead to another tie. In this case the chairperson will be forced to announce a tie (after the second game or after *predefined* number of games with the tying parties from previous ones). The second case is seen as a *"weak tie situation"*, as the tied participants did not actually have any proposed classifications at the end of the game. In such cases enforcing a second game may be of great benefit, but with the requirement that the participants should propose as many reasons for their classifications as they can this time.

### 6.3.2. The Argumentation Tree Basic Functions

The Argumentation Tree grows and changes in each round of the dialogue as new leafs presenting the attacks and counter attacks played in the current round are added to the structure. Three basic functions are identified to maintain this tree throughout the dialogue:

- **Adding a new node to the Argumentation Tree**, representing dialogue move *dm* played by participant *P* at some round *R*.
- **Pruning the Argumentation Tree**: to remove all the nodes played by one participant *P* from the tree once this player leaves the game for any of the reasons listed previously.
- **Colouring the Argumentation Tree**. Once a new node is added or after pruning the tree a "*re-colouring process*" of the Argumentation Tree is triggered following the colouring scheme given above.

For the purposes of realising the above functions, three algorithms were designed. The first algorithm ($A6_1$) adds new leaves to the current "*Argumentation Tree*". Figure 6.2 illustrates the pseudo code for this algorithm.

```
Input: Argumentation Tree ArgT, dialogue move dm, round R.

node = create_new_node(dm)
if R = 1 then
    Set ArgT.Root = $node_1$
    Set ArgT.Root.colour = green
else
    if ∃ leaf ∈ leaf_nodes(ArgT): participant(leaf)=target(dm) then
        update_attack(leaf, node)
apply colouring algorithm on ArgT.
```

**Figure 6.2. Algorithm $A6_1$ – Add new node to the Argumentation Tree.**

- The algorithm in Figure 6.2 uses the following functions:
- create_new_node(dialogue move) returns a new node representing the input.
- leaf_nodes(Argumentation Tree) returns the leaf nodes of the input tree.
- participant (node) returns the *speaker* of the move represented by the input.
- update_attacks(node, node) updates the attack array of the parent node to include the new node.

The second algorithm (A6$_2$) implements the pruning function and is illustrated in Figure 6.3. The algorithm serves to removes any nodes, representing some participant P, when this participant leaves the ongoing PISA dialogue game.

```
Input: Argumentation Tree ArgT, Participant P, round R.

delete_leaf_nodes(ArgT, p).
Traverse ArgT bottom up
 for each level l ∈ [R -1, 1[ do
  for each node n ∈ node(p,l) do
   delete_blue_children(ArgT, n).
  for each node gc ∈ green_children(n) do
   if ∃ n2 ∈ node(parent(gc)): original_colour(n2) = green and
   round(n2) <l and confidence(n2)> confidence(gc) then
    delete_node(gc)
   else
    if participant(parent(n))= participant (gc) then
     substitute parent(n) with gc.
    else
     update_links(gc) such that gc attacks parent(n).
   delete_node(ArtT, n).
  if participant(ArgT.Root)=p then
   if children(ArgT.Root, colour=green) ≠ ∅ then
    substitute the root with the direct child with the higher
    confidence then delete the old root.
   else
    substitute ArgT with the sub tree rooted under the highest
    confidence node in ArgT.
```

**Figure 6.3. Algorithm A6$_2$ – Pruning the Argumentation Tree.**

This algorithm makes use of the following functions:

- node (participant , round) returns the nodes played by the input in the given round.
- delete_blue_children (Argumentation Tree, node) deletes all the nodes attacking *the input* from *the Argumentation Tree* if their original colour is blue, also deletes the sub trees with these nodes as root.
- green_children(node) returns all the nodes attacking the input such that the original colour of these nodes is green.
- delete_node(node) deletes the input node and the sub tree it roots.

The third and last algorithm ($A6_3$) (re-)colours the parts of the "*Argumentation Tree"* influenced by the application of either the addition or the pruning algorithms described above. Figure 6.4 presents a pseudo code to realise this algorithm. In this figure, nodes(Argumentation Tree, round) returns the nodes added to the Argumentation Tree at the given round.

### 6.3.3. Fictional Example of the Argumentation Tree – Example (1)

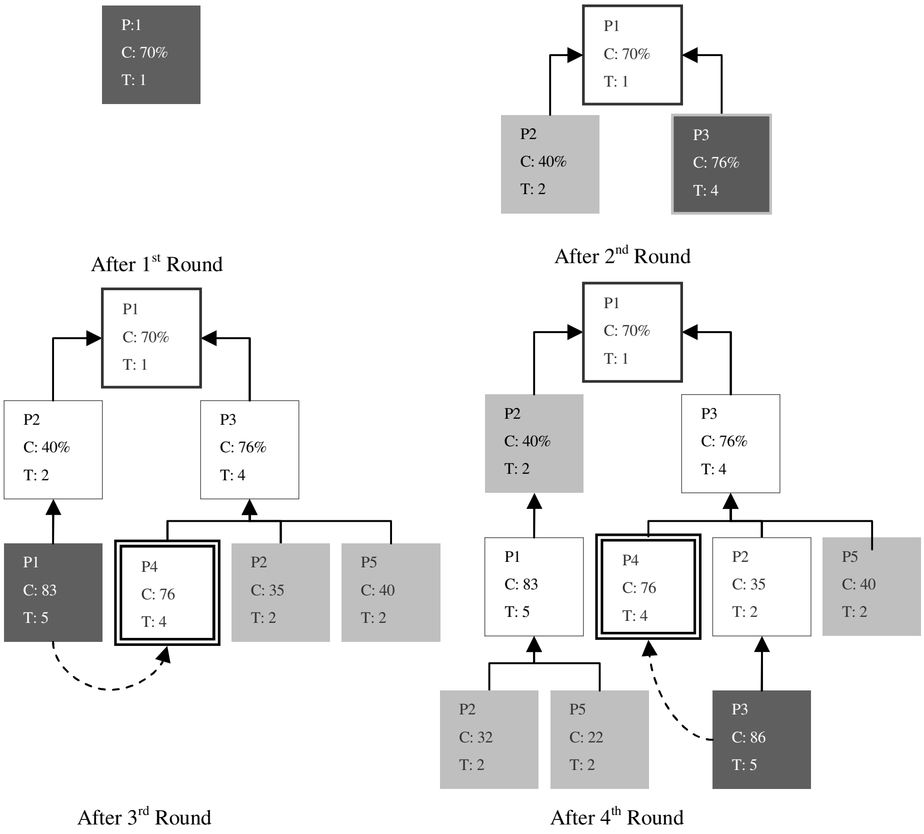
This sub-section provides an example to summarise the key ideas in relation to the Argumentation Tree. This summary is intended to provide a reference to the operation of the Argumentation Tree, in particular, the three functions identified above. Figure 6.5 illustrates the discussed example. In this figure the dotted lines present indirect attacks, solid lines present direct attacks. Each tree node has three entries: P (the participant), C (confidence of the rule) and T (the type of the move). The example follows a fictitious scenario in which five participants have joined a PISA dialogue game to consider the classification of some case.

```
Input: Argumentation Tree (ArgT), round R.

if R=1 then
 Set ArgT.Root.colour = green
 Set ArgT.greenConfidence = confidence(ArgT.Root).
else
 for each node $n_R$ ∈ nodes(ArgT,R) do
  Set $n_R$.colour = blue or green depending on the attack type.
  if confidence($n_R$)>ArgT.greenConfidence then
   Set ArgT.greenConfidence = confidence($n_R$).
  for each node $n_{R-1}$ ∈ nodes(ArgT,R-1) do
   Set the colour of the node to red if it is attacked
  Traverse ArgT bottom up
   for each level $l$ ∈ [R -2, 1] do
    for each node $n_l$ ∈ nodes(ArgT,l) do
     if all the node's children are red then change this node
     colour back to blue or green depending on the attack.
     if at least one of the children nodes is blue or green
     change the colour of the parent node to red.
     if all the node's children are red then change this node
     colour back to blue or green depending on the attack
```

**Figure 6.4. Algorithm A6$_3$ – the colouring Algorithm.**

The details of this case, or the description of the participants, including their proposed classifications of the case, are not essential for the purposes of illuminating the operation of the Argumentation Tree. It is also assumed that the dialogue among the five participants continues for four rounds only. Note that by the end of the fourth round participant (P$_3$) has achieved a winning position: the only green node belongs to this participant.



After 1st Round

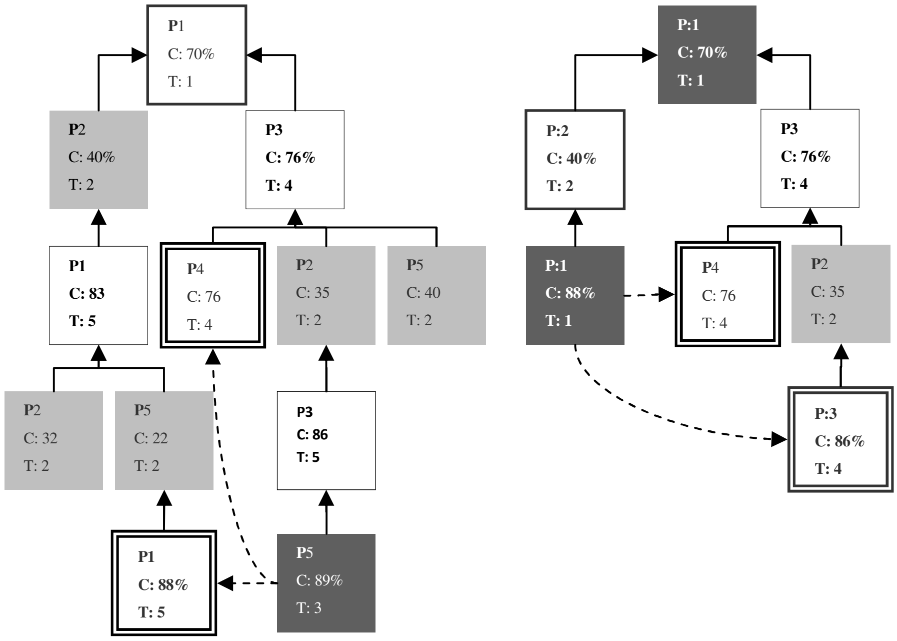
After 2nd Round

After 3rd Round

After 4th Round

**Figure 6.5. Example (1) – The progression of the Argumentation Tree.** *Dark Grey=Green nodes, Double Lined=Purple nodes, and Single Lined=Red nodes.*

Let us now assume that the dialogue continues for another round, after which $P_5$ decides to leave the game, therefore all the moves this participant has played should consequently be removed from the Argumentation Tree. Figure 6.6 illustrates the structure of the Argumentation Tree before and after applying the pruning algorithm ($A6_2$). It is worth noting, that the participant in the current winning position has changed from $P_5$ before pruning to $P_1$ after pruning. This concludes the discussion of the Argumentation Tree data structure. The following section returns to the chairperson to describe in details the structure of this facilitator agent.



Original Tree

Pruned Tree.

**Figure 6.6. Pruning P5 from the Argumentation Tree of Example (1).** *Dark Grey=Green nodes, Double Lined=Purple nodes, and Single Lined=Red nodes.*

## 6.4. The Chairperson

The basic entity in the control structure outlined previously in Section 6.2 is a neutral agent referred to as the "*chairperson*", which administers the various tasks assigned to this layer. In the following details of how this agent is realised are presented. This mediator agent resembles the mediator artefact suggested by Oliva et al (2008a). The responsibilities of the chairperson concern the facilitation of an "*Arguing from Experience*" dialogue among any number of agents to reach some sort of decision about the classification of given case. The chairperson also ensures that the decision is reached within a limited number of rounds; thus preventing infinite dialogues from taking place in the promoted framework. The responsibilities of the "*chairperson*" are summarised as follows:

- Starting a dialogue involving a set of participants to classify a given case.

- Making decisions regarding agents requesting to join or to withdraw.
- Monitoring the dialogue. This involves registering, for each played round, which agents have taken part and which have not.
- Performing the three Argumentation Tree functions discussed above.
- Terminate the dialogue game, once a termination condition is satisfied.
- Announcing the game's winner through consultation of the Argumentation Tree. In case of ties, the chairperson initiates the appropriate course of action to resolve the tie situation.
- Exclude (remove) participants from the game upon failing to contribute in the game for a predetermined number of rounds.

One of the interesting questions related to multiparty argumentation is whether participants are allowed to repeat any dialogue moves or not. This issue was considered earlier in this thesis in the context of two-party dialogue. Recall from Chapter 4 that PADUA forbids players from proposing the same rule twice. This simple restriction is not adequate for multiparty "*Arguing from Experience*", as here two or more participants may consider playing the same move, or playing different moves with the same content. This may happen when a number of participants coincidentally attack a particular previous move using the same attack and/or the same content, in the same round. Also, one participant, or indeed a number of participants, may use similar moves to attack different positions on the Argumentation Tree at different rounds. A careful consideration of the different aspects of the generic protocol described in Chapter 3, and its multiparty adaption, is essential to identifying situations where repeating arguments could be tolerated. Such consideration raises a number of questions:

- **Q1**: Could a participant play similar moves against the same opponent?
- **Q2**: Could a participant play similar moves against different opponents?
- **Q3**: Could different participants play similar moves against one opponent?

Two moves (or attacks) are considered similar if they have the same speech act and similar content. The latter condition applies if the association rules of both moves are identical (same premises, consequents and confidence), or if both moves have the same premises and consequents but different confidence values.

To answer the above questions, two guiding principles should be kept in mind. The first is that no participant is allowed to repeat the same move against the same opponent, if this move introduces a new rule, or in PISA terms, could be represented by a green node on the Argumentation Tree (Q1). The second principle is that the formation of endless loops in the dialogue must be avoided. Taking these two principles into consideration, an additional set of three rules could be triggered in repetition situations:

- One Participant cannot repeat the same attacking move (with the same AR) against **different** opponents (Q2) if:
  - This attack is either a distinguishing or unwanted consequences attack.
  - Or, if all of the other previously played moves using this attack are still green (undefeated) on the Argumentation Tree.
- Participants cannot attack their opponents using moves that have already been played against and defeated by these opponents (Q1).
- If two or more participants have coincidentally attacked the same opponent, in the same round, using similar attacks (as identified above). If the confidence is equal in all of these attacks, then the participant under these attacks is required to defend its proposal against them once only (Q3). Otherwise the chairperson chooses the attack with the highest confidence (lowest confidence in case of distinguishing) and discards the rest.

In order to apply the above set of rules, the chairperson is equipped with a data structure, "*History Log*", to store information about the moves played in each round in a way that facilitates a quick verification of each new move against all the other moves played so far in the dialogue, so that any move failing to satisfy the above rules could be excluded instantly from the game. The *History Log* stores a summary of the moves played, so far, in the PISA dialogue. One of the requirements for this data structure was to accommodate for (potentially) substantial number of moves. Computational efficiency (update and lookup speed) was therefore an important factor in this structure. Comparing new moves against old ones needs to be fast, otherwise the performance of PISA will be hampered. Taking this into consideration, in order to look up moves/rules in an efficient manner, the *History Log* groups the dialogue moves in a number of lists; each list containing the moves played by one source against one target. The moves, indexed by both target and source, are stored in a compressed form. Entry (0, 0) in this structure contains a special move: the first move in the dialogue, as this initial move does not represent any attack or respond to any previous move. When a new dialogue move is received the chairperson has to check this move against the non-repetition conditions discussed previously. If the move successfully passes the test then the chairperson adds it to the proper entry in the History Log, and to the Argumentation Tree. Otherwise the move is discarded and the participant is warned that it has failed to play a legal move.

## 6.5. Summary of Multiparty Arguing from Experience

Having described the basic elements in the PISA Framework, a summary of the key changes that have been made to the "*Arguing from Experience*" framework outlined in Section 3.4 is now presented. The discussions given, thus far, make it clear that changes are required to the original model introduced in Chapter 3:

- Recall from Chapter 3 that each agent (player) $a \in A$ was defined as $a = <name_a, C_a, \Sigma_a, CS_a, S_a>$. For the purposes of PISA this definition included a view of the Argumentation Tree. The implementation of this view depended on the agents' strategies: it could take into consideration only the moves that have been added to this structure in the last round; or a more sophisticated view covering the entire tree, or anything in between. The details of these views are discussed in Chapter 7. For now it suffices to mention that the definition of agents taking part in PISA dialogues is given as $a_{PISA} = <name_a, C_a, \Sigma_a, CS_a, V_a, S_a>$; where $V_a$ represents the agent's view of the underlying Argumentation Tree structure. The chairperson agent is identified as: *chairperson* = <*" chairperson", ∅, ∅, ∅, ∅*, ∅>.
- The control layer was extended to include the chairperson as an independent neutral agent responsible for monitoring PISA games and

constructing the Argumentation Tree. This layer is defined in the terms of the following:

- ArgT: the argumentation tree, and is identified as follows ArgT: <Nodes, Attack>. Where Nodes = $\{n_0, .., n_m\}$ is the nodes of the argumentation tree such that $max(|Nodes\ |)$ < predefined threshold. Attack is the attack relation between the tree's nodes, as represented by the direct and indirect attacks identified previously.
- $G_{tie}$: set of PISA dialogue games to resolve ties. Such that $|\ G_{tie}\ | \leq$ predefined threshold.
- $g_{initial}$: PISA initial dialogue game.
- *start*: a function that begins a certain PISA dialogue game, start($g_{tie} \in G_{tie}$) begins a tie resolution dialogue, *start*($g_{initial}$) begins the initial one.

- The "*outcome rules"* of PISA defined, for each dialogue *d* and instance $\varphi$, the winners and losers of *d* with respect to instance$\varphi$. Sub-section 6.3.1 discussed the winner identification in PISA. Formally speaking, the set of outcome rules O reflects that discussion as follows:
  - Winners: $w_{\varphi}(d, A_w \subseteq A)$ = true if $\forall a_w \in A_w$ then $Gaw \in$ O $(d, \varphi)$.
  - Losers: $l_{\varphi}(d , A_l \subseteq A)$ = true if $\forall a_l \in A_l$ then G$al \notin$.O $(d, \varphi)$
  - $O\ (d \in D, \varphi) = \{o_1,..,o_{|A|}\}$: $\forall o \in O$ then $o \in \cup\ (G_a : a \in A)$, and
    - $o \in$ consequences(AR($N_G$)): $N_G \subseteq$Node(ArgT) and the colour of each node from $N_G$ = Green, or
    - $o \in G_{dominantBlue}$ such that dominantBlue$\in$ A and $\forall$ n $\in$Node(ArgT): The colour of n = blue then Participant(n) = dominantBlue
  - If $|\ w_{\varphi}\ |>1$ then there is a tie between the agents in $A_w$.
  - The two functions $w_{\varphi}$ and $l_{\varphi}$ satisfy the following conditions:
    - $w_{\varphi}(d, A) \cap l_{\varphi}(d, A) = \varnothing$.
    - if $|A| = 2$, then $w_{\varphi}$ (d, A) and $l_{\varphi}$ (d, A) are at most singletons.

PISA could incorporate two different sub-models of dialogues: "*Dispute model for Arguing from Experience*" and "*Dissents model for Arguing from Experience*". Both were described in Sub-section 3.2.2.

In the first model, *"disputes"*, all parties engaged in the dialogue have positive burden of the proof. Thus they can win the dialogue by, and only by, proving that the case under discussion classifies according to the class they promote. In the second model, "*dissents*", the burden of the proof lays with the first participant, which initiates the dialogue. The other participants can attack the first participant's proposals in any manner, and can win the dialogue by simply undermining the first participant's position. The first participant is identified as follows $a_{\text{first}} \in A$ and $a_{\text{first}} = speaker(d_{m1})$ : $d_{m1}$ is the first rule in the current dialogue $d_{current}$. The winners and losers functions rules are re-identified:

- Winners (dissents): $w_{\varphi}\,(d, a_w \subseteq A) = \text{true if } aw = a_1$ and $Gaw \in \text{O}\,(d, \varphi)$. Or, $w_{\varphi}\,\text{d}, (A_w \subseteq A) = \text{true if } a_1 \notin Aw$ and $\text{G}_{\text{a1}} \notin \text{O}\,(d, \varphi)$.
- Losers (dissents): $\text{l}_{\varphi}\,(d, A_l \subseteq A) = \text{true if a}_1 \notin \text{A}_{\text{l}}$ and $G_{a1} \in \text{O}\,(d, \varphi)$ such that $\text{A}_{\text{l}} = \text{A} - \text{a}_1$. Or $\text{l}_{\varphi}\,(d, \text{a}_1 \in \text{A}) = \text{true if } G_{a1} \notin \text{O}\,(\text{d}, \varphi)$.

However, the work presented in this thesis focuses on the first sub-model. This is because dissents lead to different flavour of dialogues than the one intended in this thesis. A discussion of dissents will be given in the conclusions chapter.

## 6.6. The Implementation of PISA and Example Dialogues

This section presents an overview of the implementation of the PISA Framework, along with an associated GUI interface. The implementation combines the various components described previously into a functional application, using the Java programming language. The objective of this implementation was to provide a tool to produce multiparty "*Arguing from Experience*" dialogues. The accompanying GUI enables the user to test and assess the resulting dialogues, and can also be used to examine the various components of PISA, such as the Argumentation Tree and the History Log. The implementation described here was used to evaluate a variety of test scenarios and examples discussed in the remainder of this chapter, and the next chapter. Some further specialised components for this application will be included in the next chapter. A more detailed description of the implemented PISA Application, and the accompanying design documentation, can be found in Appendix B.

The PISA implementation, described here, allows for dialogues among a number of participants to be undertaken. An embedded chairperson agent monitors these dialogues and ensures that the participants follow the protocol rules. The GUI interface enables the user to import a *game dictionary* file (Section 4.1). The software then provides the user with options to add at least one participant per every possible classification given in the game dictionary. Chapter 7 will discuss the concept of groups in PISA: here, it is assumed that there exists exactly one *player* for every possible classification. When adding a new *player* the user should load a background dataset for this player. The user also has the option to change the *confidence/support* values and the strategy configuration[35] for new players. The GUI has a special display area dedicated to the participants taking part in the current dialogue, which the user can consult for information about the participants taking part in the dialogue (e.g. data files and strategies). After specifying the participating agents, the user chooses a case to argue about. Once the case is loaded the dialogue can commence. The outcome of which, along with the actual dialogue is printed to a special tab screen. This tab screen includes other options such as displaying the History Log and the Argumentation Tree. The presented implementation of PISA does not cover open-dialogues scenarios, where participants are allowed to join an ongoing game whenever they like. However, the software allows participants to leave at any point of time. The chairperson can also discard "*idle*" participants, which have not taken part in *Skip_Threshold* number of the rounds. The user can decide on the value of this variable prior to the start of the game, otherwise the software gives it the default value of two rounds. When a new dialogue commences the chairperson shuffles the participants and then $P_1$ opens the dialogue by proposing a new rule. If $P_1$ fails to propose a new rule, the "*chairperson*" requests from the other participants, in order, to propose a new rule. This is important to prevent the game from terminating prematurely because the first participant has failed to propose an initial argument.

[35] The issue of strategy in PISA is discussed in details in the next chapter.

However, if all the participants fail to propose an opening rule the game terminates with failure. Once the initial rule is proposed, the chairperson updates the *History Log* repository with this move. A new Argumentation Tree is then instantiated with a root node representing this move. From the second round, onwards, the participants place their moves in the style discussed previously. The chairperson terminates the game once two rounds have passed without any changes to the Argumentation Tree. Illegal moves as not considered a valid reason to keep the game going. Once the game is terminated the chairperson consults the tree to determine the winner. In the case of ties (strong or weak), the chairperson instantiates another dialogue between the tied parties, should the user wish to do so. Otherwise, the dialogue, along with the resulting classification, is printed to the output screen of the GUI interface. The dialogue representation is very similar to the one used in the PADUA GUI application.

### 6.6.1. PISA Framework Application Example (1): The Housing Benefit Example

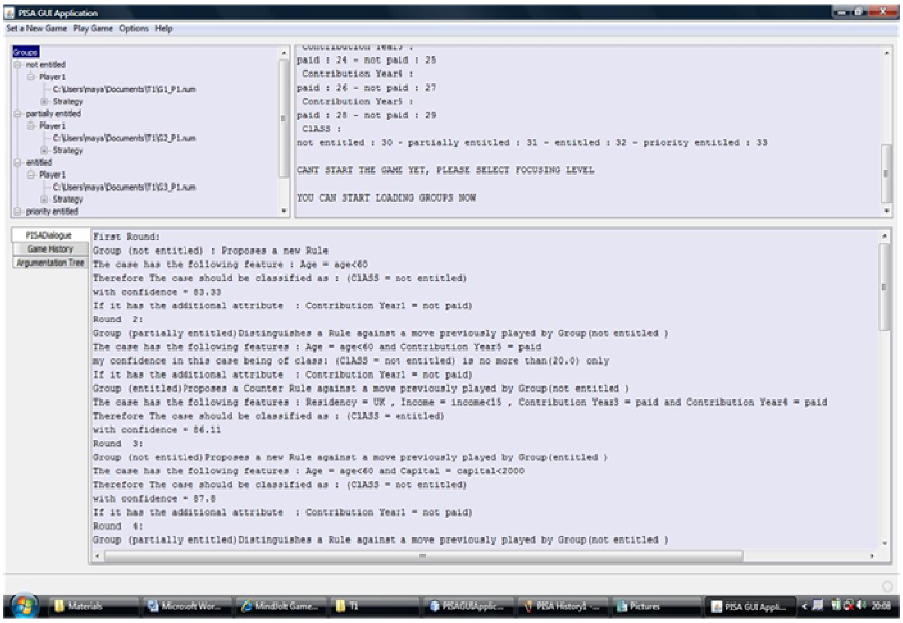
An example is now provided to demonstrate the operation of the PISA Framework and the style of dialogues produced. In this example, PISA is applied to a variation of the RPHA scenario (Sub-section 4.1.2) reinterpreted so that the number of classes was increased from two classes; entitled or not entitled: to four; entitled, entitled with priority, partially entitled and not entitled. The conditions for each of the four classes to apply were defined as follows:

- *(Fully) Entitled*: Candidates entitle to full housing benefit allowance if they satisfy all the RPHA five conditions (Sub-section 4.1.2).
- *Entitled with Priority*: Candidates entitle to housing benefit allowance with priority if they satisfy the entitling conditions and also satisfy one of the following: (i) they have paid contributions in four out of the last five years and either have less capital than the original limit (this is interpreted as £1000 less than the original limit) or have has less income than the original limit (by 5%), or (ii) they are member of the armed forces and have paid contributions in five out of the last five years.

- *Partially Entitled*: Candidates entitle to a lower rate of benefit if they satisfy the age condition, and they either: (i) have slightly more capital than the original limit (+£1000 more than the original limit), but have paid contributions in 4 (or 5) years out of the last five, (ii) have slightly more available income ( +5%) than the original limit, but have paid contributions in 4 (5) years out of the last five, (iii) are employed in the Merchant Navy and have paid contributions in five out of the last five years.
- *Not Entitled*: The candidate fails to satisfy any of the above.

Assume that there are four different offices providing RPHA services in four different regions, each has a dataset of 6,000 benefit records. Each dataset was assigned to a PISA Player Agent. Thus a total of four Players Agents would engage in dialogues regarding the classification of RPHA applicants, each player defending one of the four possible classifications described above. Support and confidence thresholds of 1% and 50%, respectively, were used when mining ARs. PISA was then applied to the case of a male applicant, aged around 55 years, who was a UK resident whose capital (less than £3000) and income falls in the right range (less than 15%), and who has paid contributions in three out of the last five years. According to the conditions listed above this applicant should not be awarded any benefit, since he fails on the age condition. Figures 6.7(a) and 6.7(b) show how the example dialogue produced by the PISA reaches this conclusion. A more detailed account of how this example was produced can be found in Appendix B. This example shows how the PISA can be used to construct meaningful dialogues explaining the reason behind assigning a classification to each input case. Just like in the PADUA application, no intervention, on the behalf of the user, is necessary beyond the input activities. After the dialogue game has finished, the user can inspect the resulting dialogue, and decide whether another run of the application using different input parameters (changing the support/confidence values) is necessary. To help users assess the quality of the dialogues produced, the GUI provides plenty of easy-to-access output data, in addition to the actual dialogue, including: a graphical representation of the Argumentation Tree at the end of each dialogue (Figure 6.8), and a textual description of the moves stored in the

History Log. A partial view of this representation is given in Figure 6.9. These additional pieces of information were targeted at users interested in closely examining the structure of the PISA Framework, and the quality of the dialogue produced.



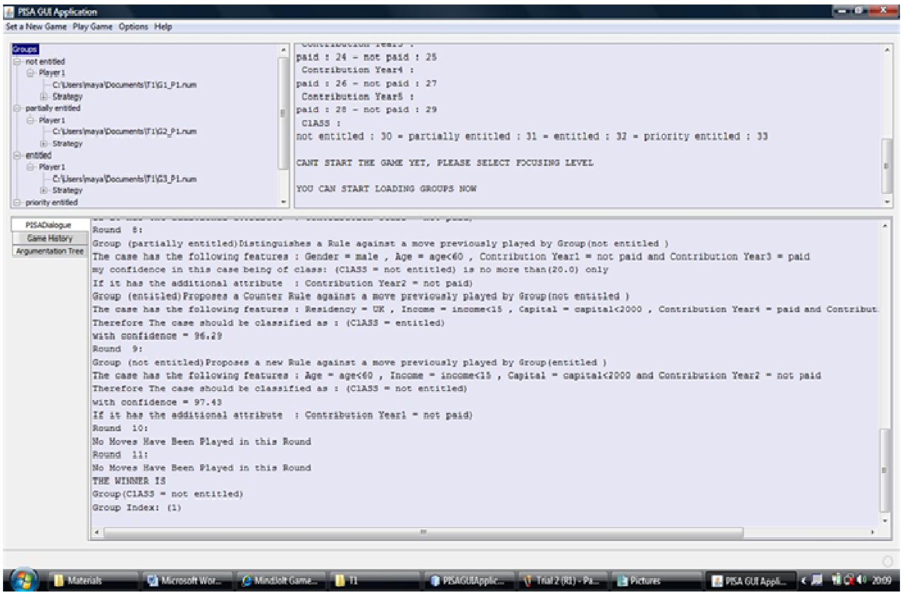
```
paid : 24 - not paid : 25
 Contribution Year4 :
paid : 26 - not paid : 27
 Contribution Year5 :
paid : 28 - not paid : 29
 ClASS :
not entitled : 30 - partially entitled : 31 - entitled : 32 - priority entitled : 33

CANT START THE GAME YET, PLEASE SELECT FOCUSING LEVEL

YOU CAN START LOADING GROUPS NOW
```

```
First Round:
Group (not entitled) : Proposes a new Rule
The case has the following feature : Age = age<60
Therefore The case should be classified as : (ClASS = not entitled)
with confidence = 83.33
If it has the additional attribute  : Contribution Year1 = not paid)
Round  2:
Group (partially entitled)Distinguishes a Rule against a move previously played by Group(not entitled )
The case has the following features : Age = age<60 and Contribution Year5 = paid
my confidence in this case being of class: (ClASS = not entitled) is no more than(20.0) only
If it has the additional attribute  : Contribution Year1 = not paid)
Group (entitled)Proposes a Counter Rule against a move previously played by Group(not entitled )
The case has the following features : Residency = UK , Income = income<15 , Contribution Year3 = paid and Contribution Year4 = paid
Therefore The case should be classified as : (ClASS = entitled)
with confidence = 86.11
Round  3:
Group (not entitled)Proposes a new Rule against a move previously played by Group(entitled )
The case has the following features : Age = age<60 and Capital = capital<2000
Therefore The case should be classified as : (ClASS = not entitled)
with confidence = 87.8
If it has the additional attribute  : Contribution Year1 = not paid)
Round  4:
Group (partially entitled)Distinguishes a Rule against a move previously played by Group(not entitled )
```

**Figure 6.7(a) . The Housing Benefit dialogue.**



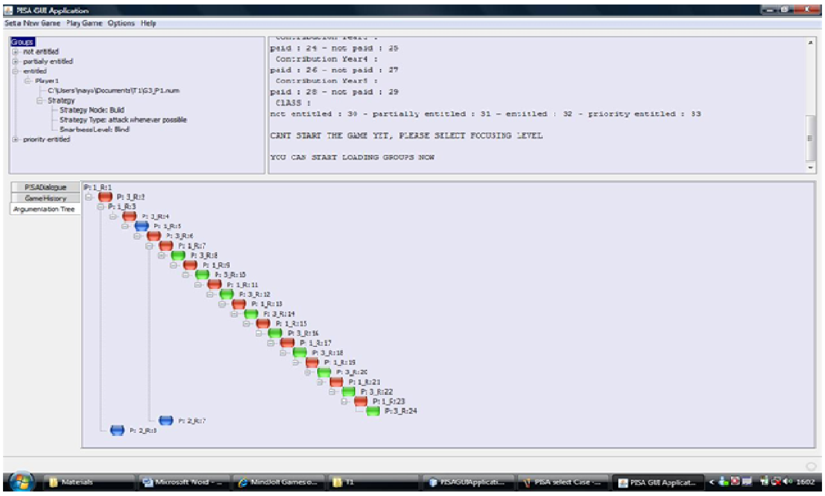
```
paid : 24 - not paid : 25
 Contribution Year4 :
paid : 26 - not paid : 27
 Contribution Year5 :
paid : 28 - not paid : 29
 ClASS :
not entitled : 30 - partially entitled : 31 - entitled : 32 - priority entitled : 33

CANT START THE GAME YET, PLEASE SELECT FOCUSING LEVEL

YOU CAN START LOADING GROUPS NOW
```

```
Round  8:
Group (partially entitled)Distinguishes a Rule against a move previously played by Group(not entitled )
The case has the following features : Gender = male , Age = age<60 , Contribution Year1 = not paid and Contribution Year3 = paid
my confidence in this case being of class: (ClASS = not entitled) is no more than(20.0) only
If it has the additional attribute  : Contribution Year2 = not paid)
Group (entitled)Proposes a Counter Rule against a move previously played by Group(not entitled )
The case has the following features : Residency = UK , Income = income<15 , Capital = capital<2000 , Contribution Year4 = paid and Contribut
Therefore The case should be classified as : (ClASS = entitled)
with confidence = 96.29
Round  9:
Group (not entitled)Proposes a new Rule against a move previously played by Group(entitled )
The case has the following features : Age = age<60 , Income = income<15 , Capital = capital<2000 and Contribution Year2 = not paid
Therefore The case should be classified as : (ClASS = not entitled)
with confidence = 97.43
If it has the additional attribute  : Contribution Year1 = not paid)
Round  10:
No Moves Have Been Played in this Round
Round  11:
No Moves Have Been Played in this Round
THE WINNER IS
Group(ClASS = not entitled)
Group Index: (1)
```

**Figure 6.7(b) . The Housing Benefit dialogue (continued).**

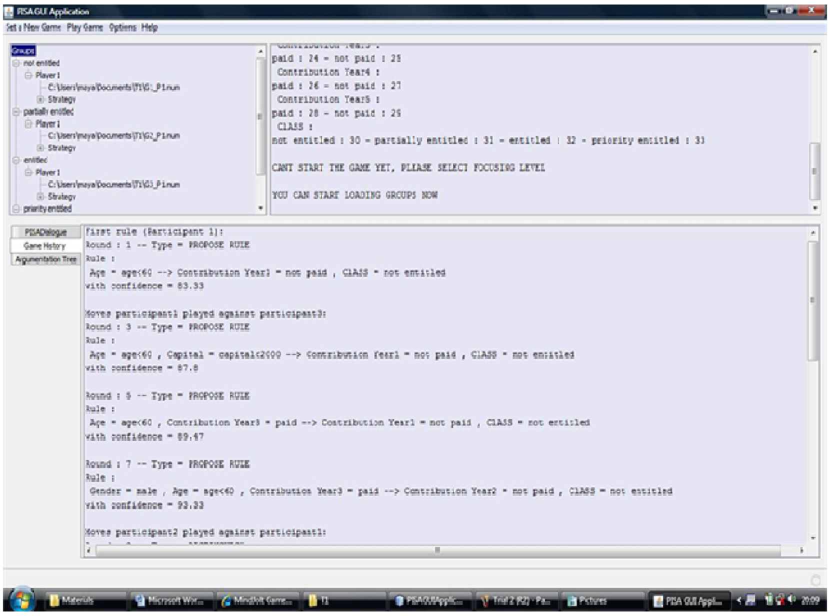


```
paid : 24 - not paid : 25
 Contribution Year4 :
paid : 26 - not paid : 27
 Contribution Year5 :
paid : 28 - not paid : 29
 CLASS :
not entitled : 30 - partially entitled : 31 - entitled : 32 - priority entitled : 33

CANT START THE GAME YET, PLEASE SELECT FOCUSING LEVEL

YOU CAN START LOADING GROUPS NOW
```

**Figure 6.8. The Argumentation Tree of the Housing Benefits example.**



```
paid : 24 - not paid : 25
 Contribution Year4 :
paid : 26 - not paid : 27
 Contribution Year5 :
paid : 28 - not paid : 29
 CLASS :
not entitled : 30 - partially entitled : 31 - entitled : 32 - priority entitled : 33

CANT START THE GAME YET, PLEASE SELECT FOCUSING LEVEL

YOU CAN START LOADING GROUPS NOW
```

```
First rule (Participant 1):
Round : 1 -- Type = PROPOSE RULE
Rule :
 Age = age<60 --> Contribution Year1 = not paid , CLASS = not entitled
with confidence = 83.33

Moves participant1 played against participant3:
Round : 3 -- Type = PROPOSE RULE
Rule :
 Age = age<60 , Capital = capital<2000 --> Contribution Year1 = not paid , CLASS = not entitled
with confidence = 87.8

Round : 5 -- Type = PROPOSE RULE
Rule :
 Age = age<60 , Contribution Year5 = paid --> Contribution Year1 = not paid , CLASS = not entitled
with confidence = 89.47

Round : 7 -- Type = PROPOSE RULE
Rule :
 Gender = male , Age = age<60 , Contribution Year3 = paid --> Contribution Year2 = not paid , CLASS = not entitled
with confidence = 93.33

Moves participant2 played against participant1:
```

**Figure 6.9. The History Log of the Housing Benefits example.**

### 6.6.2. PISA Application Example (2): The Nursery Example[36]

A more detailed example, focusing only on the PISA dialogue and the structure of the Argumentation Tree, rather than the Java application, is now given. The purpose of this example is twofold: (i) to establish the style of dialogues

[36] This example was previously published in (Wardeh et al., 2009c).

produced by PISA, and (ii) to illustrate that PISA can of handle real world as well as hypothetical scenarios. Here, PISA is applied using a dataset describing the processing of applications for a nursery school in Ljubljana (Olave et al, 1989). The dataset was obtained from the UCL data repository (Blake and Merz, 1998). The Nursery dataset was originally derived from a hierarchical decision model developed to rank applications for nursery schools. It was used during the 1980s when there was excessive enrolment to these schools in Ljubljana, Slovenia, where there were often two applications for every place. The final decision depended on eight factors forming three sub problems: the occupation of parents and the child's current nursery provision; the family structure and its financial standing; and the social and health picture of the family. The model was developed using the DECMAK expert system shell for decision making (Bohanec and Rajkovic, 1990). The original dataset consisted of 12960 records classified into five levels of recommendation: not recommended, recommended, highly recommended, priority and special priority. The distribution of the classes in terms of records was as follows: 33.33%, 0.015%, 2.53%, 32.91% and 31.204%. Note that the recommended and highly recommended classifications are rather rare. For the purpose of the experiment described here the records in the recommended class were removed from the dataset. A four player game was therefore designed. The dataset has also 8 attributes other than the class attribute:

- Parents occupation: usual, pretentious, of great pretension
- Childs nursery: proper, less proper, improper, critical, very critical
- Form of the family: complete, completed, incomplete, foster
- Number of children: 1,2,3, more than 3
- Housing conditions: convenient, less convenient, critical
- Financial standing of the family: convenient, inconvenient
- Social conditions: non-problematic, slightly problematic, problematic
- Health conditions: recommended, priority, not recommended

## The PISA Debate:

For this example a run of PISA with four players was performed, each representing one of the four possible classifications. The players are referred to as NR (not recommended), HR (highly recommended), PR (priority) and SP (special priority). The input case was chosen as one that should be classified as highly recommended, since this is the rarest, and hence the classification most likely to be in dispute. Specifically the case has the following attributes: parents have a usual occupation, has less than proper nursery, completed family, two children, convenient housing, inconvenient finance, non problematic social conditions and recommended health conditions. The support/confidence thresholds were set at 1% and 50% respectively. These values were chosen as they are well established as the default thresholds in the data mining community (e.g. (Coenen and Leng, 2005), (Li et al, 2001) and (Liu et al, 1998)).

The dialogue commences when the chairperson invites the HR player agent to propose the opening argument, HR proposes the following rule (R1):

> HR - Proposes a New Rule: The case has the following features: usual occupation, less than proper nursery, convenient housing and recommended health. Therefore this case should be classified as (highly recommended). **With confidence = 52.38%.**

This rule is attacked by the other three agents in the second round (R2, R3 and R4 respectively) as follows (the reader might find it helpful to refer to the completed argument tree shown in Figure 6.10 as the debate develops):

> PR - Counter Rule: The case has the following feature: recommended health Therefore this case should be classified as (priority recommended). **With confidence = 55.72%.**

> NR - Counter Rule: The case has the following features: usual occupation, complete family, 2 children, convenient housing and inconvenient finance. Therefore this case should be classified as (not recommended). **With confidence = 55.55%.**

```
SP - Distinguishes the previous rule: The case has the
following additional feature: family=complete. Therefore
my confidence in this case being of class (highly
recommended) is no more than 20% only.
```

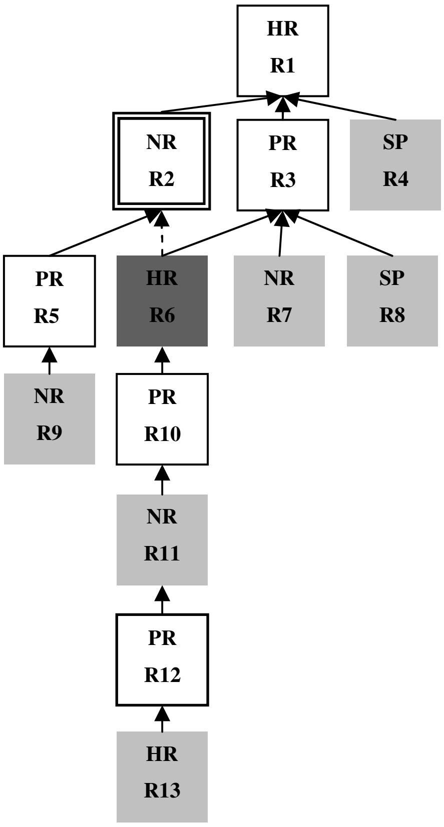


**Figure 6.10. The Argumentation Tree of the nursery example.** *Dark Grey=Green nodes, Double Lined=Purple nodes, and Single Lined=Red nodes.*

Note that SP does not propose a rule of its own. Since the case falls into the narrow band of highly recommended we might expect to find reasons for the classifications on either side, but not the very different special priority. Nonetheless, SP could play a useful role in critiquing the arguments of the other players. At this stage PR is ahead as it has the best un-attacked rule. In round three all four players make moves:

- HR: proposes a new rule to attack the current best rule. In fact it has an excellent rule (R5):

> HR - Proposes a New Rule: The case has the following features: usual parent, less than proper nursery, complete family, convenient housing and recommended health. Therefore this case should be classified as (highly recommended). **With confidence = 85.71%.**

- NR: distinguishes PR's argument by pointing out that usual occupation and recommended health only gives priority with 18.64% confidence (R6).
- PR: proposes a counter rule against NR's rule from round two (R7):

> PR - Proposes Counter Rule: The case has the following features: usual occupation, less than proper nursery and recommended health. Therefore this case should be classified as (priority). **With confidence = 61.16%.**

- SP: distinguishes PR's rule from round two by pointing to the usual occupation, but from its data the modified rule has 19.9% confidence (R8).

Now HR is back in the lead. Note that the proposed rule is the same as the rule modified by SP in round two. This difference in confidence is explained by the fact that SP may have very few highly recommended cases in its database.

In round four SP has no move. The other two agents can, however, make moves:

- NR: distinguishes PR's rule from round three, by pointing out that recommended health reduces the confidence of the priority classification to only 20% (R9).
- PR: proposes a counter rule against HR's rule of round three (R10):

> PR - Counter Rule: The case has the following features: less than proper nursery, completed form, inconvenient finance and recommended health. Therefore this case should be classified as (priority). **With confidence = 86.95%.**

Now PR is winning, but in the fifth round this can be distinguished by NR, since the addition of non-problematic social problems reduces the confidence to just 20% (R11). In the sixth PR proposes another rule (R12):

```
PR – Proposes a Counter Rule: The case has the following
features: usual occupation and less than proper nursery
and recommended health Therefore this case should be
classified as (priority). With confidence = 65.95%.
```

This, however, can be distinguished by HR since adding non-problematic social behaviour again reduces the confidence to 20% (R13). This reinstates the argument of HR made in round 3. No more arguments are possible at this stage, and so the final classification is highly recommended

## 6.7. Summary

This chapter has given details of how multiparty "*Arguing from Experience*" can be achieved using the PISA Framework. This framework allows for any number of participants to take part in the underlying dialogues. The original contribution of PISA is the mechanisms whereby it addresses the many challenges found in multiparty dialogues which are either not present or not of significance in the two-party ones. Of particular note is the control structure used in PISA, the turn taking policy, the approach to game termination and the definition of the roles of the participants allowing them to adopt differing strategies. The supporting Argumentation Tree data structure is also significant. Overall, PISA offers several advantages:

- It allows argumentation between any number of participants rather than the more usual two. This required consideration and resolution of a range of issues associated with dialogues with more than two participants.
- It operates without the need for a (hand-crafted) knowledge base but instead allows participants to generate arguments using ARM techniques.
- The process leads to a reasoned consensus, which is not obtained through, say, voting, which increases the acceptability of the outcome to all parties.

The next chapter will investigate other areas of interest regarding multiparty "*Arguing from Experience*", and the possible treatments of these issues using the

PISA Framework. In particular, the development of an advanced game strategy model to accommodate the features of multiparty dialogues. The potential for participants to form dynamic groups, and how such groups once formed can act like one unit on the behalf of all their members, will also be addressed.

## 6.6. The Implementation of PISA and Example Dialogues.

# Chapter 7: Advanced Issues in PISA

This chapter continues the discussion about the design of the PISA Framework. The basic structure and the main functions of PISA were described in the previous chapter. Nevertheless, the suggested model "*so far*" still has room for improvements, in particular regarding the issues of participants' strategies and the formation of groups of individual players. The structure of PISA suggested in the previous chapter raises two questions:

> *What is the strategy design most suited for the purposes of multiparty dialogues produced in PISA?*
>
> *How can groups of individual players be constructed most effectively?*

The first question concerns the issue of strategies and tactics in PISA. A discussion of the PISA strategy problem and the advocated solution is given in Section 7.1. The second question relates to the design structure discussed in the previous chapter, having modelled the participants in PISA dialogues as either individual players or groups of players sharing the same objective. Section 7.2 embarks upon the structure of such groups, mainly the arrangements related to the decision making process within each group. Section 7.3 concludes this chapter with a summary. The analysis given in this chapter will provide examples of the issues investigated using classification scenarios based on the RPHA specification given in the previous chapter (Section 6.5).

## 7.1. Strategy Design for PISA

Chapter 4 proposed a layered strategy model for players taking part in PADUA dialogue games to facilitate the selection of the kind of move to be put forward at each round of the dialogue, and the content of this move. The proposed strategy model comprises four levels:

- Level 0: Defines the game mode.
- Level 1: Defines the players (agents) profiles.
- Level 2: Defines the strategy mode.
- Level 3: Defines some appropriate argumentative content depending on the promoted tactics.

The issue of strategy design for PISA dialogues is considerably different from that in the two-party PADUA dialogues, mainly because the dialogue game in PISA is more complicated than in PADUA. This complexity arises from the fact that, unlike PADUA, PISA dialogues often take place between more than two participants. A second difference is that PISA determines which party has won a completed dialogue game in a different manner from PADUA. In PISA it is not always the case that the last contributing participant wins: dialogues may continue for a number of rounds after a winning argument (represented by a legal move) has been placed. These two differences indicated that the strategy design proposed for PADUA cannot simply be applied in the PISA framework, and so some in-depth modification is required if they are to be considered suitable for PISA dialogues. Another point of distinction between PADUA and PISA, over the issue of strategy, is that although strategies are often designed for individual (players) agents, they are applied in the context of dialogues. In PADUA, dialogues are clear-cut two-party exchanges of speech acts. However, dialogues conducted in PISA involve an indefinite number of participants which might be individual players or groups of players. Thus, the strategy design for PISA should also account for this point. PADUA strategies are intended for individual players participating directly in a dialogue game, while in PISA individual players may not be involved directly in the ongoing dialogue, but instead they may engage in an inner dialogue within their own group, the result of which is determined by the leader of the group. The issue of strategy design for groups is further explained in Section 7.2.

Acknowledging the above differences means that any appropriate strategy model for PISA needs to take into consideration the following points:

- *The status of the argumentation tree.* Participants can incorporate different views of this status in their own strategies. Each agent can consider as many, or as few previous moves (tree nodes), in order to make a decision with respect to the best next move. By taking more previous moves into their consideration, participants can potentially plan their next move better than those which consider (say) only the moves played in the last round.
- *Whether the player is a member of a group or not.* It is only when an individual player (agent) is the only advocate of its own "*view*" that this player is the master of its own moves. Where two or more agents advocate the same "*view*", then they are subjects to the decision making process of their assigned group. The issues related to the strategy design within each group are discussed in detail in Section 7.2.
- *Whether the participant (player or group) has to participate in a certain round or not.* As discussed in the previous chapter, in PISA, participants are not obliged to take part in each single round of the dialogue game (turn skipping was discussed in the previous chapter). PISA also allows for participants currently winning the game to skip rounds as long as their position is not undermined by the other participants.

Strategies in PISA should, however, maintain the four levels from PADUA's strategy model. In this sense PISA extends and builds upon the basic strategy outlines for PADUA. Taking all the discussed points into consideration, a six-level strategy model was designed for individual player agents taking part in PISA dialogues. Figure 7.1 illustrates this model. The proposed six levels are divided into two tiers, the *lower tier* encapsulates the strategy model inherited from PADUA, while the *upper tier* provides the scope through which PISA players deduce their next moves with respect to the argumentation tree.

The proposed strategy model works as follows:

- The lower tier encapsulates the same four-level layered strategy model used for PADUA (Section 4.2). However, the notion of *agent profile* is slightly altered to accommodate multiparty dialogues. PISA players applying an agreeable profile may either try to agree with all the other participants or with a pre-specified group of participants. For simplicity it is assumed that when applying an agreeable agent profile, the players will attempt to agree with all the moves represented by the argumentation tree leaves. The issue of agreeing with a subset of the moves presented by these leaf nodes is further discussed in Sub-section 7.1.4.

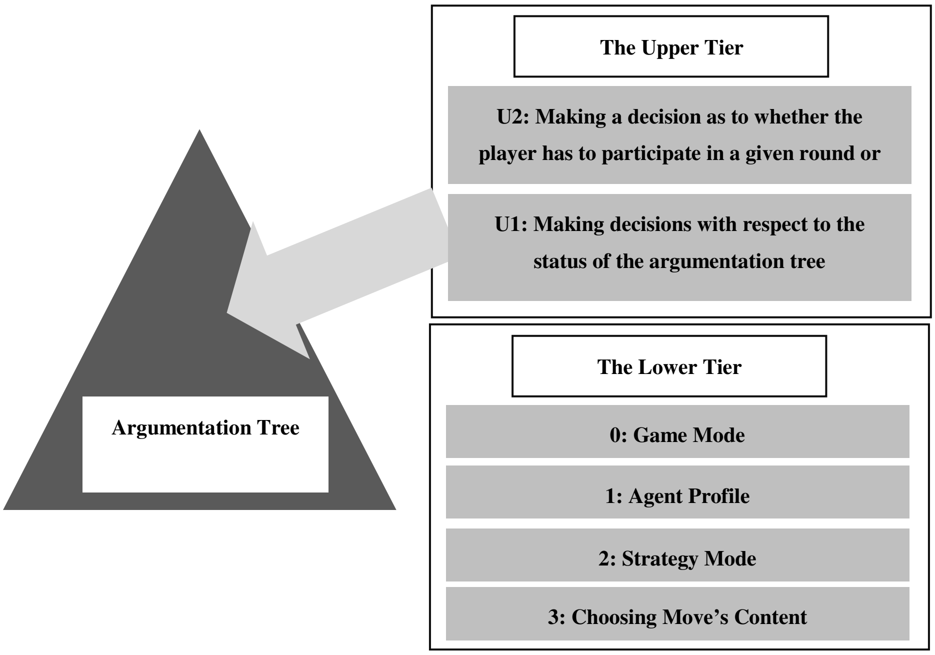

The Upper Tier
U2: Making a decision as to whether the player has to participate in a given round or
U1: Making decisions with respect to the status of the argumentation tree
The Lower Tier
0: Game Mode
1: Agent Profile
2: Strategy Mode
3: Choosing Move's Content
Argumentation Tree

**Figure 7.1. The two-tier strategy design.**

- The upper tier identifies the manner by which PISA players infer what move to play next from the current status of the argumentation tree. This tier comprises two strategy levels (Figure 7.1):
  - **Level U2**: defines whether the player "*has to*" participate in the current round of dialogue or not. Different strategies can be derived from this level. These strategies are explained in Sub-section 7.1.1.
  - **Level U1**: defines the process by which PISA players choose their next moves with respect to the argumentation tree. The argumentation tree encompasses details of the moves played thus far in the dialogue game, and the attacks relation amongst these moves. By consulting this tree, players can base their decision on a number of issues with relation to their next moves, such as: which opponent to attack next and which speech act to use.

Level U1 of the layered strategy advocated promotes three different modes for deducing the next moves from the current status of the argumentation trees:

- *Full Tree Inference Mode*: enables the derivation of strategies using a full view of the argumentation tree. Players can select their next move on the basis of their interpretation of the whole argumentation tree (their interpretation of the dialogue thus far).
- *Leaf Nodes Inference Mode*: leads to strategies with a limited view of the argumentation tree. Such PISA players consider only the leaf nodes of the argumentation tree (undefeated moves/attacks) when making decision about what moves to play next.
- *One Leaf Node Only Inference Mode*: from which all the derived strategies are basically two-party PADUA-style strategies.

Recall from Chapter 4 that a strategy function for players (agents) taking part in games governed by the PADUA protocol was identified in Section 4.2. This function was called $Play_a$ where $a \in A$ is a given player agent. Another version of this function can now be identified for the purposes of PISA strategies. Individual players taking part in PISA dialogues may use this function to select their next moves. For each agent (player), a∈ A, $Play_a$ is defined as follows:

$$Play_a : M_{poss} \times R_{poss} \times D_{current} \times S_a \times Tactics_a \rightarrow M_{poss}$$

For the purposes of PISA only $S_a$ (the *Strategy Matrix* for the given agent) is changed such that $S_a = [havePart_a, tm_a, gm_a, profile_a, sm_a]$. Where: (i) $\text{havePart}_a$ identifies if the *a* has to take part in the next round or not, (ii) $\text{havePart}_a \in$ {true, false}. $tm_a \in TM$ is the tree inference mode, where *TM={full, leaves, one leaf}*, (iii) $gm_a \in GM$ is the game mode, where *GM* = {*win*, *dialogue*}, (iv) $profile_a \in$ *Profile* is the player profile, where *Profile* = {*agreeable*, *disagreeable*}, and (v) $sm_a \in SM$ is the strategy mode, where *SM* = {*build*, destroy}. Recall, $Tactics_a$ is the tactics matrix including the move preference and the best move content tactics. These tactics are explained in the context of each possible strategy in the forthcoming sub-sections.

### 7.1.1. Three Sub Strategies for PISA

A number of different strategies can be derived from the two-tier strategy model discussed in Figure 7.1 according to the values given to each of the parameters of $S_a$ of the strategy function $Play_a$ defined above. Three *Basic Strategies* were derived from the given model in relation to level U2 from the upper tier. These strategies form a basis from which other sub strategies are built as shall be discussed in the following paragraphs. Each strategy makes use of a set of tactics similar to that identified in Section 4.3. Note that strategies are numbered from S1 to SN, and where appropriate sub-strategies of a strategy are indicated using SK-M, for example S1-1, and so on.

**(S1) Attack Whenever Possible Strategy**

The idea behind the S1 strategy is simple: PISA players following this strategy will attack any opponent they can identify "*whenever possible*". Here, "*whenever possible*" means that players will attack their opponents whenever they can mine a suitable attack move from their background datasets, regardless of whether they need to do so or not. This strategy enhances the players' chances of winning the game by being as aggressive as possible, based on the assumption that if they attack even when they do not need to, and/or when their attacks are blindly directed against random opponents, they may win the game by undermining the arguments proposed by as many opponents as possible. Based on Level U1 of the upper tier of the advocated strategy model for the PISA framework, three sub strategies could be derived from *Attack Whenever Possible Strategy* along the three *Inference Modes*. Each mode will make use of a different opponent identification process:

- **Blind Attack Whenever Possible Sub-strategy (S1-1):** Here, individual players will attempt to arbitrarily attack any of the undefeated previous moves (leaf nodes); thus the opponent identification process is random (blind). This sub-strategy applies a *One Leaf Node Only Inference Mode* by which the attention of players will be focused on one and only one leaf node until a successful attack could be made against one of the argumentation tree's leaf nodes. Here two sub-sub-strategies are distinguished along the *Strategy Mode* element as inherited from PADUA:

– *Blind Attack Whenever Possible by Proposing new rules (Build Mode) - (S1-1-1):* Here, players will attempt to propose new rules whenever possible, therefore increasing their chances to win the game by proposing as many rules as possible, regardless of against which opponents they are directed. The idea is that, proposing high confidence rules will have the same effect regardless of which opponents they are directed against.

– *Blind Attack Whenever Possible by undermining the opponent (Destroy Mode) - (S1-1-2):* Here, players will plan to win the dialogue game by undermining their opponents' proposals whenever possible. The idea is that if one player manages to undermine all the proposals played by all the other players then this player wins the game. Of course this is not always feasible, particularly when other players are following different strategies.

Note that following a build or destroy strategy means that the player will try to apply this strategy against all the possible leaf nodes, before turning to the opposite strategy mode (e.g. destroy if the original strategy mode is build); instead of using build then destroy (or the other way around) tactics against each leaf node in order.

- **Focused Attack Whenever Possible Sub-strategy (S1-2):** Directs the players' attacks according to some ordering of the identified undefeated previous moves (leaf nodes). Thus, it promotes a *Leaf Nodes Inference Mode*, and therefore targets the next moves against the most appropriate undefeated previous move. Here the manner by which each player orders the leaf nodes is significant and can be achieved in various ways. The strategy model used here adopts a simple, yet effective, ordering: the agents will attempt to defeat any leaves representing the most threatening direct attacks against moves they have previously played. Such attacks are identified, at this level, as the leaf nodes directly attacking moves placed by a particular individual player (or the group it belongs to). If no such moves are found, or if the player has failed to defeat any of the direct attacks against its proposals, then it will try to apply S1-1. It is assumed this sub-strategy separates the leaf nodes of the argumentation tree into two groups one comprising the direct threats and the other containing the rest of the leaf nodes, no further ordering is applied on the leaf nodes in each group. Here also two sub-sub-strategies are distinguished:

– Focused Attack Whenever Possible by Proposing new rules (Build) - (S1-2-1)
– Focused Attack Whenever Possible by undermining the opponent (Destroy) - (S1-2-2).

These two types are similar to the ones discussed above (S1-1-1 and S1-1-2), the only difference being that they order the argumentation tree leaves prior to try to attack them.

- **Flexible Attack Whenever Possible Sub-strategy (S1-3):** Represents the most sophisticated of the three sub strategies derived from S1. Here individual players adopt a focused strategy similar to the one discussed above. However, instead of being restricted to a build or destroy strategy mode, players may choose whether to undermine an existing leaf node (undefeated move) or to propose a counter attack against this node. The switch from build to destroy mode, and the other way around, depends on the inference mode (Level U1) of the applied strategy. Two sub-sub-strategies are derived:

– *Leaf Nodes Inference Flexible Attack Whenever Possible - (S1-3-1): D*educes next moves from the set of the previous undefeated moves (leaf nodes). Here besides distinguishing between most threatening direct attacks and other leaf nodes. PISA players apply some ordering on the most threatening direct attacks, and then try to defeat these attacks regardless how this is done (build or destroy). For the purposes of this thesis, this sub-strategy assumes a simple order of direct threats: Green attacks are ordered in a descending order according to their confidence followed by the blue attacks ordered in an ascending order according to their confidence.
– *Full Tree Inference Flexible Attack Whenever Possible - (S1-3-2):* Promotes a full tree perception with the intention of determining an

order by which the player should attempt to attack its opponents. Figure 7.2 suggests an outline of such strategy. In this figure ArgT.GreenConfidence represents the current value green confidence, nodes(value) return a set of nodes which confidence=value, leafs(node N) returns the leafs nodes of the sub-tree which root =N. Dominant Blue means that the players has the highest number of blue nodes on the tree

```
Input: The Argumentation Tree ArgT.

Try proposing an AR with confidence > green confidence; direct
this move against the leaf node with the highest confidence.
else
 if ∃ leaf node LN : confidence(LN)==ArgT.GreenConfidence then
 attack LN.
 else
  for each N ∊ nodes(ArgT.GreenConfidence): colour(N)=Green do
  if ∃ leaf node LN ∊ leafs(N) then attempt to attack LN such
  that the colour of the N after attack = red.
  else
   if adding a blue node to the argumentation tree make the
   player dominant blue then attempt to play a blue move.
   else switch to S1-3-1.
```

**Figure 7.2. The proposed S3-2 strategy.**

### (S2) Attack Only When Needed Strategy:

The Attack Only When Needed strategy allows a player to choose whether it is necessary to take part in the current round of the dialogue game or not. Recall that participants taking part in PISA dialogues can choose to not contribute for pre-determined number of rounds without being forced to leave the dialogue. Also, participants in a winning position may not need to take an active part in the dialogue as long their position holds. Thus, the players will need to attack only when all their (past) moves have been successfully defeated by other participants; or when their attempts to undermine the proposals of all the other participants have failed (when any of the other participants puts forward a legal move). Here also three sub-strategies were derived according to the same criteria applied with the attack whenever possible strategy. However, players using these sub-strategies prefer not to contribute as long as they are in a wining position. Once this position is compromised (defeated) the players switch immediately to an underlying S1 sub-strategy. The advocated strategy model distinguishes between two types of "*wait and see*". The first is the *Build Mode Wait and See* where the player will only attack when it no longer has any green nodes on the tree, by proposing a new rule (or by advancing any other equivalent move), otherwise the player would rather "*wait and see*". The second is *Destroy Mode Wait and See* where the player would rather "*wait and see*" as long as there are either: no green nodes on the tree, or all the blue nodes belong to this player. According to these criteria S2 is further divided into the following sub-strategies, corresponding to the variants proposed for S1:

- Blind Attack Whenever Needed Sub-strategy - (S2-1)
  - Blind Attack only when needed by proposing rules - (S2-1-1)
  - Blind Attack only when needed by undermining the opponent - (S2-1-2)
- Focused Attack Whenever Needed Sub-strategy - (S2-2)
  - Focused Attack only when needed by proposing rules - (S2-2-1)
  - Focused Attack only when needed by undermining the opponent - (S2-2-2)
- Tree Dependant (Flexible) Attack Whenever Needed Sub-strategy (S2-3)
  - Leaf Nodes Inference Flexible Attack When Needed - (S2-3-1)
  - Full Tree Inference Flexible Attack When Needed - (S2-3-2)

**(S3) Attack to Prevent Forecasted Threat Strategy**

The Attack to Prevent Forecasted Threat Strategy anticipates forthcoming attacks against the participant's existing proposals; thus it is the most sophisticated strategy type in PISA. Here players deduce their best next moves based on the entire argumentation tree and use their own heuristics trying to calculate which of their previous moves may be the weakest link in their argument, and then either propose new rules to strengthen their position, or attack the positions of other participants before they have the chance to attack them. Figure 7.3 suggests an outline for this strategy. In this figure: nodes (Participant P) returns all the nodes played by P, player (node N) returns the participants who has played the move represented by N and colour (node N) returns the colour of N.

```
Input: The Argumentation Tree ArgT.

Try to propose a rule with a confidence higher than any other node on
ArgT, direct this move against the highest confidence leaf node.
else
 Try to propose a rule with confidence higher than the green
 confidence; direct this move against the highest confidence leaf node.
 else
  if ∃ leaf node LN and confidence(LN)==ArgT.GreenConfidence then
  attempt to attack LN.
  else
   for each N ∈ nodes(ArgT.GreenConfidence) and colour(N)=Green do
    if ∃ leaf node LN ∈ leafs(N) and colour (LN) !=purple then
     attempt to attack LN such that the colour of the N after attack =
     red.
    else
     if adding a blue node to the argumentation tree make the player
     dominant blue then attempt to play a blue move.
     else
      identify the current blue dominant Participant PDB.
      if ∃ leaf node LN such that player(LN)== PDB and colour(LN)=blue
      then attempt to attack LN.
      else
       for each node N ∈ nodes(PDB) such that colour(N)=blue do
       if ∃ leaf node LN ∈ leafs(N) then
        attempt to attack LN such that the colour(N) after attack =
        red.
switch to (S2-3-2).
```

**Figure 7.3. The proposed Attack to Prevent Forecasted Threat strategy.**

### 7.1.2. Integrating the Strategy Model in the PISA Application

The three broad strategies discussed above (S1, S2 and S3) are all implemented within the PISA Framework Application (Section 6.5). Recall from the previous chapter that the user could enter some information when adding a new player.

The following describes the sort of information required when adding a new *Player Agent*, in particular the input parameters defining the new player's strategy. First the user has to select some dataset for the new player. This is done by browsing through the user directories and selecting the required dataset. Secondly, the user can decide which strategy this player will follow in the PISA dialogue game. The application provides default value of the new player's strategy (S1-1-1) should the user wish to skip this step. The user has to select values for the following: (i) one of the three strategies S1, S2 or S3; (ii) the strategy mode (*build* or *destroy*) and (iii)The inference level (One Leaf, Leaf Nodes or Full Tree) for the new player. By changing these settings the user has the option to select different strategies. PISA applies a disagreeable agent profile as the default profile for its player agents, as this framework was originally intended to model multiparty persuasion from experience rather than deliberation. However, this default value can be changed for any Player Agent to agreeable using the "*advanced*" options available for users, as discussed below. Also the ARM parameters such as confidence and support thresholds must also be fixed, the PISA Framework Application default values are confidence =50% and support =1%. Should the user wish to change the ARM parameter values or the agents' profiles/their game mode then they can use the "*advanced options*" button in the interface to access a special window and alter these values.

### 7.1.3. Example: Strategy in PISA

This section illustrates the strategies identified above using a number of example dialogues produced by the *PISA Application* using the strategy parameters ($S_a$) discussed previously. These examples are drawn from the RPHA artificial benefits configurations previously applied in Sub-section 6.5.1, for the purposes of emphasizing the role of the players' strategies in PISA dialogues. Three examples are chosen to demonstrate this point, each representing a PISA dialogue between four individual Player Agents over the classification of an RPHA case. The input case is that of: *a 63 years old female applicant, who satisfies all the benefits condition and has served in the armed* *forces and has paid her contribution in the past five years*. This case should classify as entitled to priority benefits. The four players engaging in the following examples are referred to as PR (priority entitled), EN (entitled), PE (partially entitled) and NE (not entitled). These players are all disagreeable, and engage in the dialogues in game mode. The other parameters in their strategy configurations differ according to the examples setup. Each of the following examples will consider the first four rounds of the PISA dialogue between the four participants in order to illustrate the effects different strategies have on PISA dialogues. The details of the full length dialogues are give in Appendix B.

### PISA Strategy Example 1

Here each of the four players applies the same strategy: a *focused attack when possible strategy in build mode (S1-2-1)*. The chairperson invites the EN player agent to propose the opening rule (argument). Thus EN suggests the following association:

```
EN - Proposes a New Rule: The case has the following
features: 2000£<capital<3000£ and 15%<Income<20%.
Therefore this case should be classified as (Entitled).
With confidence = 67.54%.
```

The reader might find it helpful to refer to the completed argument tree shown in Figure 7.4 as the debate develops. The initial rule is attacked by the other three player agents in the second round, as follows:

```
NE - Counter Rule: The case has the following features:
60<Age<65 and 15%<Income<20% Therefore this case should
be classified as (Not Entitled). With confidence = 69%.

PE - Counter Rule: The case has the following features:
Contribution Y1 = paid and Contribution Y5 = paid.
Therefore this case should be classified as (Partially
Entitled). With confidence = 68.4%.
```

> PR - Counter Rule: The case has the following features: Residency = armed forces and Contribution Y1 = paid. Therefore this case should be classified as (Priority Entitled). **With confidence = 75.4%.**

Note that NE uses the fact that the case under discussion is of a candidate whose age is between 60 and 65 years to attacks the EN argument, since all the males applicants in this age groups are not entitled to benefits; at this stage PR is ahead as it has the best un-attacked rule. In round three all four players make moves:

- EN, PE and NE proposes a new rule to attack the current best rule (the previous PR rule):

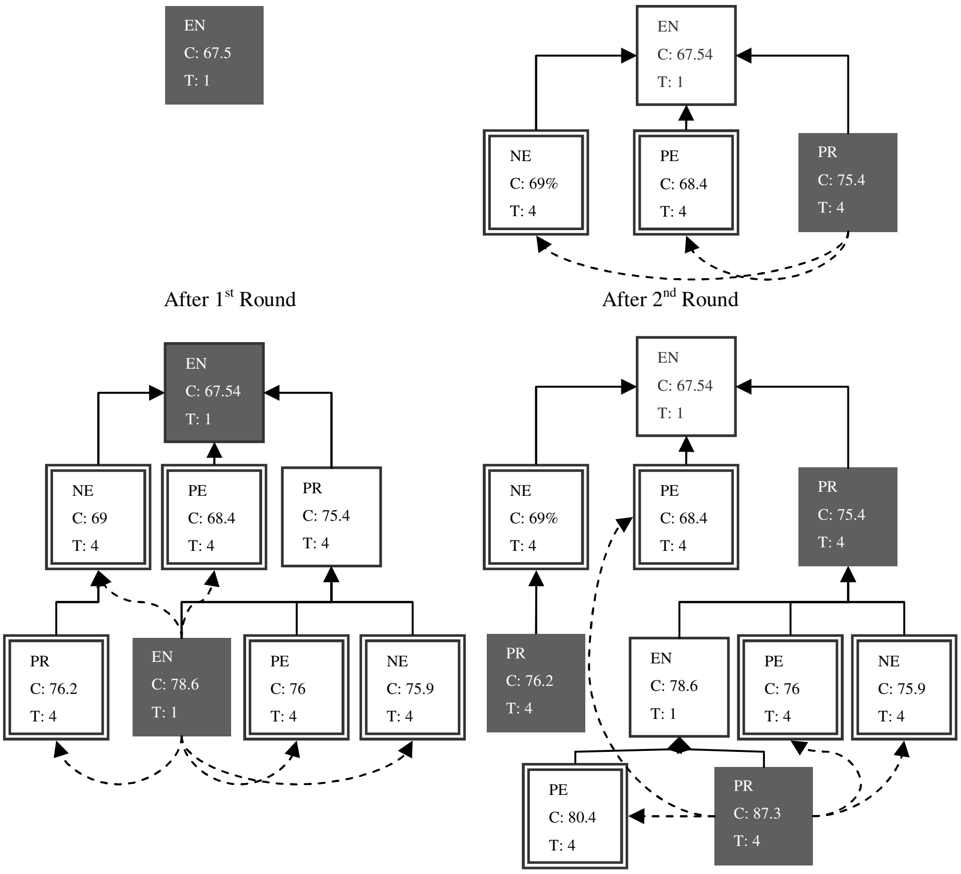
  > EN - Proposes a New Rule: The case has the following features: Gender = female, 60<Age<65, 2000£<Capital<3000£ and 15%<Income<20%. Therefore this case should be classified as (Entitled). **With confidence = 78.6%.**

  > NE - Counter Rule: The case has the following features: 60<Age<65, 15%<Income<20% and 2000£<Capital<3000£. Therefore this case should be classified as (Not Entitled). **With confidence = 75.99%.**

  > PE - Counter Rule: The case has the following features: Contribution Y1 = paid, Contribution Y2 = paid and Contribution Y5 = paid. Therefore this case should be classified as (Partially Entitled). **With confidence = 76%.**

- PR attacks the previous NE move. Although this move in not necessary, yet PR followed its own attack when possible strategy and attacked this node because it could do so:

  > PR - Proposes a Counter Rule: The case has the following features: Residency= armed forces, 15%<Income<20% and contribution Y1=paid. Therefore this case should be classified as (Priority Entitled). **With confidence = 76.2%.**



After 1st Round After 2nd Round

After 3rd Round After 4th Round

**Figure 7.4. The Argumentation Tree throughout PISA Strategy Example 1.** *Dark Grey=Green nodes, Double Lined=Purple nodes, and Single Lined=Red nodes.*

Now EN is back in the lead. Note that NE has again played a rule based on the age of the candidate to try and persuade the other participants to not issue any benefit to this candidate, this is the last move this participants would be able to play to stress this fact. In the fourth round NE and EN have no moves. The other two agents can, however, make moves against the winning position from last round as follows:

> PE - Counter Rule: The case has the following features: Contribution Y1 = paid, Contribution Y2 = paid, Contribution Y4 = paid and Contribution Y5 = paid Therefore this case should be classified as (Partially Entitled). **With confidence = 80.4%.**

```
PR - Counter Rule: The case has the following features:
Residency=armed forces, 15%<Income<20% and Contribution
Y1 = paid Therefore this case should be classified as
(Priority Entitled). With confidence = 87.3%.
```

The last round concludes this example. Note that PR has managed to win the dialogue, thus the resulting classification is correctly identified as priority entitled.

### PISA Strategy Example 2

In this second example the strategy configuration of PISA Example1 is changed such that agent players NE and PR apply a *destroy strategy*. Thus, NE will have more scope to critique other players' positions. More importantly, this example will reveal that by changing the strategy the output of the dialogue will drastically differ. The reader can refer to Figure 7.5 for the development of the argumentation tree for this example. The new dialogue commences in a similar manner to the previous example, as the chairperson invites EN once more to start the dialogue, and EN responds by playing the same opening rule from the previous example. This rule is attacked by the other three player agents in the second example, as follows:

- PE proposes the same rule it used in the same round in Strategy Example 1.
- NE distinguishes EN's argument from the first round by demonstrating that 60<age<65 only gives Entitled with a confidence of 19.9%.
- PR distinguishes EN's argument from the first round by demonstrating that contribution year 4 = paid and contribution year 5 = paid only gives Entitled with a confidence of 20%.
- Thus after the second round, PE is in the winning position rather than PR because PR is applying a destroy strategy instead of the build one it has used in the previous example.

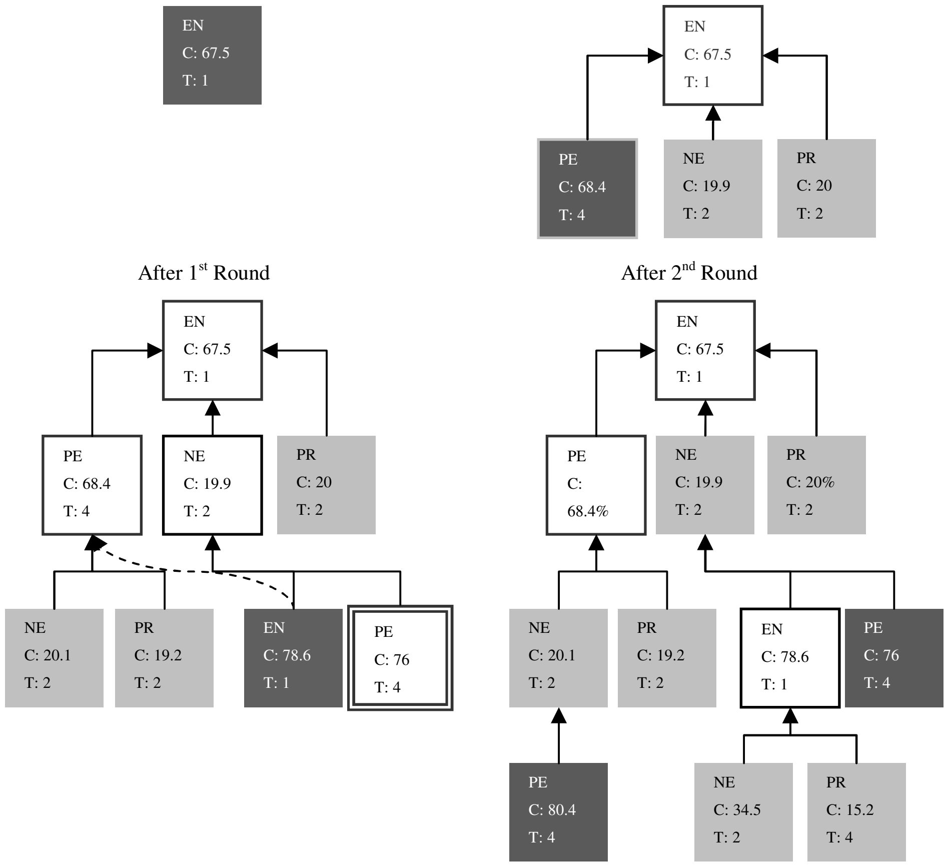


**Figure 7.5.** **›n Tree throughout f PISA Strategy Example 2.** *Dark Grey=Green nodes, Light Grey=Blue nodes, Double Lined=Purple nodes, and Single Lined=Red nodes.*

All the four players take part in the third round. Note that PE participates in this round by playing the same move it played in the same round in the previous example, but this time it is directed against NE's move from round two. Again this move is not necessary, and if PE were applying an attack when needed strategy then it would have not played it. The other three participants play the following moves:

- EN proposes the same rule it used in the same round in PISA Strategy Example 1.
- NE distinguishes PE's argument from the last round by demonstrating that 60<age<65 only gives Partially Entitled with a confidence of 20.1%.

- PR distinguishes PE's argument from the last round by demonstrating that 20003<capital<3000£ only gives Partially Entitled with a confidence of 19.2%.

Now, EN is back in the winning position, in the same manner as the previous example. Note that NE has used the age group as a distinguishing factor this time rather than as a key attribute in arguing for advocating its own "*view*".

In the fourth round, EN has no moves. The other three agents can, however, make moves against the winning position from the last round as follows:

- PE proposes the rule it used in the same round in PISA Strategy Example 1.
- NE distinguishes EN's argument from the last round by demonstrating that 60<age<65, contribution year1 = paid, contribution year2 = paid only gives Entitled with a confidence of 34.5%.
- PR distinguishes EN's argument from the last round by demonstrating that paying contribution in years 1, 2, 3 and 4 only gives Entitled with a confidence of 15.2%.

Note that this example has evolved in a different manner to the previous one: by the end of round four, PE is winning the dialogue instead of PR. This is because PR is applying a destroy strategy rather than a build one. This emphasises the importance of the strategy mode in multiparty dialogues. However, the ultimate result of this dialogue is rather different to the one discussed here. The actual dialogue produced by PISA took ten rounds, the last two of which had no moves, and EN emerged as winner by the end of that dialogue. For reasons of space, the last four rounds were omitted from this example. However, the same result still applies: PR lost this game because it was not equipped with an adequate strategy.

### PISA Strategy Example 3

Let us now assume that the four players taking part in the above example apply more perceptive strategies, in relation to the argumentation tree, than the previous two examples as follows:

- PR and EN: apply S2-3-2 (full tree inference attack when needed) .
- PE applies S3 (preventing forecasted threat).
- NE applies S2-2-2 (destroy focused attack when needed).

These strategies will produce a different dialogue and a different argumentation tree from the ones discussed in the previous example. Figure 7.6 shows the development of the argumentation tree for this example. The dialogue commences in a similar manner to the previous two examples, with EN opening the dialogue with the sane initial rule as the previous two examples. This rule is attacked by the other three player agents in round two, as follows:

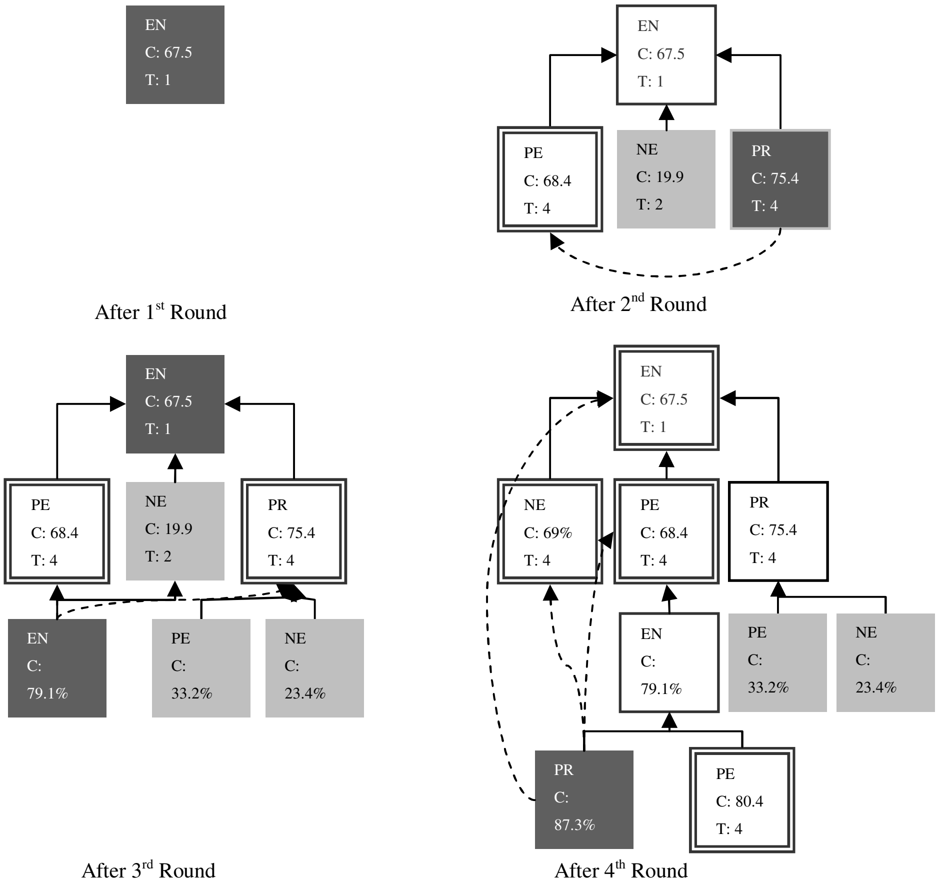
- First PE and PR propose the same counter rules they have presented in the same round in PISA Strategy Example 1.
- NE distinguishes EN's argument from the first round using the same rule from PISA Strategy Example 2.

Note that after the second round, PR is in the winning position as it has played the rule with the highest confidence so far. Only three players take part in round three; PR skips this round because it is in the winning position so there is no need for it to take part in the dialogue at this stage. The other three participants play the following moves:

- EN plays an increase confidence move against NE's move from last round:

```
EN - Increases the confidence of a previous rule by
stating that the case has the additional features:
Contribution Y1 =paid and Contribution Y2= paid.
Therefore this case should be classified as (entitled).
With confidence = 79.1%.
```

- NE distinguishes PR's argument from the last round by demonstrating that 60<age<65 only gives Priority Entitled with a confidence of 23.4%.
- PE distinguishes PR's argument from the last round by demonstrating that 2000£<capital<3000£ only gives *priority entitled* with 3.22% confidence.



After 1st Round

After 2nd Round

After 3rd Round

After 4th Round

**Figure 7.6. The Argumentation Tree throughout PISA Strategy Example 3.** *Dark Grey=Green nodes, Light Grey=Blue nodes, Double Lined=Purple nodes, and Single Lined=Red nodes.*

Now, EN is back in the winning position, in the same manner as the previous two examples. However, the current example differs from the previous:

- PE has played a distinguishing move rather than a counter example move as it has forecasted that such a move is better than playing a build move, anticipating that the other participants may play moves with better confidence than the counter attack move it has mined against PR's position.
- NE has again used one attribute from the case under discussion to undermine the argument of the player in the winning position.
- EN has chosen to direct its move against NE's move from round two because, according to its strategy, this is better than playing a proposing new rules move against one of the other two nodes on the argumentation

tree, because it thus managed to defend its original position and attack all the other players' positions with one move.

Only two players take part in the fourth round. EN does not contribute to this round because it has the winning position, and NE has no more moves. The other two players attack EN's move from round three by counter attacking it using the same moves they have played previously in the same round in PISA Strategy Example 1. Note here that PR managed to gain a win in this dialogue because it has chosen d its move using larger number of previous moves that in the previous two examples.

### 7.1.4. Discussion

This section has examined some of the issues related to strategy design for the individual players (agents) engaging in multiparty "*Arguing from 'Experience*" dialogues within the PISA framework. The suggested two-tier strategy model provides PISA players with a range of different possible strategies varying in complexity, in particular regarding the manner in which the players make use of the argumentation tree. The above discussion can be reinforced with two additional points. The first of which considers the issues of "*agreeing with other participants*" in multiparty "*Arguing from Experience*" dialogues. The second point relates to the issue of "*temporary coalitions"* between different participants against one particular opponent.

With respect to the first point it was assumed previously in this chapter that agreeable PISA players will try to agree with all the moves represented by the argumentation tree leaf nodes. In other words, these players will attempt to agree with all the arguments that have not yet been defeated, and then to launch their attacks only against arguments they could not agree with (because no adequate ARs could be mined from the players' datasets). Also one player may prefer, for strategic reasons, to agree with certain other participants and not with the rest. For these reasons, each PISA player maintains a list, referred to as the *to-agree-with list*, comprising the participants it will attempt to agree with during the course of the game rather than attacking. Such a list is composed on the basis of the discussion domain: One player may prefer agreement with participants advocating classifications adjacent[36] to the one it is advocating, rather than losing the dialogue to other parties. This style of agreeable profiles is referred to as "*Biased Agreeable Profile*", in order to distinguish it from the (fully) "*Agreeable Profile*" discussed above. The notion of "*Biased Agreeable Profiles*" is of importance in domains where there are a number of adjacent classifications. Take for example the RPHA fictional domain from the previous sub-section: PR may settle for the "*entitled to benefits*" classification proposed by EN rather than not getting any benefits or getting just partial benefits (as proposed by the other two players in the game NE and PE respectively). EN and PE on the other hand may settle for anything other than not getting any benefits, while NE will not prefer agreement with any of the other three participants.

The above notion of *Biased Agreeable Profiles* could be applied as a mechanism for *coalition formation*, in which a number of participants may attempt to "t*emporarily agree*" with each others for strategic reasons, in order to overcome stronger opponents In this case, a number of participants could form a "*temporary coalition*" by which they join forces and cease attacking each other for a limited number of rounds for the purposes of defeating the stronger opponent(s). Once this goal is achieved, say when the stronger opponent(s) drops out of the dialogue game, then the participants in the "*temporary coalition*" can break up and resume attacking each other as they would have done prior to forming the coalition. Note that "*temporary coalitions*" differ from "*Biased Agreeable Profiles*" in two ways. Firstly "*temporary coalitions*" are temporary, which means that once the goal of the coalition has been achieved the participants in the coalition have no reason to continue being in this coalition. Secondly participants in a temporary *coalition* cease attacking each other, while participants with "*Biased Agreeable Profiles*" try to avoid attacking participants in their lists if possible; also there is nothing stopping the participants in the players' "*to-agree-with*" lists from attacking these players.

[36] Adjacent classification here refers to a class value related to or close to the classification this particular participant tries to prove true. Alternatively an agent might choose to agree with all those agents that give a better (or worse) outcome to the claimant.

Equipping PISA players with mechanism to enforce temporary coalitions is an ambitious extension of the PISA Framework. However, a number of issues must be addressed, if a successful implementation of coalitions is to be brought together. Chapter 9 will give a summary of these issues, based on the above discussion, and provide directions to tackling them in future extensions of PISA.

## 7.2. Groups and Leadership in PISA

Recall from the previous chapter that individual PISA players advocating the same thesis (for example the same possible classification of the input case) are required to “*join forces*” and act as a single "*group of players*". Every group is allowed only one move per round. This restriction aims at simplifying PISA dialogues. The proposed notion of groups prevents individual players sharing the same objective from arguing without consulting each other and consequently causing contradictions amongst themselves or attacking each other. This may, however, lead to a situation in which the weaker parties (within the groups) are forced to withdraw from the game and the remaining stronger members no longer have sufficient shared experience to win. Group formation is automatic in PISA. When a new individual player joins a dialogue game, over a case from a particular domain, it has to make its objective clear. The player’s objective represents the thesis this player proposes ($G_a$). The chairperson then decides if this new player should participate in the dialogue as an individual player, or should become a member of an existing group of other players which advocated thesis matches the one proposed by the new player. In each group, the members have to select a leader from amongst them. This leader will act as a representative of its group in the dialogue, and is usually the “*smartest*” and “*most experienced*” (the one with the largest amount of data available at its disposal) member of the group. Player’s *smartness* relates to the strategy this player applies. Hence, the smartest member is the one with the most sophisticated strategy amongst the group’s members, where strategies are ranked according to their level of understanding of the history and the process of the dialogue.

The leader guides the inter-group dialogue, and selects which of the moves suggested by the group's members, including the leader's move, is the best to be played in the next round. This inter-group dialogue is a variation of "*targeted broadcasting"*, in which only group members can listen to what is being "*discussed in the group*", while other participants are completely unaware of these dialogues. The leader can also redirect other members' moves against different opponents, or advise them to follow its own strategy, an act that makes the group benefit from the different strategies applied by its members and from the differences in their experience.

Group formation in PISA is a clear-cut process when compared with the work on group (team) formation in the literature on cooperation among intelligent agents (e.g. (Kinny et al, 1992), (Cohen et al, 1999), and (Ogston et al, 2005)). This is mainly because what matters for PISA players, is not achieving a complex task by distributing actions amongst the group members. Rather, the argumentation dialogue process is supposed to lead to a coherent classification of the cases under discussion. All the group's members perform the same task: mining the best possible argument in the context of the ongoing dialogue, according to their strategy and experience. The following sub-sections describe in details the different types of groups in PISA and the decision making process within each type.

### 7.2.1. Groups Types

The internal structure of groups in PISA varies greatly depending on the strategy and the experience of each of its members. Mainly, because in each group a decision making process takes place at the beginning of each round to settle on the best move to play (or whether it is better to not contribute) in this round. Such a process implies that a minimum level of discipline should be respected by the group's members. For this purpose, a particular member in each group is chosen as its leader, to facilitate this decision making process and ensure that the other group's members work in harmony to convince the other participants in the dialogue that the case under discussion classifies as advocated by the group.

Two factors are essential to each group: the strategy factor and the experience factor. The strategy factor concerns the strategies of the individual players in the group. In some cases, all the members may have incorporated the same strategy, while in others each member applies its own strategy and thus a strategy ranking is required in order to determine who is going to be the group leader. The second factor relates to the experience of the group's members, measured in relation to the size of the dataset in which this experience is stored. Thus, an individual player with (say) 1200 records in its database is considered more experienced than one with only (say) 600 records. This factor is necessary to the operation of the group as will be discussed later in this section.

Groups in PISA are divided into two types according to the strategy factor: Homogenous and Heterogeneous groups.

**Homogenous groups:** consist of a number of individual players which share the same goal and apply the same strategy (same type in the same mode, and using the same agent profile). However, each individual player may use its own confidence/support values. In such groups the most experienced player (the one with the largest background dataset) is chosen to be the group's leader. If two or more of the group members share the same level of experience then one of them is selected at random to represent the group. Once the leader is agreed on, the group members will attempt, in each round of the dialogue, to mine the best rules according to their strategy each from their own datasets. The leader will then select the best move according to the group agreed strategy. For example, if all the group's members have adopted a *build attack only when needed blind* strategy, then the leader will select the build move with the highest confidence to place forward in the dialogue. If no such move was suggested, the leader will promote a destroy move with the lowest accuracy.

**Heterogeneous groups'** members apply different strategies. Therefore a strategy ranking is applied to determine who is the "*smartest*" amongst the group members and thus best suited for its leadership. If two or more players happen to incorporate the smartest strategy then the most experienced one is selected for leadership. If they also have the same experience then one of them is selected at random. In heterogeneous groups, the leader has the authority to force its own strategy on the other players causing them to adjust their suggested moves to suit the leader's strategy. Thus the role of the leader in this type of group is more sophisticated than in homogenous groups. A more detailed account of leadership is given in the following sub-section. PISA applies the strategy ranking described in Table 7.1 to determine the smartest possible strategy. The advocated ranking does not take into account the differences in the Game Mode (level 0) or Agent Profile (level 1) of each strategy, when assigning a rank to a given strategy. Thus, these two levels are omitted from Table 7.1. The suggested ranking also assumes that the best possible strategy is S3, followed by S2 then S1.

| **Rank** | **Name** | **Strategy (S1, S2, S3)** | **Sub-Strategy** | **Strategy Mode** |
|---|---|---|---|---|
| **1** | S3 | S3 | - | - |
| **2** | S2-3-2 | S2 | Tree Dependent - Full | - |
| **3** | S1-3-2 | S1 | Tree Dependent – Full | - |
| **4** | S2-3-1 | S2 | Tree Dependent - Leaves | - |
| **5** | S1-3-1 | S1 | Tree Dependent - Leaves | - |
| **6** | S2-2-1 | S2 | Focused | Build |
| **7** | S2-2-2 | S2 | Focused | Destroy |
| **8** | S2-1-1 | S2 | Blind | Build |
| **9** | S2-1-2 | S2 | Blind | Destroy |
| **10** | S1-2-1 | S1 | Focused | Build |
| **11** | S1-2-2 | S1 | Focused | Destroy |
| **12** | S1-1-1 | S1 | Blind | Build |
| **13** | S1-1-2 | S1 | Blind | Destroy |

**Table 7.1. Suggested ranking of PISA strategies.**

### 7.2.2. The Role of the Group Leader

Having distinguished between two types of groups, and established the leader selection process according to each type, a more detailed account of the role of the leader of the group is now given. Once a leader has been selected, this particular agent will have authority over the other members of its group. This authority entitles the leader to perform the following tasks:

- The leader's most essential task, as far as the group is concerned, is to select the best move at every round of the dialogue, from the selection of moves suggested by the group's members. The leader often chooses the moves following its own strategy. This does not mean that the leader will select its own move all the time. Rather, the leader aims at selecting the best move from amongst the suggested moves. For instance, the move with the highest confidence. Here, the differences in the members' experiences will greatly influence the leader's decision: members with different experience will often promote different content for their chosen moves, even where all the members apply similar strategies.
- The leader can compel the more experienced members (if any) to act according to the leader's strategy. This happens on a round by round basis. If a more experienced member suggests one move, in a given round, and if the leader assumes that a similar move with a better confidence, or a move with a different speech act better matching the game context, could be produced by this player, then the leader can ask this player to attempt generating another move using the leader's strategy. The leader then compares the new move (the one produced using its own strategy parameters) against the old one (the one the player has initially suggested) and chooses the best move. Consider, for example, the case where one of the experienced players has suggested a destroy move (following its own strategy) distinguishing some previously undefeated move in the dialogue. Then the leader will ask it to produce a build move. If this player replies with a build move with a high confidence (say higher than the moves suggested by the other members) then the leader will discard this player's initial move, otherwise it will discard the new move. Information about the members experience and strategy is available to its leader, through a simple dialogue, by which the leader request these information from the group's members. Additional conditions are applied to ensure that the leader practice the above authority only when needed: If the experienced members of the group apply weak strategies, and where other members have failed to produce adequate moves.

- The leader can redirect moves suggested by the other members against opponents other than the ones they have chosen. For instance, if one member suggests an "*increase confidence*" move against one opponent (say for strategic reasons), then the leader may change this move to a "*propose new rule*" and directs it against another opponent (say because this opponent threatens the group more than the one originally picked upon by the group member). Here as well, the leader is allowed to redirect the members' moves only when redirection is more rewarding according to the leader's strategy than the original move.

Note that the group's leader is not fixed. It may change when a new member joins the group, or when the current leader leaves the dialogue, and therefore the group. In the first case, the current leader has to compare its strategy and experience with the newcomer. If the newcomer satisfies the leadership conditions better, then the current leader has to step down, allowing the newest member to become the group's leader. In the second case, when the current leader leaves the game, the group members have to select a new leader from amongst them in the same manner prior to the start of the game.

This possibility of changing the leadership, from one player to another, demands a careful consideration of the leader identification process, i.e. the process by which the group's members identify the leader and communicate with it. The problem of leader selection could be solved by adopting the standard technique of token passing as used in computer networks. See for example (Ambroszkiewicz et al., 1998) where the token is used as a sign of decision power amongst a team of software agents, so that a member of the team who has currently the token enjoys the exclusive authority to decide on the status of the team. PISA implements a technique similar to token passing, but instead of passing tokens from one member to another, the leadership is identified in PISA by a "*Leadership Unit*"; each group has one "*Leadership Unit*" residing with the current group leader. This unit enables one Player Agent to perform the leadership tasks described above. When a new leader is selected this unit is passed from the previous leader to the new one. The "*Leadership Unit*" simplifies the inter-group dialogue. At the beginning of every round, all the group members (including the leader) send their moves to this unit. The leader compares these moves against its own strategy and against the experience of each group member, before deciding which move to play next in the game and against which opponent.

### 7.2.3. Groups in the PISA Framework Application

The notions of groups and leadership discussed above have been integrated in the PISA Framework Application (Section 6.5). The application gives the user the option of adding any number of players in each of the groups identified by a joint possible classification advocated by all the group's members. Figure 7.7 provides a screen shot of the group formation process in the given application. The user should first select a group, from amongst the set of possible groups[37], such that each group corresponds to one possible classification of the domain. Recall that PISA requires a description of the dialogue game to be uploaded to the system by the user prior to the start of any dialogue about any case from that domain (Section 6.5). After selecting a group, the user could add any number of player agents to this group. Once the user has inserted the required number of players into the group, the software forms the group with the desired number of players.

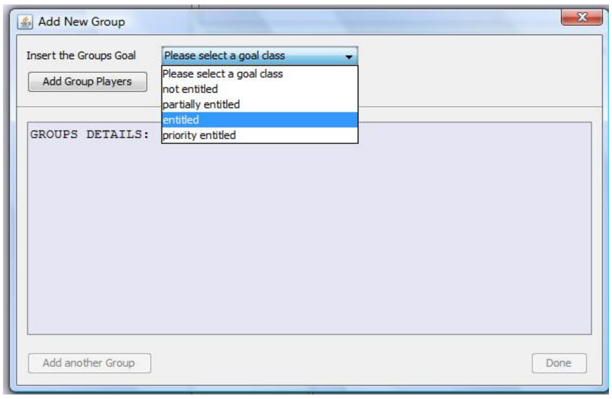
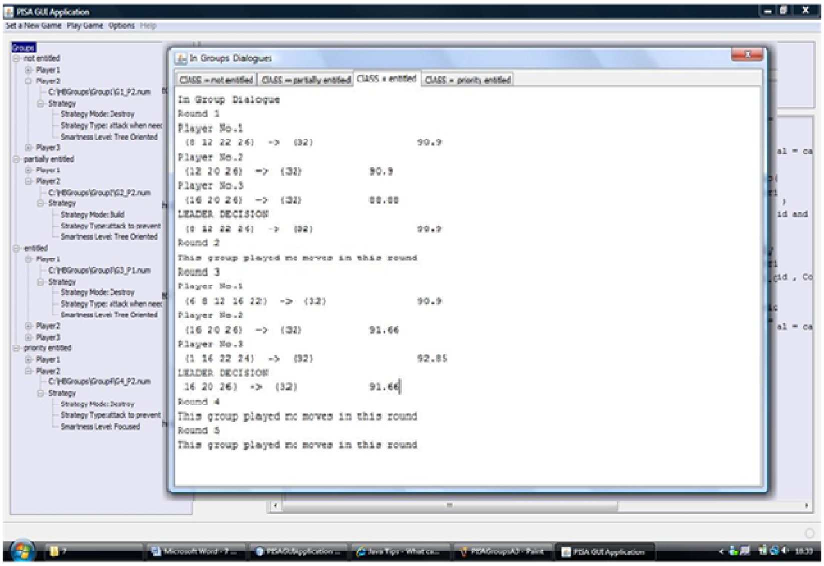


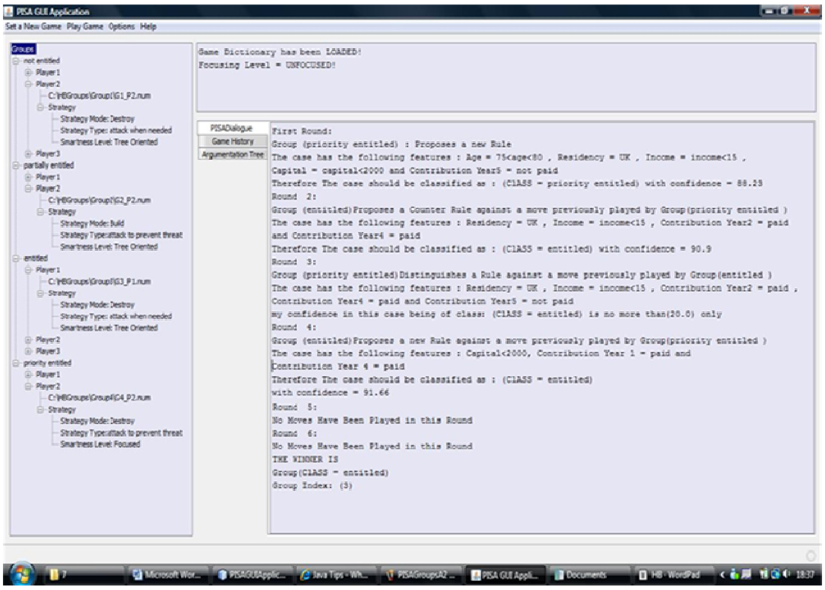
Figure 7.7. Create a new group.

[37] Note that the list of all possible groups matches that of all possible classifications and is automatically generated upon loading the game description file (game dictionary) prior to start adding new players using the application.



```
In Group Dialogue
Round 1
Player No.1
 (8 12 22 26)  ->  (32)                90.9
Player No.2
 (12 20 26)  ->  (32)          90.9
Player No.3
 (16 20 26)  ->  (32)          88.88
LEADER DECISION
 (8 12 22 26)  ->  (32)                90.9
Round 2
This group played no moves in this round
Round 3
Player No.1
 (6 8 12 16 22)  ->  (32)              90.9
Player No.2
 (16 20 26)  ->  (32)          91.66
Player No.3
 (1 16 22 24)  ->  (32)                92.85
LEADER DECISION
 16 20 26)  ->  (32)           91.66
Round 4
This group played no moves in this round
Round 5
This group played no moves in this round
```

**Figure 7.8. An example of inter group decision process.**



```
Game Dictionary has been LOADED!
Focusing Level = UNFOCUSED!
```

```
First Round:
Group (priority entitled) : Proposes a new Rule
The case has the following features : Age = 75<age<80 , Residency = UK , Income = income<15 ,
Capital = capital<2000 and Contribution Year5 = not paid
Therefore The case should be classified as : (ClASS = priority entitled) with confidence = 88.23
Round  2:
Group (entitled)Proposes a Counter Rule against a move previously played by Group(priority entitled )
The case has the following features : Residency = UK , Income = income<15 , Contribution Year2 = paid
and Contribution Year4 = paid
Therefore The case should be classified as : (ClASS = entitled) with confidence = 90.9
Round  3:
Group (priority entitled)Distinguishes a Rule against a move previously played by Group(entitled )
The case has the following features : Residency = UK , Income = income<15 , Contribution Year2 = paid ,
Contribution Year4 = paid and Contribution Year5 = not paid
my confidence in this case being of class: (ClASS = entitled) is no more than(20.0) only
Round  4:
Group (entitled)Proposes a new Rule against a move previously played by Group(priority entitled )
The case has the following features : Capital<2000, Contribution Year 1 = paid and
Contribution Year 4 = paid
Therefore The case should be classified as : (ClASS = entitled)
with confidence = 91.66
Round  5:
No Moves Have Been Played in this Round
Round  6:
No Moves Have Been Played in this Round
THE WINNER IS
Group(ClASS = entitled)
Group Index: (3)
```

**Figure 7.9. The resulting dialogue (from example presented in Section 7.2.3).**

## 7.2.4. Discussion

Thus far, in this thesis, the processes by which groups of individual players with common objective (goal, classification) are formed, and a leader for each group is selected, have been established. The two suggested processes, however, may lead to situations in which the weaker parties within each group are ignored in favour of more powerful group members. In such situations, it would be undesirable for the chairperson to force the weaker parties to withdraw from the dialogue game, because the group would be deprived of the experience of these members. It could be the case, for instance, that only the weakest members of a group are aware of the fact that water birds with black feathers living in Australia could be swans. In this case the group would find it very hard to win the argument that the water bird under discussion is indeed a swan, now that the members lack the essential information available only from the weak members. In order to address this situation in PISA, the chairperson leaves the decision whether to keep the weakest members of a group or not to the group's leader. For simplicity it was assumed that the leader would keep all the group members throughout the dialogue even if some of these members have not contributed to the dialogue for a number of rounds.

## 7.3. Summary

This chapter discussed some of the issues in relation to the PISA Framework for multiparty "*Arguing from Experience*" dialogues. In particular, the strategy model for individual players (agents), the process by which groups of individual players could be formed, and possible extensions to the role of the chairperson. A two-tier strategy model was discussed, and three basic strategies were derived from this design, an example was given to illustrate the effect of the participants' strategies on the form of the argumentation tree and on the dialogue output. However, the advocated strategy model considered individual players only, regardless of whether they were members of some group or not. To further enhance the promoted strategy design, the structure of groups composed of two or more individual players was discussed in detail, and two types of groups were defined according to the strategies of their members. A leader identification process was also suggested for each type, along with a ranking of the strategies of the individual players to facilitate this selection process.

The group leader was given authority over the moves suggested by other members of the group. This authority meant that the overall strategy model of the group follows that of the leader, but still benefits from the experience, and to a lesser degree, the strategy of each other individual participant in the group.

Another interesting question, with respect to the promoted PISA Framework, that was not answered in this chapter is:

> *Should the chairperson be involved in the PISA dialogues? And if the answer is yes then what are the limits of such involvement?*

This question raises the issue of the role of the chairperson in PISA games. In the previous chapter this agent had a neutral standpoint limited to the simple management of argument flow from the participants to the argumentation tree, together with some other administrative responsibilities. For reasons of space the discussion of this issue is given in a separate appendix (Appendix C). This appendix discusses some extensions to the role of the chairperson allowing it more control over the dialogue process itself. Consequently the chairperson will have a direct impact on the results of the dialogue games.

The following chapter will further establish PISA by presenting empirical evidence to demonstrate the nature of the underlying dialogues. The ability of PISA to produce coherent dialogues to classify cases from different domains will be examined via a series of experiments. These experiments will assume that a PISA dialogue is successful if the final result of this dialogue matches the correct classification of the case under discussion. An assessment of the overall operation of PISA will be made on the basis of these results.

# Chapter 7: Advanced Issues in PISA

This chapter continues the discussion about the design of the PISA Framework. The basic structure and the main functions of PISA were described in the previous chapter. Nevertheless, the suggested model "*so far*" still has room for improvements, in particular regarding the issues of participants' strategies and the formation of groups of individual players. The structure of PISA suggested in the previous chapter raises two questions:

> *What is the strategy design most suited for the purposes of multiparty dialogues produced in PISA?*
>
> *How can groups of individual players be constructed most effectively?*

The first question concerns the issue of strategies and tactics in PISA. A discussion of the PISA strategy problem and the advocated solution is given in Section 7.1. The second question relates to the design structure discussed in the previous chapter, having modelled the participants in PISA dialogues as either individual players or groups of players sharing the same objective. Section 7.2 embarks upon the structure of such groups, mainly the arrangements related to the decision making process within each group. Section 7.3 concludes this chapter with a summary. The analysis given in this chapter will provide examples of the issues investigated using classification scenarios based on the RPHA specification given in the previous chapter (Section 6.5).

## 7.1. Strategy Design for PISA

Chapter 4 proposed a layered strategy model for players taking part in PADUA dialogue games to facilitate the selection of the kind of move to be put forward at each round of the dialogue, and the content of this move. The proposed strategy model comprises four levels:

- Level 0: Defines the game mode.
- Level 1: Defines the players (agents) profiles.
- Level 2: Defines the strategy mode.
- Level 3: Defines some appropriate argumentative content depending on the promoted tactics.

The issue of strategy design for PISA dialogues is considerably different from that in the two-party PADUA dialogues, mainly because the dialogue game in PISA is more complicated than in PADUA. This complexity arises from the fact that, unlike PADUA, PISA dialogues often take place between more than two participants. A second difference is that PISA determines which party has won a completed dialogue game in a different manner from PADUA. In PISA it is not always the case that the last contributing participant wins: dialogues may continue for a number of rounds after a winning argument (represented by a legal move) has been placed. These two differences indicated that the strategy design proposed for PADUA cannot simply be applied in the PISA framework, and so some in-depth modification is required if they are to be considered suitable for PISA dialogues. Another point of distinction between PADUA and PISA, over the issue of strategy, is that although strategies are often designed for individual (players) agents, they are applied in the context of dialogues. In PADUA, dialogues are clear-cut two-party exchanges of speech acts. However, dialogues conducted in PISA involve an indefinite number of participants which might be individual players or groups of players. Thus, the strategy design for PISA should also account for this point. PADUA strategies are intended for individual players participating directly in a dialogue game, while in PISA individual players may not be involved directly in the ongoing dialogue, but instead they may engage in an inner dialogue within their own group, the result of which is determined by the leader of the group. The issue of strategy design for groups is further explained in Section 7.2.

Acknowledging the above differences means that any appropriate strategy model for PISA needs to take into consideration the following points:

- *The status of the argumentation tree.* Participants can incorporate different views of this status in their own strategies. Each agent can consider as many, or as few previous moves (tree nodes), in order to make a decision with respect to the best next move. By taking more previous moves into their consideration, participants can potentially plan their next move better than those which consider (say) only the moves played in the last round.
- *Whether the player is a member of a group or not.* It is only when an individual player (agent) is the only advocate of its own "*view*" that this player is the master of its own moves. Where two or more agents advocate the same "*view*", then they are subjects to the decision making process of their assigned group. The issues related to the strategy design within each group are discussed in detail in Section 7.2.
- *Whether the participant (player or group) has to participate in a certain round or not.* As discussed in the previous chapter, in PISA, participants are not obliged to take part in each single round of the dialogue game (turn skipping was discussed in the previous chapter). PISA also allows for participants currently winning the game to skip rounds as long as their position is not undermined by the other participants.

Strategies in PISA should, however, maintain the four levels from PADUA's strategy model. In this sense PISA extends and builds upon the basic strategy outlines for PADUA. Taking all the discussed points into consideration, a six-level strategy model was designed for individual player agents taking part in PISA dialogues. Figure 7.1 illustrates this model. The proposed six levels are divided into two tiers, the *lower tier* encapsulates the strategy model inherited from PADUA, while the *upper tier* provides the scope through which PISA players deduce their next moves with respect to the argumentation tree.

The proposed strategy model works as follows:

- The lower tier encapsulates the same four-level layered strategy model used for PADUA (Section 4.2). However, the notion of *agent profile* is slightly altered to accommodate multiparty dialogues. PISA players applying an agreeable profile may either try to agree with all the other participants or with a pre-specified group of participants. For simplicity it is assumed that when applying an agreeable agent profile, the players will attempt to agree with all the moves represented by the argumentation tree leaves. The issue of agreeing with a subset of the moves presented by these leaf nodes is further discussed in Sub-section 7.1.4.

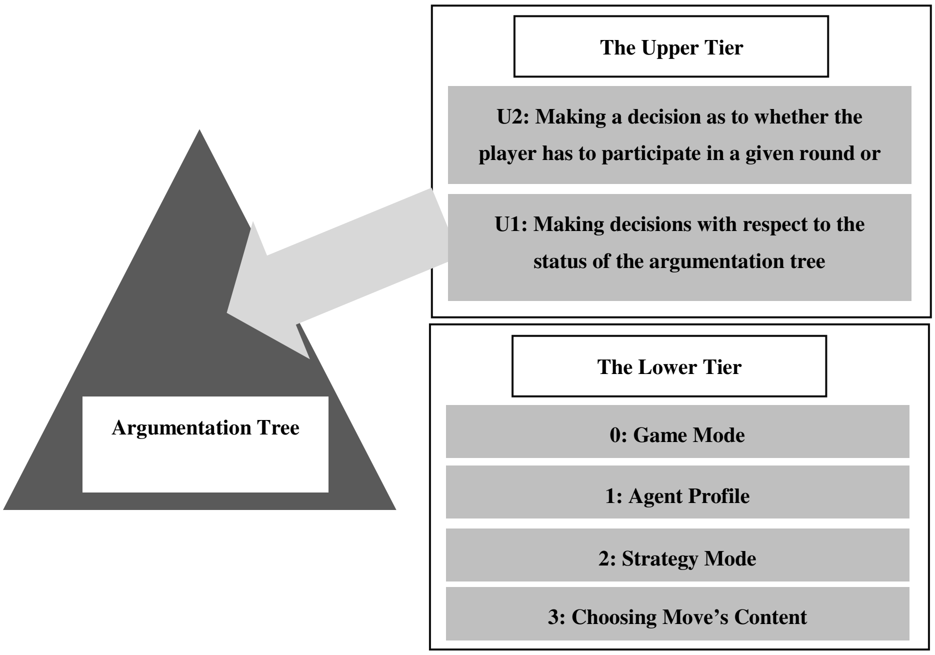

The Upper Tier
U2: Making a decision as to whether the player has to participate in a given round or
U1: Making decisions with respect to the status of the argumentation tree
The Lower Tier
0: Game Mode
1: Agent Profile
2: Strategy Mode
3: Choosing Move's Content
Argumentation Tree

**Figure 7.1. The two-tier strategy design.**

- The upper tier identifies the manner by which PISA players infer what move to play next from the current status of the argumentation tree. This tier comprises two strategy levels (Figure 7.1):
  - **Level U2**: defines whether the player "*has to*" participate in the current round of dialogue or not. Different strategies can be derived from this level. These strategies are explained in Sub-section 7.1.1.
  - **Level U1**: defines the process by which PISA players choose their next moves with respect to the argumentation tree. The argumentation tree encompasses details of the moves played thus far in the dialogue game, and the attacks relation amongst these moves. By consulting this tree, players can base their decision on a number of issues with relation to their next moves, such as: which opponent to attack next and which speech act to use.

Level U1 of the layered strategy advocated promotes three different modes for deducing the next moves from the current status of the argumentation trees:

- *Full Tree Inference Mode*: enables the derivation of strategies using a full view of the argumentation tree. Players can select their next move on the basis of their interpretation of the whole argumentation tree (their interpretation of the dialogue thus far).
- *Leaf Nodes Inference Mode*: leads to strategies with a limited view of the argumentation tree. Such PISA players consider only the leaf nodes of the argumentation tree (undefeated moves/attacks) when making decision about what moves to play next.
- *One Leaf Node Only Inference Mode*: from which all the derived strategies are basically two-party PADUA-style strategies.

Recall from Chapter 4 that a strategy function for players (agents) taking part in games governed by the PADUA protocol was identified in Section 4.2. This function was called $Play_a$ where $a \in A$ is a given player agent. Another version of this function can now be identified for the purposes of PISA strategies. Individual players taking part in PISA dialogues may use this function to select their next moves. For each agent (player), a∈ A, $Play_a$ is defined as follows:

$$Play_a : M_{poss} \times R_{poss} \times D_{current} \times S_a \times Tactics_a \rightarrow M_{poss}$$

For the purposes of PISA only $S_a$ (the *Strategy Matrix* for the given agent) is changed such that $S_a$ = [$havePart_a$, $tm_a$, $gm_a$, $profile_a$, $sm_a$]. Where: (i) $\text{havePart}_a$ identifies if the *a* has to take part in the next round or not, (ii) $\text{havePart}_a \in$ {true, false}. $tm_a \in TM$ is the tree inference mode, where *TM={full, leaves, one leaf}*, (iii) $gm_a \in GM$ is the game mode, where *GM* = {*win*, *dialogue*}, (iv) $profile_a \in$ *Profile* is the player profile, where *Profile* = {*agreeable*, *disagreeable*}, and (v) $sm_a \in SM$ is the strategy mode, where *SM* = {*build*, destroy}. Recall, $Tactics_a$ is the tactics matrix including the move preference and the best move content tactics. These tactics are explained in the context of each possible strategy in the forthcoming sub-sections.

### 7.1.1. Three Sub Strategies for PISA

A number of different strategies can be derived from the two-tier strategy model discussed in Figure 7.1 according to the values given to each of the parameters of $S_a$ of the strategy function $Play_a$ defined above. Three *Basic Strategies* were derived from the given model in relation to level U2 from the upper tier. These strategies form a basis from which other sub strategies are built as shall be discussed in the following paragraphs. Each strategy makes use of a set of tactics similar to that identified in Section 4.3. Note that strategies are numbered from S1 to SN, and where appropriate sub-strategies of a strategy are indicated using SK-M, for example S1-1, and so on.

**(S1) Attack Whenever Possible Strategy**

The idea behind the S1 strategy is simple: PISA players following this strategy will attack any opponent they can identify "*whenever possible*". Here, "*whenever possible*" means that players will attack their opponents whenever they can mine a suitable attack move from their background datasets, regardless of whether they need to do so or not. This strategy enhances the players' chances of winning the game by being as aggressive as possible, based on the assumption that if they attack even when they do not need to, and/or when their attacks are blindly directed against random opponents, they may win the game by undermining the arguments proposed by as many opponents as possible. Based on Level U1 of the upper tier of the advocated strategy model for the PISA framework, three sub strategies could be derived from *Attack Whenever Possible Strategy* along the three *Inference Modes*. Each mode will make use of a different opponent identification process:

- **Blind Attack Whenever Possible Sub-strategy (S1-1):** Here, individual players will attempt to arbitrarily attack any of the undefeated previous moves (leaf nodes); thus the opponent identification process is random (blind). This sub-strategy applies a *One Leaf Node Only Inference Mode* by which the attention of players will be focused on one and only one leaf node until a successful attack could be made against one of the argumentation tree's leaf nodes. Here two sub-sub-strategies are distinguished along the *Strategy Mode* element as inherited from PADUA:

- *Blind Attack Whenever Possible by Proposing new rules (Build Mode) - (S1-1-1):* Here, players will attempt to propose new rules whenever possible, therefore increasing their chances to win the game by proposing as many rules as possible, regardless of against which opponents they are directed. The idea is that, proposing high confidence rules will have the same effect regardless of which opponents they are directed against.
- *Blind Attack Whenever Possible by undermining the opponent (Destroy Mode) - (S1-1-2):* Here, players will plan to win the dialogue game by undermining their opponents' proposals whenever possible. The idea is that if one player manages to undermine all the proposals played by all the other players then this player wins the game. Of course this is not always feasible, particularly when other players are following different strategies.

Note that following a build or destroy strategy means that the player will try to apply this strategy against all the possible leaf nodes, before turning to the opposite strategy mode (e.g. destroy if the original strategy mode is build); instead of using build then destroy (or the other way around) tactics against each leaf node in order.

- **Focused Attack Whenever Possible Sub-strategy (S1-2):** Directs the players' attacks according to some ordering of the identified undefeated previous moves (leaf nodes). Thus, it promotes a *Leaf Nodes Inference Mode*, and therefore targets the next moves against the most appropriate undefeated previous move. Here the manner by which each player orders the leaf nodes is significant and can be achieved in various ways. The strategy model used here adopts a simple, yet effective, ordering: the agents will attempt to defeat any leaves representing the most threatening direct attacks against moves they have previously played. Such attacks are identified, at this level, as the leaf nodes directly attacking moves placed by a particular individual player (or the group it belongs to). If no such moves are found, or if the player has failed to defeat any of the direct attacks against its proposals, then it will try to apply S1-1. It is assumed this sub-strategy separates the leaf nodes of the argumentation tree into two groups one comprising the direct threats and the other containing the rest of the leaf nodes, no further ordering is applied on the leaf nodes in each group. Here also two sub-sub-strategies are distinguished:

- Focused Attack Whenever Possible by Proposing new rules (Build) - (S1-2-1)
- Focused Attack Whenever Possible by undermining the opponent (Destroy) - (S1-2-2).

These two types are similar to the ones discussed above (S1-1-1 and S1-1-2), the only difference being that they order the argumentation tree leaves prior to try to attack them.

- **Flexible Attack Whenever Possible Sub-strategy (S1-3):** Represents the most sophisticated of the three sub strategies derived from S1. Here individual players adopt a focused strategy similar to the one discussed above. However, instead of being restricted to a build or destroy strategy mode, players may choose whether to undermine an existing leaf node (undefeated move) or to propose a counter attack against this node. The switch from build to destroy mode, and the other way around, depends on the inference mode (Level U1) of the applied strategy. Two sub-sub-strategies are derived:
    - *Leaf Nodes Inference Flexible Attack Whenever Possible - (S1-3-1): D*educes next moves from the set of the previous undefeated moves (leaf nodes). Here besides distinguishing between most threatening direct attacks and other leaf nodes. PISA players apply some ordering on the most threatening direct attacks, and then try to defeat these attacks regardless how this is done (build or destroy). For the purposes of this thesis, this sub-strategy assumes a simple order of direct threats: Green attacks are ordered in a descending order according to their confidence followed by the blue attacks ordered in an ascending order according to their confidence.
    - *Full Tree Inference Flexible Attack Whenever Possible - (S1-3-2):* Promotes a full tree perception with the intention of determining an

order by which the player should attempt to attack its opponents. Figure 7.2 suggests an outline of such strategy. In this figure ArgT.GreenConfidence represents the current value green confidence, nodes(value) return a set of nodes which confidence=value, leafs(node N) returns the leafs nodes of the sub-tree which root =N. Dominant Blue means that the players has the highest number of blue nodes on the tree

```
Input: The Argumentation Tree ArgT.

Try proposing an AR with confidence > green confidence; direct
this move against the leaf node with the highest confidence.
else
 if ∃ leaf node LN : confidence(LN)==ArgT.GreenConfidence then
 attack LN.
 else
  for each N ∊ nodes(ArgT.GreenConfidence): colour(N)=Green do
  if ∃ leaf node LN ∊ leafs(N) then attempt to attack LN such
  that the colour of the N after attack = red.
  else
   if adding a blue node to the argumentation tree make the
   player dominant blue then attempt to play a blue move.
   else switch to S1-3-1.
```

**Figure 7.2. The proposed S3-2 strategy.**

### (S2) Attack Only When Needed Strategy:

The Attack Only When Needed strategy allows a player to choose whether it is necessary to take part in the current round of the dialogue game or not. Recall that participants taking part in PISA dialogues can choose to not contribute for pre-determined number of rounds without being forced to leave the dialogue. Also, participants in a winning position may not need to take an active part in the dialogue as long their position holds. Thus, the players will need to attack only when all their (past) moves have been successfully defeated by other participants; or when their attempts to undermine the proposals of all the other participants have failed (when any of the other participants puts forward a legal move). Here also three sub-strategies were derived according to the same criteria applied with the attack whenever possible strategy. However, players using these sub-strategies prefer not to contribute as long as they are in a wining position. Once this position is compromised (defeated) the players switch immediately to an underlying S1 sub-strategy. The advocated strategy model distinguishes between two types of "*wait and see*". The first is the *Build Mode Wait and See* where the player will only attack when it no longer has any green nodes on the tree, by proposing a new rule (or by advancing any other equivalent move), otherwise the player would rather "*wait and see*". The second is *Destroy Mode Wait and See* where the player would rather "*wait and see*" as long as there are either: no green nodes on the tree, or all the blue nodes belong to this player. According to these criteria S2 is further divided into the following sub-strategies, corresponding to the variants proposed for S1:

- Blind Attack Whenever Needed Sub-strategy - (S2-1)
  - Blind Attack only when needed by proposing rules - (S2-1-1)
  - Blind Attack only when needed by undermining the opponent - (S2-1-2)
- Focused Attack Whenever Needed Sub-strategy - (S2-2)
  - Focused Attack only when needed by proposing rules - (S2-2-1)
  - Focused Attack only when needed by undermining the opponent - (S2-2-2)
- Tree Dependant (Flexible) Attack Whenever Needed Sub-strategy (S2-3)
  - Leaf Nodes Inference Flexible Attack When Needed - (S2-3-1)
  - Full Tree Inference Flexible Attack When Needed - (S2-3-2)

**(S3) Attack to Prevent Forecasted Threat Strategy**

The Attack to Prevent Forecasted Threat Strategy anticipates forthcoming attacks against the participant's existing proposals; thus it is the most sophisticated strategy type in PISA. Here players deduce their best next moves based on the entire argumentation tree and use their own heuristics trying to calculate which of their previous moves may be the weakest link in their argument, and then either propose new rules to strengthen their position, or attack the positions of other participants before they have the chance to attack them. Figure 7.3 suggests an outline for this strategy. In this figure: nodes (Participant P) returns all the nodes played by P, player (node N) returns the participants who has played the move represented by N and colour (node N) returns the colour of N.

```
Input: The Argumentation Tree ArgT.

Try to propose a rule with a confidence higher than any other node on ArgT, direct this move against the highest confidence leaf node.
else
  Try to propose a rule with confidence higher than the green confidence; direct this move against the highest confidence leaf node.
  else
    if ∃ leaf node LN and confidence(LN)==ArgT.GreenConfidence then
    attempt to attack LN.
    else
      for each N ∈ nodes(ArgT.GreenConfidence) and colour(N)=Green do
        if ∃ leaf node LN ∈ leafs(N) and colour (LN) !=purple then
          attempt to attack LN such that the colour of the N after attack = red.
        else
          if adding a blue node to the argumentation tree make the player dominant blue then attempt to play a blue move.
          else
            identify the current blue dominant Participant PDB.
            if ∃ leaf node LN such that player(LN)== PDB and colour(LN)=blue then attempt to attack LN.
            else
              for each node N ∈ nodes(PDB) such that colour(N)=blue do
              if ∃ leaf node LN ∈ leafs(N) then
                attempt to attack LN such that the colour(N) after attack = red.
switch to (S2-3-2).
```

**Figure 7.3. The proposed Attack to Prevent Forecasted Threat strategy.**

### 7.1.2. Integrating the Strategy Model in the PISA Application

The three broad strategies discussed above (S1, S2 and S3) are all implemented within the PISA Framework Application (Section 6.5). Recall from the previous chapter that the user could enter some information when adding a new player.

The following describes the sort of information required when adding a new *Player Agent*, in particular the input parameters defining the new player's strategy. First the user has to select some dataset for the new player. This is done by browsing through the user directories and selecting the required dataset. Secondly, the user can decide which strategy this player will follow in the PISA dialogue game. The application provides default value of the new player's strategy (S1-1-1) should the user wish to skip this step. The user has to select values for the following: (i) one of the three strategies S1, S2 or S3; (ii) the strategy mode (*build* or *destroy*) and (iii)The inference level (One Leaf, Leaf Nodes or Full Tree) for the new player. By changing these settings the user has the option to select different strategies. PISA applies a disagreeable agent profile as the default profile for its player agents, as this framework was originally intended to model multiparty persuasion from experience rather than deliberation. However, this default value can be changed for any Player Agent to agreeable using the "*advanced*" options available for users, as discussed below. Also the ARM parameters such as confidence and support thresholds must also be fixed, the PISA Framework Application default values are confidence =50% and support =1%. Should the user wish to change the ARM parameter values or the agents' profiles/their game mode then they can use the "*advanced options*" button in the interface to access a special window and alter these values.

### 7.1.3. Example: Strategy in PISA

This section illustrates the strategies identified above using a number of example dialogues produced by the *PISA Application* using the strategy parameters ($S_a$) discussed previously. These examples are drawn from the RPHA artificial benefits configurations previously applied in Sub-section 6.5.1, for the purposes of emphasizing the role of the players' strategies in PISA dialogues. Three examples are chosen to demonstrate this point, each representing a PISA dialogue between four individual Player Agents over the classification of an RPHA case. The input case is that of: *a 63 years old female applicant, who satisfies all the benefits condition and has served in the armed* *forces and has paid her contribution in the past five years*. This case should classify as entitled to priority benefits. The four players engaging in the following examples are referred to as PR (priority entitled), EN (entitled), PE (partially entitled) and NE (not entitled). These players are all disagreeable, and engage in the dialogues in game mode. The other parameters in their strategy configurations differ according to the examples setup. Each of the following examples will consider the first four rounds of the PISA dialogue between the four participants in order to illustrate the effects different strategies have on PISA dialogues. The details of the full length dialogues are give in Appendix B.

### PISA Strategy Example 1

Here each of the four players applies the same strategy: a *focused attack when possible strategy in build mode (S1-2-1)*. The chairperson invites the EN player agent to propose the opening rule (argument). Thus EN suggests the following association:

```
EN - Proposes a New Rule: The case has the following
features: 2000£<capital<3000£ and 15%<Income<20%.
Therefore this case should be classified as (Entitled).
With confidence = 67.54%.
```

The reader might find it helpful to refer to the completed argument tree shown in Figure 7.4 as the debate develops. The initial rule is attacked by the other three player agents in the second round, as follows:

```
NE - Counter Rule: The case has the following features:
60<Age<65 and 15%<Income<20% Therefore this case should
be classified as (Not Entitled). With confidence = 69%.
```

```
PE - Counter Rule: The case has the following features:
Contribution Y1 = paid and Contribution Y5 = paid.
Therefore this case should be classified as (Partially
Entitled). With confidence = 68.4%.
```

PR - Counter Rule: The case has the following features: Residency = armed forces and Contribution Y1 = paid. Therefore this case should be classified as (Priority Entitled). **With confidence = 75.4%.**

Note that NE uses the fact that the case under discussion is of a candidate whose age is between 60 and 65 years to attacks the EN argument, since all the males applicants in this age groups are not entitled to benefits; at this stage PR is ahead as it has the best un-attacked rule. In round three all four players make moves:

- EN, PE and NE proposes a new rule to attack the current best rule (the previous PR rule):

  EN - Proposes a New Rule: The case has the following features: Gender = female, 60<Age<65, 2000£<Capital<3000£ and 15%<Income<20%. Therefore this case should be classified as (Entitled). **With confidence = 78.6%.**

  NE - Counter Rule: The case has the following features: 60<Age<65, 15%<Income<20% and 2000£<Capital<3000£. Therefore this case should be classified as (Not Entitled). **With confidence = 75.99%.**

  PE - Counter Rule: The case has the following features: Contribution Y1 = paid, Contribution Y2 = paid and Contribution Y5 = paid. Therefore this case should be classified as (Partially Entitled). **With confidence = 76%.**

- PR attacks the previous NE move. Although this move in not necessary, yet PR followed its own attack when possible strategy and attacked this node because it could do so:

  PR - Proposes a Counter Rule: The case has the following features: Residency= armed forces, 15%<Income<20% and contribution Y1=paid. Therefore this case should be classified as (Priority Entitled). **With confidence = 76.2%.**

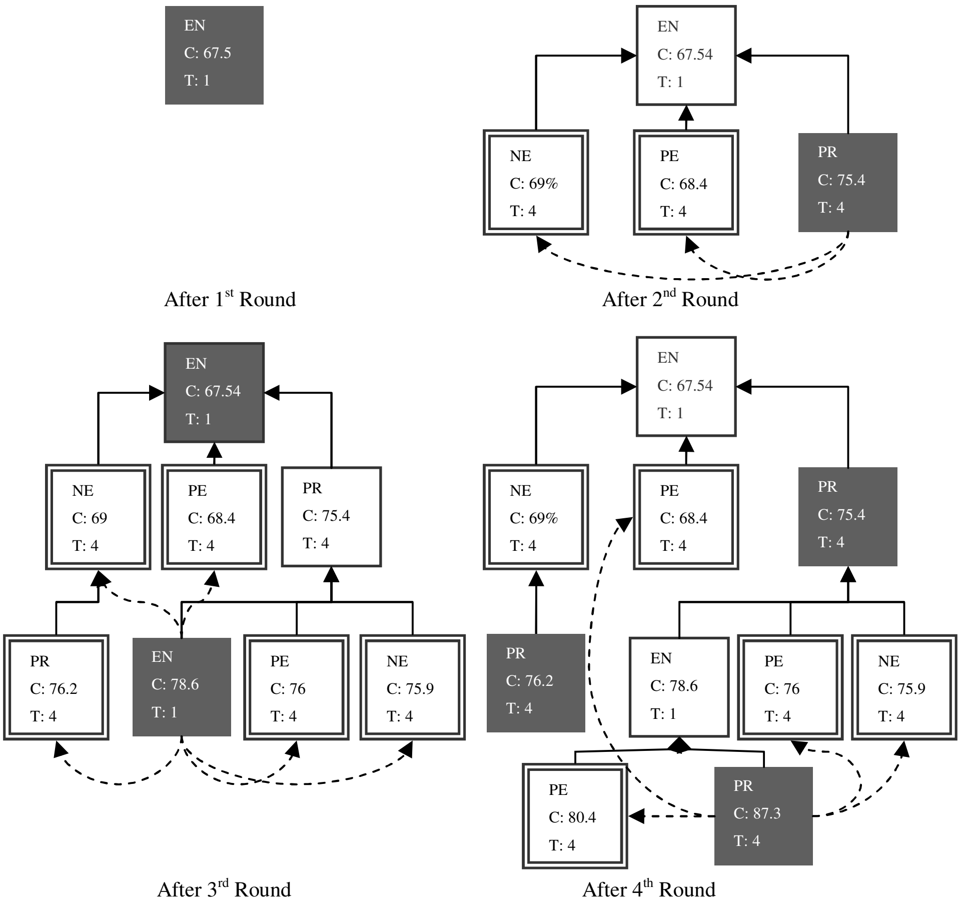


**Figure 7.4. The Argumentation Tree throughout PISA Strategy Example 1.** *Dark Grey=Green nodes, Double Lined=Purple nodes, and Single Lined=Red nodes.*

Now EN is back in the lead. Note that NE has again played a rule based on the age of the candidate to try and persuade the other participants to not issue any benefit to this candidate, this is the last move this participants would be able to play to stress this fact. In the fourth round NE and EN have no moves. The other two agents can, however, make moves against the winning position from last round as follows:

> PE – Counter Rule: The case has the following features: Contribution Y1 = paid, Contribution Y2 = paid, Contribution Y4 = paid and Contribution Y5 = paid Therefore this case should be classified as (Partially Entitled). **With confidence = 80.4%.**

```
PR - Counter Rule: The case has the following features:
Residency=armed forces, 15%<Income<20% and Contribution
Y1 = paid Therefore this case should be classified as
(Priority Entitled). With confidence = 87.3%.
```

The last round concludes this example. Note that PR has managed to win the dialogue, thus the resulting classification is correctly identified as priority entitled.

### PISA Strategy Example 2

In this second example the strategy configuration of PISA Example1 is changed such that agent players NE and PR apply a *destroy strategy*. Thus, NE will have more scope to critique other players' positions. More importantly, this example will reveal that by changing the strategy the output of the dialogue will drastically differ. The reader can refer to Figure 7.5 for the development of the argumentation tree for this example. The new dialogue commences in a similar manner to the previous example, as the chairperson invites EN once more to start the dialogue, and EN responds by playing the same opening rule from the previous example. This rule is attacked by the other three player agents in the second example, as follows:

- PE proposes the same rule it used in the same round in Strategy Example 1.
- NE distinguishes EN's argument from the first round by demonstrating that 60<age<65 only gives Entitled with a confidence of 19.9%.
- PR distinguishes EN's argument from the first round by demonstrating that contribution year 4 = paid and contribution year 5 = paid only gives Entitled with a confidence of 20%.
- Thus after the second round, PE is in the winning position rather than PR because PR is applying a destroy strategy instead of the build one it has used in the previous example.

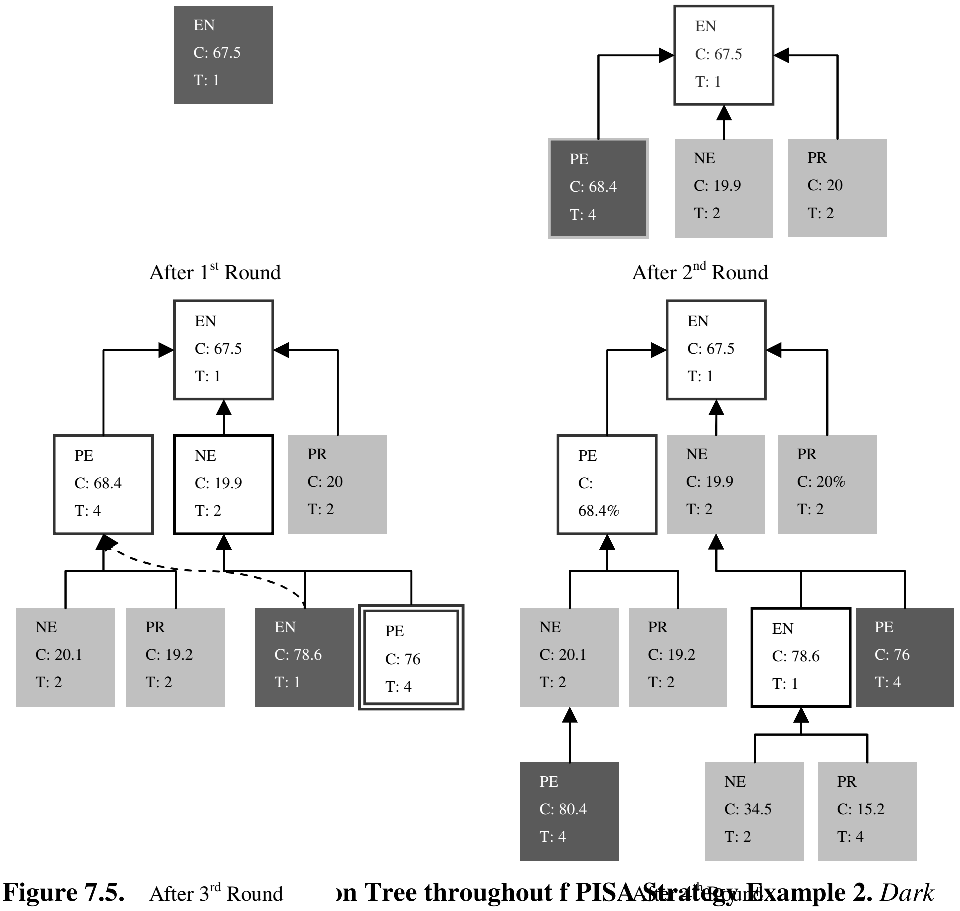


**Figure 7.5.** ›n Tree throughout f PISA Strategy Example 2. *Dark Grey=Green nodes, Light Grey=Blue nodes, Double Lined=Purple nodes, and Single Lined=Red nodes.*

All the four players take part in the third round. Note that PE participates in this round by playing the same move it played in the same round in the previous example, but this time it is directed against NE's move from round two. Again this move is not necessary, and if PE were applying an attack when needed strategy then it would have not played it. The other three participants play the following moves:

- EN proposes the same rule it used in the same round in PISA Strategy Example 1.
- NE distinguishes PE's argument from the last round by demonstrating that 60<age<65 only gives Partially Entitled with a confidence of 20.1%.

- PR distinguishes PE's argument from the last round by demonstrating that 20003<capital<3000£ only gives Partially Entitled with a confidence of 19.2%.

Now, EN is back in the winning position, in the same manner as the previous example. Note that NE has used the age group as a distinguishing factor this time rather than as a key attribute in arguing for advocating its own "*view*".

In the fourth round, EN has no moves. The other three agents can, however, make moves against the winning position from the last round as follows:

- PE proposes the rule it used in the same round in PISA Strategy Example 1.
- NE distinguishes EN's argument from the last round by demonstrating that 60<age<65, contribution year1 = paid, contribution year2 = paid only gives Entitled with a confidence of 34.5%.
- PR distinguishes EN's argument from the last round by demonstrating that paying contribution in years 1, 2, 3 and 4 only gives Entitled with a confidence of 15.2%.

Note that this example has evolved in a different manner to the previous one: by the end of round four, PE is winning the dialogue instead of PR. This is because PR is applying a destroy strategy rather than a build one. This emphasises the importance of the strategy mode in multiparty dialogues. However, the ultimate result of this dialogue is rather different to the one discussed here. The actual dialogue produced by PISA took ten rounds, the last two of which had no moves, and EN emerged as winner by the end of that dialogue. For reasons of space, the last four rounds were omitted from this example. However, the same result still applies: PR lost this game because it was not equipped with an adequate strategy.

### PISA Strategy Example 3

Let us now assume that the four players taking part in the above example apply more perceptive strategies, in relation to the argumentation tree, than the previous two examples as follows:

- PR and EN: apply S2-3-2 (full tree inference attack when needed) .
- PE applies S3 (preventing forecasted threat).
- NE applies S2-2-2 (destroy focused attack when needed).

These strategies will produce a different dialogue and a different argumentation tree from the ones discussed in the previous example. Figure 7.6 shows the development of the argumentation tree for this example. The dialogue commences in a similar manner to the previous two examples, with EN opening the dialogue with the sane initial rule as the previous two examples. This rule is attacked by the other three player agents in round two, as follows:

- First PE and PR propose the same counter rules they have presented in the same round in PISA Strategy Example 1.
- NE distinguishes EN's argument from the first round using the same rule from PISA Strategy Example 2.

Note that after the second round, PR is in the winning position as it has played the rule with the highest confidence so far. Only three players take part in round three; PR skips this round because it is in the winning position so there is no need for it to take part in the dialogue at this stage. The other three participants play the following moves:

- EN plays an increase confidence move against NE's move from last round:

```
EN - Increases the confidence of a previous rule by
stating that the case has the additional features:
Contribution Y1 =paid and Contribution Y2= paid.
Therefore this case should be classified as (entitled).
With confidence = 79.1%.
```

- NE distinguishes PR's argument from the last round by demonstrating that 60<age<65 only gives Priority Entitled with a confidence of 23.4%.
- PE distinguishes PR's argument from the last round by demonstrating that 2000£<capital<3000£ only gives *priority entitled* with 3.22% confidence.

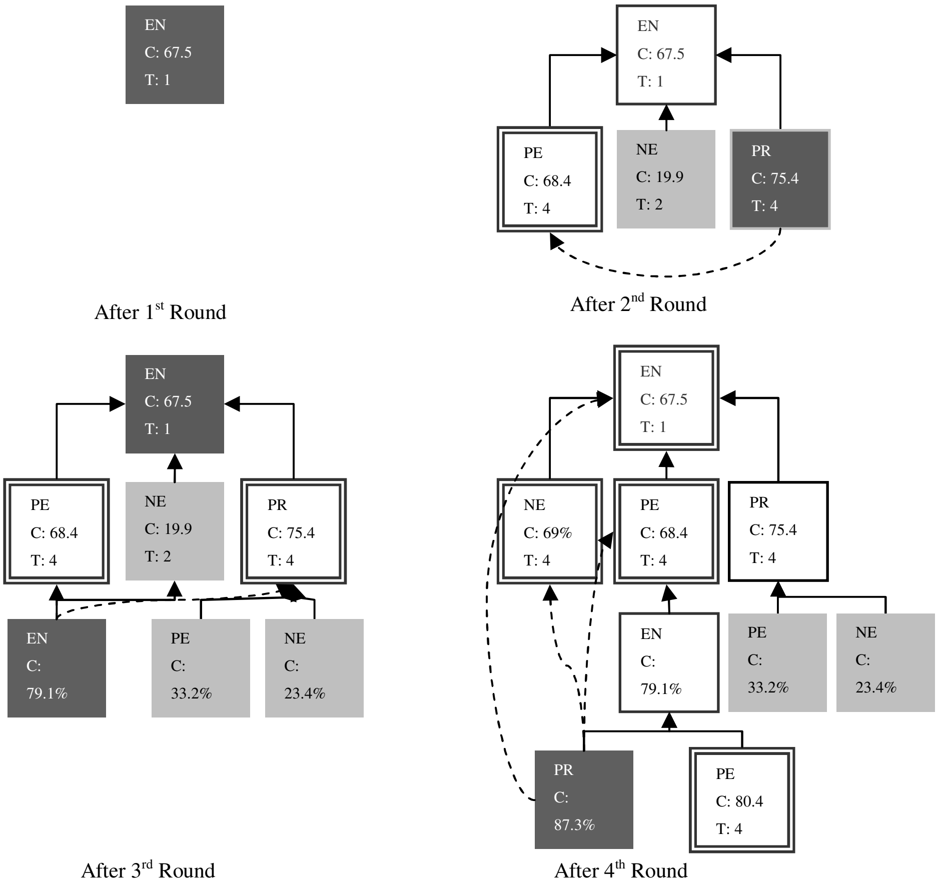


**Figure 7.6. The Argumentation Tree throughout PISA Strategy Example 3.** *Dark Grey=Green nodes, Light Grey=Blue nodes, Double Lined=Purple nodes, and Single Lined=Red nodes.*

Now, EN is back in the winning position, in the same manner as the previous two examples. However, the current example differs from the previous:

- PE has played a distinguishing move rather than a counter example move as it has forecasted that such a move is better than playing a build move, anticipating that the other participants may play moves with better confidence than the counter attack move it has mined against PR's position.
- NE has again used one attribute from the case under discussion to undermine the argument of the player in the winning position.
- EN has chosen to direct its move against NE's move from round two because, according to its strategy, this is better than playing a proposing new rules move against one of the other two nodes on the argumentation

tree, because it thus managed to defend its original position and attack all the other players' positions with one move.

Only two players take part in the fourth round. EN does not contribute to this round because it has the winning position, and NE has no more moves. The other two players attack EN's move from round three by counter attacking it using the same moves they have played previously in the same round in PISA Strategy Example 1. Note here that PR managed to gain a win in this dialogue because it has chosen d its move using larger number of previous moves that in the previous two examples.

### 7.1.4. Discussion

This section has examined some of the issues related to strategy design for the individual players (agents) engaging in multiparty "*Arguing from 'Experience*" dialogues within the PISA framework. The suggested two-tier strategy model provides PISA players with a range of different possible strategies varying in complexity, in particular regarding the manner in which the players make use of the argumentation tree. The above discussion can be reinforced with two additional points. The first of which considers the issues of "*agreeing with other participants*" in multiparty "*Arguing from Experience*" dialogues. The second point relates to the issue of "*temporary coalitions"* between different participants against one particular opponent.

With respect to the first point it was assumed previously in this chapter that agreeable PISA players will try to agree with all the moves represented by the argumentation tree leaf nodes. In other words, these players will attempt to agree with all the arguments that have not yet been defeated, and then to launch their attacks only against arguments they could not agree with (because no adequate ARs could be mined from the players' datasets). Also one player may prefer, for strategic reasons, to agree with certain other participants and not with the rest. For these reasons, each PISA player maintains a list, referred to as the *to-agree-with list*, comprising the participants it will attempt to agree with during the course of the game rather than attacking. Such a list is composed on the basis of the discussion domain: One player may prefer agreement with participants advocating classifications adjacent[36] to the one it is advocating, rather than losing the dialogue to other parties. This style of agreeable profiles is referred to as "*Biased Agreeable Profile*", in order to distinguish it from the (fully) "*Agreeable Profile*" discussed above. The notion of "*Biased Agreeable Profiles*" is of importance in domains where there are a number of adjacent classifications. Take for example the RPHA fictional domain from the previous sub-section: PR may settle for the "*entitled to benefits*" classification proposed by EN rather than not getting any benefits or getting just partial benefits (as proposed by the other two players in the game NE and PE respectively). EN and PE on the other hand may settle for anything other than not getting any benefits, while NE will not prefer agreement with any of the other three participants.

The above notion of *Biased Agreeable Profiles* could be applied as a mechanism for *coalition formation*, in which a number of participants may attempt to "t*emporarily agree*" with each others for strategic reasons, in order to overcome stronger opponents In this case, a number of participants could form a "*temporary coalition*" by which they join forces and cease attacking each other for a limited number of rounds for the purposes of defeating the stronger opponent(s). Once this goal is achieved, say when the stronger opponent(s) drops out of the dialogue game, then the participants in the "*temporary coalition*" can break up and resume attacking each other as they would have done prior to forming the coalition. Note that "*temporary coalitions*" differ from "*Biased Agreeable Profiles*" in two ways. Firstly "*temporary coalitions*" are temporary, which means that once the goal of the coalition has been achieved the participants in the coalition have no reason to continue being in this coalition. Secondly participants in a temporary *coalition* cease attacking each other, while participants with "*Biased Agreeable Profiles*" try to avoid attacking participants in their lists if possible; also there is nothing stopping the participants in the players' "*to-agree-with*" lists from attacking these players.

[36] Adjacent classification here refers to a class value related to or close to the classification this particular participant tries to prove true. Alternatively an agent might choose to agree with all those agents that give a better (or worse) outcome to the claimant.

Equipping PISA players with mechanism to enforce temporary coalitions is an ambitious extension of the PISA Framework. However, a number of issues must be addressed, if a successful implementation of coalitions is to be brought together. Chapter 9 will give a summary of these issues, based on the above discussion, and provide directions to tackling them in future extensions of PISA.

## 7.2. Groups and Leadership in PISA

Recall from the previous chapter that individual PISA players advocating the same thesis (for example the same possible classification of the input case) are required to "*join forces*" and act as a single "*group of players*". Every group is allowed only one move per round. This restriction aims at simplifying PISA dialogues. The proposed notion of groups prevents individual players sharing the same objective from arguing without consulting each other and consequently causing contradictions amongst themselves or attacking each other. This may, however, lead to a situation in which the weaker parties (within the groups) are forced to withdraw from the game and the remaining stronger members no longer have sufficient shared experience to win. Group formation is automatic in PISA. When a new individual player joins a dialogue game, over a case from a particular domain, it has to make its objective clear. The player's objective represents the thesis this player proposes ($G_a$). The chairperson then decides if this new player should participate in the dialogue as an individual player, or should become a member of an existing group of other players which advocated thesis matches the one proposed by the new player. In each group, the members have to select a leader from amongst them. This leader will act as a representative of its group in the dialogue, and is usually the "*smartest*" and "*most experienced*" (the one with the largest amount of data available at its disposal) member of the group. Player's *smartness* relates to the strategy this player applies. Hence, the smartest member is the one with the most sophisticated strategy amongst the group's members, where strategies are ranked according to their level of understanding of the history and the process of the dialogue.

The leader guides the inter-group dialogue, and selects which of the moves suggested by the group's members, including the leader's move, is the best to be played in the next round. This inter-group dialogue is a variation of "*targeted broadcasting"*, in which only group members can listen to what is being "*discussed in the group*", while other participants are completely unaware of these dialogues. The leader can also redirect other members' moves against different opponents, or advise them to follow its own strategy, an act that makes the group benefit from the different strategies applied by its members and from the differences in their experience.

Group formation in PISA is a clear-cut process when compared with the work on group (team) formation in the literature on cooperation among intelligent agents (e.g. (Kinny et al, 1992), (Cohen et al, 1999), and (Ogston et al, 2005)). This is mainly because what matters for PISA players, is not achieving a complex task by distributing actions amongst the group members. Rather, the argumentation dialogue process is supposed to lead to a coherent classification of the cases under discussion. All the group's members perform the same task: mining the best possible argument in the context of the ongoing dialogue, according to their strategy and experience. The following sub-sections describe in details the different types of groups in PISA and the decision making process within each type.

### 7.2.1. Groups Types

The internal structure of groups in PISA varies greatly depending on the strategy and the experience of each of its members. Mainly, because in each group a decision making process takes place at the beginning of each round to settle on the best move to play (or whether it is better to not contribute) in this round. Such a process implies that a minimum level of discipline should be respected by the group's members. For this purpose, a particular member in each group is chosen as its leader, to facilitate this decision making process and ensure that the other group's members work in harmony to convince the other participants in the dialogue that the case under discussion classifies as advocated by the group.

Two factors are essential to each group: the strategy factor and the experience factor. The strategy factor concerns the strategies of the individual players in the group. In some cases, all the members may have incorporated the same strategy, while in others each member applies its own strategy and thus a strategy ranking is required in order to determine who is going to be the group leader. The second factor relates to the experience of the group's members, measured in relation to the size of the dataset in which this experience is stored. Thus, an individual player with (say) 1200 records in its database is considered more experienced than one with only (say) 600 records. This factor is necessary to the operation of the group as will be discussed later in this section.

Groups in PISA are divided into two types according to the strategy factor: Homogenous and Heterogeneous groups.

**Homogenous groups:** consist of a number of individual players which share the same goal and apply the same strategy (same type in the same mode, and using the same agent profile). However, each individual player may use its own confidence/support values. In such groups the most experienced player (the one with the largest background dataset) is chosen to be the group's leader. If two or more of the group members share the same level of experience then one of them is selected at random to represent the group. Once the leader is agreed on, the group members will attempt, in each round of the dialogue, to mine the best rules according to their strategy each from their own datasets. The leader will then select the best move according to the group agreed strategy. For example, if all the group's members have adopted a *build attack only when needed blind* strategy, then the leader will select the build move with the highest confidence to place forward in the dialogue. If no such move was suggested, the leader will promote a destroy move with the lowest accuracy.

**Heterogeneous groups'** members apply different strategies. Therefore a strategy ranking is applied to determine who is the "*smartest*" amongst the group members and thus best suited for its leadership. If two or more players happen to incorporate the smartest strategy then the most experienced one is selected for leadership. If they also have the same experience then one of them is selected at random. In heterogeneous groups, the leader has the authority to force its own strategy on the other players causing them to adjust their suggested moves to suit the leader's strategy. Thus the role of the leader in this type of group is more sophisticated than in homogenous groups. A more detailed account of leadership is given in the following sub-section. PISA applies the strategy ranking described in Table 7.1 to determine the smartest possible strategy. The advocated ranking does not take into account the differences in the Game Mode (level 0) or Agent Profile (level 1) of each strategy, when assigning a rank to a given strategy. Thus, these two levels are omitted from Table 7.1. The suggested ranking also assumes that the best possible strategy is S3, followed by S2 then S1.

| **Rank** | **Name** | **Strategy (S1, S2, S3)** | **Sub-Strategy** | **Strategy Mode** |
|---|---|---|---|---|
| **1** | S3 | S3 | - | - |
| **2** | S2-3-2 | S2 | Tree Dependent - Full | - |
| **3** | S1-3-2 | S1 | Tree Dependent – Full | - |
| **4** | S2-3-1 | S2 | Tree Dependent - Leaves | - |
| **5** | S1-3-1 | S1 | Tree Dependent - Leaves | - |
| **6** | S2-2-1 | S2 | Focused | Build |
| **7** | S2-2-2 | S2 | Focused | Destroy |
| **8** | S2-1-1 | S2 | Blind | Build |
| **9** | S2-1-2 | S2 | Blind | Destroy |
| **10** | S1-2-1 | S1 | Focused | Build |
| **11** | S1-2-2 | S1 | Focused | Destroy |
| **12** | S1-1-1 | S1 | Blind | Build |
| **13** | S1-1-2 | S1 | Blind | Destroy |

**Table 7.1. Suggested ranking of PISA strategies.**

### 7.2.2. The Role of the Group Leader

Having distinguished between two types of groups, and established the leader selection process according to each type, a more detailed account of the role of the leader of the group is now given. Once a leader has been selected, this particular agent will have authority over the other members of its group. This authority entitles the leader to perform the following tasks:

- The leader's most essential task, as far as the group is concerned, is to select the best move at every round of the dialogue, from the selection of moves suggested by the group's members. The leader often chooses the moves following its own strategy. This does not mean that the leader will select its own move all the time. Rather, the leader aims at selecting the best move from amongst the suggested moves. For instance, the move with the highest confidence. Here, the differences in the members' experiences will greatly influence the leader's decision: members with different experience will often promote different content for their chosen moves, even where all the members apply similar strategies.
- The leader can compel the more experienced members (if any) to act according to the leader's strategy. This happens on a round by round basis. If a more experienced member suggests one move, in a given round, and if the leader assumes that a similar move with a better confidence, or a move with a different speech act better matching the game context, could be produced by this player, then the leader can ask this player to attempt generating another move using the leader's strategy. The leader then compares the new move (the one produced using its own strategy parameters) against the old one (the one the player has initially suggested) and chooses the best move. Consider, for example, the case where one of the experienced players has suggested a destroy move (following its own strategy) distinguishing some previously undefeated move in the dialogue. Then the leader will ask it to produce a build move. If this player replies with a build move with a high confidence (say higher than the moves suggested by the other members) then the leader will discard this player's initial move, otherwise it will discard the new move. Information about the members experience and strategy is available to its leader, through a simple dialogue, by which the leader request these information from the group's members. Additional conditions are applied to ensure that the leader practice the above authority only when needed: If the experienced members of the group apply weak strategies, and where other members have failed to produce adequate moves.

- The leader can redirect moves suggested by the other members against opponents other than the ones they have chosen. For instance, if one member suggests an "*increase confidence*" move against one opponent (say for strategic reasons), then the leader may change this move to a "*propose new rule*" and directs it against another opponent (say because this opponent threatens the group more than the one originally picked upon by the group member). Here as well, the leader is allowed to redirect the members' moves only when redirection is more rewarding according to the leader's strategy than the original move.

Note that the group's leader is not fixed. It may change when a new member joins the group, or when the current leader leaves the dialogue, and therefore the group. In the first case, the current leader has to compare its strategy and experience with the newcomer. If the newcomer satisfies the leadership conditions better, then the current leader has to step down, allowing the newest member to become the group's leader. In the second case, when the current leader leaves the game, the group members have to select a new leader from amongst them in the same manner prior to the start of the game.

This possibility of changing the leadership, from one player to another, demands a careful consideration of the leader identification process, i.e. the process by which the group's members identify the leader and communicate with it. The problem of leader selection could be solved by adopting the standard technique of token passing as used in computer networks. See for example (Ambroszkiewicz et al., 1998) where the token is used as a sign of decision power amongst a team of software agents, so that a member of the team who has currently the token enjoys the exclusive authority to decide on the status of the team. PISA implements a technique similar to token passing, but instead of passing tokens from one member to another, the leadership is identified in PISA by a "*Leadership Unit*"; each group has one "*Leadership Unit*" residing with the current group leader. This unit enables one Player Agent to perform the leadership tasks described above. When a new leader is selected this unit is passed from the previous leader to the new one. The "*Leadership Unit*" simplifies the inter-group dialogue. At the beginning of every round, all the group members (including the leader) send their moves to this unit. The leader compares these moves against its own strategy and against the experience of each group member, before deciding which move to play next in the game and against which opponent.

### 7.2.3. Groups in the PISA Framework Application

The notions of groups and leadership discussed above have been integrated in the PISA Framework Application (Section 6.5). The application gives the user the option of adding any number of players in each of the groups identified by a joint possible classification advocated by all the group's members. Figure 7.7 provides a screen shot of the group formation process in the given application. The user should first select a group, from amongst the set of possible groups[37], such that each group corresponds to one possible classification of the domain. Recall that PISA requires a description of the dialogue game to be uploaded to the system by the user prior to the start of any dialogue about any case from that domain (Section 6.5). After selecting a group, the user could add any number of player agents to this group. Once the user has inserted the required number of players into the group, the software forms the group with the desired number of players.

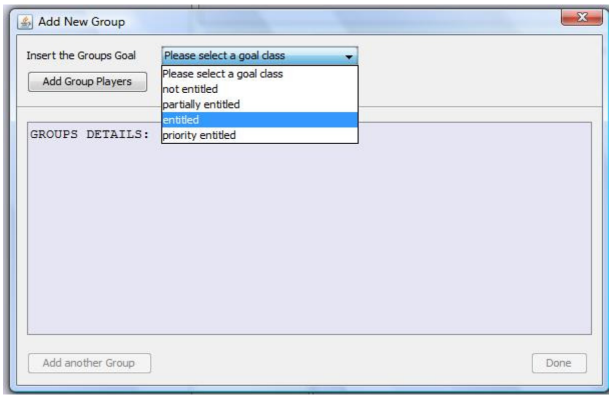
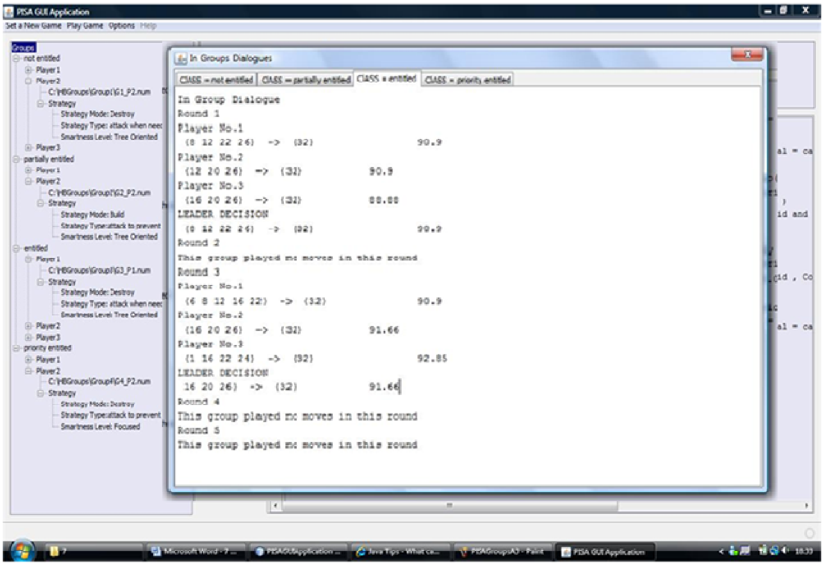


**Figure 7.7. Create a new group.**

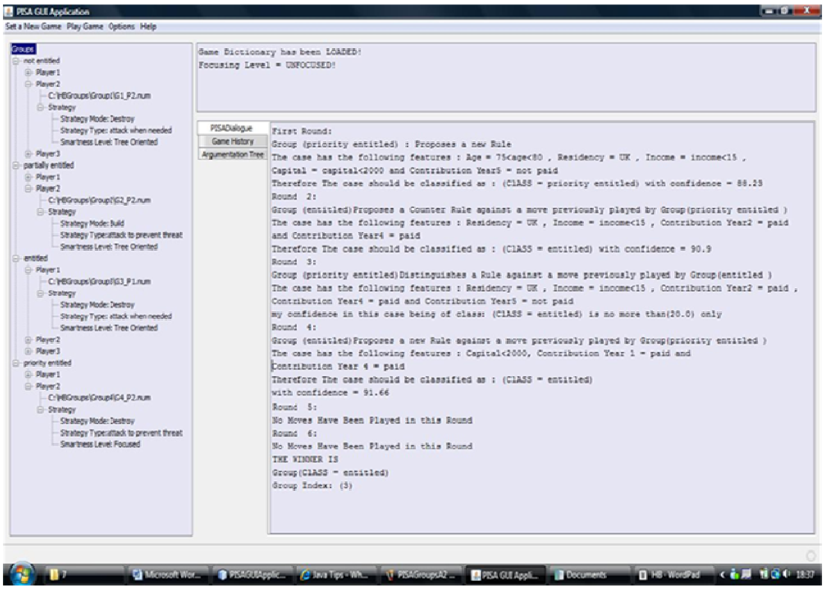
[37] Note that the list of all possible groups matches that of all possible classifications and is automatically generated upon loading the game description file (game dictionary) prior to start adding new players using the application.



```
Groups
  not entitled
    Player1
    Player2
      C:\HBGroups\Group1\G1_P2.num
      Strategy
        Strategy Mode: Destroy
        Strategy Type: attack when neec
        Smartness Level: Tree Oriented
    Player3
  partially entitled
    Player1
    Player2
      C:\HBGroups\Group1\G2_P2.num
      Strategy
        Strategy Mode: Build
        Strategy Type:attack to prevent
        Smartness Level: Tree Oriented
  entitled
    Player1
      C:\HBGroups\Group1\G3_P1.num
      Strategy
        Strategy Mode: Destroy
        Strategy Type: attack when neec
        Smartness Level: Tree Oriented
    Player2
    Player3
  priority entitled
    Player1
    Player2
      C:\HBGroups\Group4\G4_P2.num
      Strategy
        Strategy Mode: Destroy
        Strategy Type:attack to prevent
        Smartness Level: Focused
```

```
In Group Dialogue
Round 1
Player No.1
 (8 12 22 26)  ->  (32)                90.9
Player No.2
 (12 20 26)  ->  (32)         90.9
Player No.3
 (16 20 26)  ->  (32)         88.88
LEADER DECISION
 (8 12 22 26)  ->  (32)                90.9
Round 2
This group played no moves in this round
Round 3
Player No.1
 (6 8 12 16 22)  ->  (32)              90.9
Player No.2
 (16 20 26)  ->  (32)         91.66
Player No.3
 (1 16 22 24)  ->  (32)                92.85
LEADER DECISION
 16 20 26)  ->  (32)          91.66
Round 4
This group played no moves in this round
Round 5
This group played no moves in this round
```

**Figure 7.8. An example of inter group decision process.**



```
Groups
  not entitled
    Player1
    Player2
      C:\HBGroups\Group1\G1_P2.num
      Strategy
        Strategy Mode: Destroy
        Strategy Type: attack when needed
        Smartness Level: Tree Oriented
    Player3
  partially entitled
    Player1
    Player2
      C:\HBGroups\Group1\G2_P2.num
      Strategy
        Strategy Mode: Build
        Strategy Type:attack to prevent threat
        Smartness Level: Tree Oriented
  entitled
    Player1
      C:\HBGroups\Group1\G3_P1.num
      Strategy
        Strategy Mode: Destroy
        Strategy Type: attack when needed
        Smartness Level: Tree Oriented
    Player2
    Player3
  priority entitled
    Player1
    Player2
      C:\HBGroups\Group4\G4_P2.num
      Strategy
        Strategy Mode: Destroy
        Strategy Type:attack to prevent threat
        Smartness Level: Focused
```

```
Game Dictionary has been LOADED!
Focusing Level = UNFOCUSED!
```

```
First Round:
Group (priority entitled) : Proposes a new Rule
The case has the following features : Age = 75<age<80 , Residency = UK , Income = income<15 ,
Capital = capital<2000 and Contribution Year5 = not paid
Therefore The case should be classified as : (ClASS = priority entitled) with confidence = 88.23
Round  2:
Group (entitled)Proposes a Counter Rule against a move previously played by Group(priority entitled )
The case has the following features : Residency = UK , Income = income<15 , Contribution Year2 = paid
and Contribution Year4 = paid
Therefore The case should be classified as : (ClASS = entitled) with confidence = 90.9
Round  3:
Group (priority entitled)Distinguishes a Rule against a move previously played by Group(entitled )
The case has the following features : Residency = UK , Income = income<15 , Contribution Year2 = paid ,
Contribution Year4 = paid and Contribution Year5 = not paid
my confidence in this case being of class: (ClASS = entitled) is no more than(20.0) only
Round  4:
Group (entitled)Proposes a new Rule against a move previously played by Group(priority entitled )
The case has the following features : Capital<2000, Contribution Year 1 = paid and
Contribution Year 4 = paid
Therefore The case should be classified as : (ClASS = entitled)
with confidence = 91.66
Round  5:
No Moves Have Been Played in this Round
Round  6:
No Moves Have Been Played in this Round
THE WINNER IS
Group(ClASS = entitled)
Group Index: (3)
```

**Figure 7.9. The resulting dialogue (from example presented in Section 7.2.3).**

### 7.2.4. Discussion

Thus far, in this thesis, the processes by which groups of individual players with common objective (goal, classification) are formed, and a leader for each group is selected, have been established. The two suggested processes, however, may lead to situations in which the weaker parties within each group are ignored in favour of more powerful group members. In such situations, it would be undesirable for the chairperson to force the weaker parties to withdraw from the dialogue game, because the group would be deprived of the experience of these members. It could be the case, for instance, that only the weakest members of a group are aware of the fact that water birds with black feathers living in Australia could be swans. In this case the group would find it very hard to win the argument that the water bird under discussion is indeed a swan, now that the members lack the essential information available only from the weak members. In order to address this situation in PISA, the chairperson leaves the decision whether to keep the weakest members of a group or not to the group's leader. For simplicity it was assumed that the leader would keep all the group members throughout the dialogue even if some of these members have not contributed to the dialogue for a number of rounds.

## 7.3. Summary

This chapter discussed some of the issues in relation to the PISA Framework for multiparty "*Arguing from Experience*" dialogues. In particular, the strategy model for individual players (agents), the process by which groups of individual players could be formed, and possible extensions to the role of the chairperson. A two-tier strategy model was discussed, and three basic strategies were derived from this design, an example was given to illustrate the effect of the participants' strategies on the form of the argumentation tree and on the dialogue output. However, the advocated strategy model considered individual players only, regardless of whether they were members of some group or not. To further enhance the promoted strategy design, the structure of groups composed of two or more individual players was discussed in detail, and two types of groups were defined according to the strategies of their members. A leader identification process was also suggested for each type, along with a ranking of the strategies of the individual players to facilitate this selection process.

The group leader was given authority over the moves suggested by other members of the group. This authority meant that the overall strategy model of the group follows that of the leader, but still benefits from the experience, and to a lesser degree, the strategy of each other individual participant in the group.

Another interesting question, with respect to the promoted PISA Framework, that was not answered in this chapter is:

> *Should the chairperson be involved in the PISA dialogues? And if the answer is yes then what are the limits of such involvement?*

This question raises the issue of the role of the chairperson in PISA games. In the previous chapter this agent had a neutral standpoint limited to the simple management of argument flow from the participants to the argumentation tree, together with some other administrative responsibilities. For reasons of space the discussion of this issue is given in a separate appendix (Appendix C). This appendix discusses some extensions to the role of the chairperson allowing it more control over the dialogue process itself. Consequently the chairperson will have a direct impact on the results of the dialogue games.

The following chapter will further establish PISA by presenting empirical evidence to demonstrate the nature of the underlying dialogues. The ability of PISA to produce coherent dialogues to classify cases from different domains will be examined via a series of experiments. These experiments will assume that a PISA dialogue is successful if the final result of this dialogue matches the correct classification of the case under discussion. An assessment of the overall operation of PISA will be made on the basis of these results.

# Chapter 8: Empirical Observations (2) - Analysis of the Features of the PISA Framework

The previous two chapters have given a description of the PISA Framework for multiparty "*Arguing from Experience*" and addressed some of the issues and features associated with the promoted framework. In summary, PISA allows any number of participants to debate the classification of a given case, in accordance with given specifications, such as the participant's strategies. This chapter is intended to assess the underlying debates using a number of experiments, in which the resulting dialogues are considered successful if their output matches the desired classification. The analysis included aims at proving that PISA produces dialogues with reliable outcomes, allowing different parties to come to a conclusion with respect to a given case (a suitable classification). The general outline of this chapter is similar to Chapter 5, where the operation of the PADUA protocol was assessed. Here, a similar set of comparative experiments that were carried out using PISA are reported. Aside from providing an assessment of the process of multiparty "*Arguing from Experience*", the various experiments aim at establishing PISA as an effective classifier.

Most of the empirical experiments reported in this chapter were carried out using a similar approach to that used for evaluating PADUA (Section 5.1). Section 8.1 provides background information regarding these experiments. Sections 8.2 to 8.6 discuss the results of empirical studies implemented to examine the various aspects of PISA. Section 8.7 concludes with a summary. The nature of the reported experiments may be summarised as follows:

1. **The operation of PISA** as means to aid "*Arguing from Experience*" between any number of agents, and the nature of the underlying dialogues. Section 8.2 provides an empirical analysis of this operation.
2. **The operation of PISA as classifier** was one of the distinctive features to emerge from the promoted application of PISA. Section 5.3 provides a

comparative analysis of the application of PISA to a number of classification problems, in both *healthy* and *noisy* settings.

3. **The various features embodied in PISA:** The relation between the number of individual players in PISA groups and its operation is examined in Section 8.4. The advocated strategy model, associated with PISA is assessed in Section 8.5.

## 8.1. Experimental Design

This section describes the background to the evaluation reported in this chapter. Note that each experiment comprises a number of tests, each test focusing on a different aspect/feature of the subject matter of the given experiment.

### 8.1.1. The Included Datasets and Classification Paradigms

A number of multi-class real and artificial datasets were employed to provide PISA agents with sufficient data. Table 8.1 provides a summary of these datasets, all drawn from the UCI repository (Blake and Merz, 1998). For the purposes of evaluating the operation of PISA a discretised version of each of these sets was applied: the discretised datasets were obtainable by anonymous download from (Coenen, 2003). Some of the included tests also made use of a number of datasets drawn from the RPHA artificial benefits configurations previously applied in Section 6.5.1. A description of these datasets will be given upon referring to them. The chosen datasets display a variety of characteristics such as variable sizes and the inclusion of a mixture of data types, aside from being drawn from different domains. Most importantly, they include a diverse number of class labels, distributed in a different manner in each dataset, thus providing the desired variation in the experience assigned to PISA participants.

In order to fully assess its operation, PISA was compared against a wide range of classifiers including: RDT, IGDT, CBA, CMAR, TFPC, FOIL, CPAR and PRM. Additionally, PISA is evaluated against a number of ensemble methods. Section 8.3 makes use of the JBoost package (http://jboost.sourceforge.net) to implement ADABoost and ADABoost.M1 (Freund and Schapire, 1997). The latter is used with the multi-class datasets. This section also reports on comparisons with BrownBoost (Freund, 1999), as this algorithm has shown some robustness against noise (e.g. (McDonald et al., 2003)). A comparison using Bagging (Breiman, 1996) and MultiBoosting (Webb, 2000) as implemented in (Witten and Frank, 2005) was also undertaken.

| **Domain** | **Description** | **Exs** | **A** | **C** | **Classes dist. %** | **Best published accuracy (UCI)** |
|---|---|---|---|---|---|---|
| Annealing | Steel annealing data. | 898 | 39 | 6 | C1 (0.89), Class =2 (11.02), Class =3 (76.17), Class =4 (0), Class =5 (7.56), Class = U (4.45). | 96.6% (Yang et al., 1999). Euclidean metric for distance based learning. |
| Cars Evaluation | Derived from hierarchical decision model for evaluating certain cars. | 1728 | 7 | 4 | unacceptable (70.02), acceptable (22.22), good(3.99), very good (3.76). | 97.9% (Tan and Dowe, 2003). C5. |
| Flare | Each class counts the number of solar flares of a certain type that occur in a 24 hour period. | 1389 | 13 | 9 | C1 (84.31), C2 (10.51), C3 (2.88), C4 (1.44), C5 (0.65), C6 (0.29), C7 (0.22), C 8 (0), C 9 (0.07). | 83.5% (Li et al., 2004). DeEPs. Instance based lazy classifier. |
| Led 7 | Led display domain. | 3200 | 8 | 10 | C 0 (10.28), C1 (10.94), C2 (9.78), C3 (9.59), C4 (9.75), C5 (9.78), C6 (9.41), C 7 (9.81), C8 (10.22), C 9 (10.44). | 100% (Leung and Parker, 2003). Bagging using different voting methods. |
| Nursery | Derived from hierarchical decision model developed to rank applications for nurseries. | 12960 | 9 | 5 | Not recom (0.02), recom (2.53), very recommended (32.92), priority (31.2), special priority (33.33). | 99.04% (Li et al., 2004). DeEPs. Instance based lazy classifier. |
| Page Blocks | Contains information of all blocks of the page layout in a document. | 5473 | 11 | 7 | C1 (40.59), C2 (19.8), Class =3 (4.95), C4 (12.87), C5 (3.96), C6 (7.92), C7 (9.9). | 97.28% (Eschrich et al., 2002). Subsampling 25% ETS=5. |
| Pen Digits | Pen-Based Recognition of Handwritten Digits dataset. | 10992 | 17 | 10 | C 0 (10.40), C1 (10.40), C2 (10.41), C3 (9.6), C4 (10.41), C5 (9.6), C 6 (9.61), C 7 (10.39), C 8 (9.6), C 9 (9.6). | 99.35% . (Li et al., 2004). K-NN. |
| Waveform | CART book's waveform. | 5000 | 22 | 3 | C0(33.14), C1(32.94), C2(33.92). | 84.36% (Li et al., 2004). DeEPs. Instance based lazy classifier. |

**Table 8.1. Real-world datasets used with PISA.** *The columns are, in order: name of the domain, number of examples, number of attributes, the number of classes and the class distributions. Last column shows the best published accuracy according to the UCI Machine Learning repository (Blake and Merz, 1998).*

### 8.1.2. Evaluation Methodology

The evaluation methodology used for most of the included tests is the same as that reported in Chapter 5 in the context of PADUA. First, each of the given datasets was equally divided among a number of PISA participants corresponding to the number of classes in each dataset, such that each participant was given a random share of the dataset under consideration. Then a number of PISA dialogue games were played, the results of which were interpreted according to the underlying experiment and the type of the intended test. Unless stated otherwise, all the participants were directed to apply "*focused build attack whenever possible*" (S1-1-2) strategy. Note that the code used in these experiments is available for anonymous download from the author's webpage: **http://www.csc.liv.ac.uk/~maya/PISA_App.html.** For each reported evaluation the confidence level for generating the rules used by each participant was fixed by default to 50% and the support to 1%. With respect to the other classifiers, which all used a single dataset, each of them operated on the union of the participants' datasets (the original datasets); where applicable the same support and confidence threshold values were also used.

## 8.2. Evaluating the Operation of PISA

This section discusses the results of a number of experiments intended to analyse the process of multiparty "*Arguing from Experience*" as embodied in PISA. The reported experiments applied PISA using the real world datasets described in Table 8.1. Additionally, one artificial dataset, corresponding to the RPHA scenario (Section 6.5.1), was generated for the purpose of this study. Overall, the included datasets represent a diverse set of past experiences, thus providing a range of coverage suitable for experimenting with PISA. Four experiments were undertaken in order to investigate the operation of PISA. Each aimed at assessing different aspect of this operation:

- The first experiment evaluated the output of PISA dialogues by means of the accuracy of the resulting classification. A high accuracy indicated that the underlying argumentation process can successfully enable joint reasoning from experience amongst a number of different participants, each relying on their own experience.

- The second experiment evaluated the operation of PISA against that of PADUA.
- The third experiment provided an analysis of the features of the dialogues produced to come to decision regarding cases in each application domain.
- The final experiment aimed at exploring the relation between the accuracy of correct classifications and the size of the input dataset, using a number of Housing Benefits (RPHA) datasets to cover a range of data sizes.

The first experiment, as described above, involved applying a collection of Ten-fold Cross Validation (TCV) tests. This was achieved in the same manner as described in Section 5.2. Figure 8.1 (a) shows the average accuracy obtained from PISA, for each dataset. Figure 8.1 (b)[38] shows the Balanced Error Rate for each dataset.

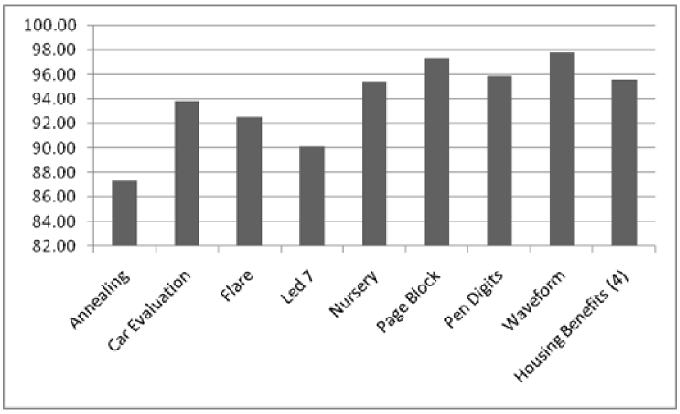


**Figure 8.1 (a). Accuracy of PISA TCV tests.**

[38] Balanced Error Rates (BER) were calculated, in each dataset, as follows: $BER = \frac{1}{C}(\sum_{i=1}^{C} \frac{F_{ci}}{T_{ci}+F_{ci}})$. Where C=the number of classes in the dataset, $T_{ci}$=the number of cases which are correctly classified as class $c_i$, and $F_{ci}$=the number of cases which should have been classified as $c_i$ but PISA has classified them otherwise. This formula for calculating BER was taken from (Mohammadi and Gharehpetian, 2008).

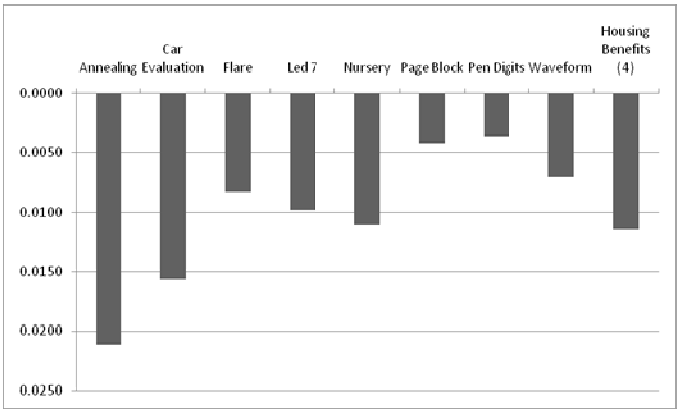


**Figure 8.1 (b). Balanced Error Rate (BER) of PISA TCV tests.**

As can be seen, PISA has achieved a high accuracy (above 90%) in all the considered domains, except for the Annealing dataset (87.31%), which indicates that the process of "*Arguing from Experience*" was productively utilised in PISA for a successful resolution of conflicts over the classification of cases in each of the included domains. Each participant mined the required arguments from their own dataset, with respect to their advocated classes, and effectively placed in the context of the underlying dialogue games. The reported fine performance of PISA encourage employing "*Arguing from Experience*" as a computerised means to enable software agents (entities) to reason on the basis of their accumulated experience. It also provides evidence to the reliability of the underlying dialectical process, for if these dialogues were misleading then the accuracy of the resulting accuracy would have suffered.

## 8.2.1. A Comparative Study of PISA and PADUA

Section 6.5 noted the possibility of applying PISA to facilitate two-party dialogues. Below, this utilisation of PISA is evaluated against that of PADUA. The results reported here emphasise that, when only two parties are to take part in an "*Arguing from Experience*" dialogue, PADUA is preferable to PISA, because it embodies a lighter application than PISA. The analysis of the behaviour of both approaches will, however, reveal some interesting observations, which are to also be discussed below. In order to compare both applications of "*Arguing from Experience*" two experiments were carried out:

- The first comprised a set of TCV tests in which PISA was applied using five binary datasets previously used to evaluate the operation of PADUA, the description of which was given in Section 5.1. The results were then compared to the ones obtained from PADUA.
- McNemar's test was then applied to compare the behaviour of PISA with the recorded behaviour of PADUA.

Figure 8.2 illustrates the result of the first experiment. The high level of accuracy (above 90%) reported in this figure indicates that PISA can conduct two-party "*Arguing from Experience*" dialogues as efficient as multiparty ones. Moreover, the average accuracy of PISA (97.76%) is only marginally worse than that of PADUA (98.03%). However, PADUA has achieved better accuracy than PISA for most of the datasets. These results indicate that the difference in the performances of the two approaches to "*Arguing from Experience*" is not substantial.

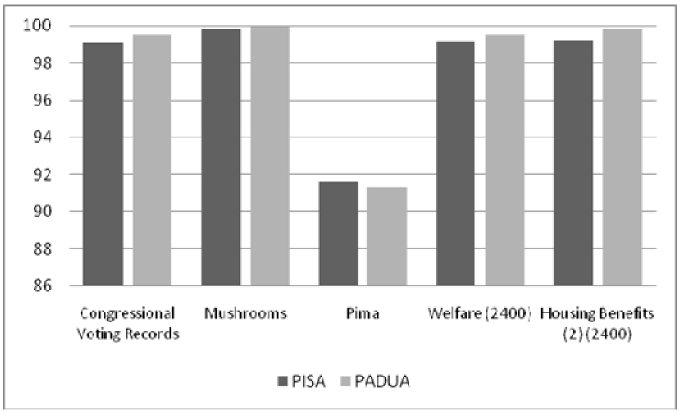


**Figure 8.2. Accuracy of PADUA and PISA using 2-class datasets.**

To emphasise the above point, the McNemar's test was applied to both PISA and PADUA using the five data sets from Figure 8.2. Table 8.2 presents the results of this test. From this table, it is evident that the difference in the performance of both methods is not significant (McNemar's and P-value). Nevertheless, a number of different "*mistakes*" were made by the two applications, in each of the datasets. This is because PISA has a different approach to identifying the role of the participants of each dialogue, other than the proponent-opponent setup of PADUA. As in the latter the proponent is fixed for each domain, while in PISA the proponent (participant initiating the dialogue) is randomly selected at the beginning of each dialogue. Also, PISA employs a different strategy model than the one used with PADUA. Thus the dissimilarity in the behaviour of both systems can be explained by the differences in the operation of PISA when compared with PADUA.

| Dataset | Congressional Votes | Mushrooms | Pima | Welfare | Housing Benefits (2 classes) |
|---|---|---|---|---|---|
| **Both Failed** | 3 | 1 | 4 | 3 | 2 |
| **PISA Failed** | 1 | 1 | 2 | 1 | 1 |
| **PADUA Failed** | 0 | 1 | 1 | 1 | 0 |
| **Both Succeeded** | 96 | 97 | 93 | 95 | 97 |
| **McNemar** | 1 | 0 | 0.33 | 0 | 1 |
| **P-value** | 1 | 0.48 | 1 | 0.48 | 1 |

**Table 8.2. Results of applying the McNemar's test to PISA and PADUA.**

The reported results indicate that PADUA is more suitable for two-party dialogues as it produces slightly better accuracy, and more importantly, because its application is lighter than that of PISA. This latter feature can be measured by the average number of rounds (dialogue length) each system requires to come to a decision regarding the classification of cases in a given dataset. Table 8.3 illustrates the average length of PISA and PADUA dialogues associated with the TCV tests reported previously. Note that the performance of PISA is more consistent, with respect to the standard deviation of the average number of rounds per each dataset. PADUA, however, produces shorter dialogues in most of the domains, and these shorter dialogues encourage the usage of PADUA, rather than PISA, to aid two-party "*Arguing from Experience*".

| Rounds | Congressional Voting | Mushrooms | Pima | Welfare | Housing Benefits (2 classes) |
|---|---|---|---|---|---|
| **PISA** | 14.3(4.7) | 12.5(2.6) | 8.6 (7.3) | 7.7(1.5) | 7.94 (1.7) |
| **PADUA** | 12.9 (5.9) | 13.7 (9.7) | 6.4 (7.4) | 7.2 (5.9) | 7.16(5.5) |

**Table 8.3. Average number of rounds PADUA and PISA take in each dataset.** *The numerical value between two brackets indicates the standard deviation.*

### 8.2.2. Discussion about the Length of the Dialogues

As stated previously, PISA allows for a number of participants to engage in a dialogue regarding the classification of some case. A detailed analysis of the characteristics of these dialogues provided confirmation of the mechanism by which PISA aids the process of "*Arguing from Experience*". Information about the average number of rounds taken by PISA to come to a conclusion with respect to binary classifications was given in the previous sub-section. An analysis of dialogue length in multi-class domains is given below. This analysis was intended to investigate the following:

- **The average length of the dialogues** measured by the average number of rounds PISA takes to come to decision regarding cases in given dataset.
- **The dialogue end-results**. Unlike PADUA, PISA dialogues have more outcomes than simply winning and losing. Some dialogues may end with a *green win* (all the undefeated green moves belong to one participant), or a *blue win* (all the undefeated blue moves belong to one participant). Additionally, some dialogues may end without a clear winner. This latter situation is referred to as a *tie*, an account of which was given in Chapter 6 and distinguished between two types of ties: *Strong* and *Weak Tie*s.

The following discussion covers both of the above commencing with the average dialogue length. Very short dialogues indicate that the argumentation process of PISA has not had a full effect on the resulting classifications, particularly considering that players taking part in the included experiments are all "*disagreeable*". Thus, a quick termination of the dialogue game, say after one or two rounds, points toward a problem in the underlying debate process: the players' failure to mining adequate arguments from their given datasets. This could arise either because these players were assigned insufficient experience, with respect to the support and/or confidence thresholds, or that these thresholds require modification. Very long dialogues, on the other hand, suggest that PISA may be rather inapplicable in the real world. Fortunately, this is not the case as exemplified in Figure 8.3. Note that the longest dialogues occur in the Annealing dataset, because the classes in this dataset are not evenly distributed amongst its records. Therefore, some participants have to rely upon distinguishing attacks should they attempt to stay in the game, which in turn contributes to prolonging the game because these attacks are easier to mine from the background dataset, due to the low associated confidence thresholds.

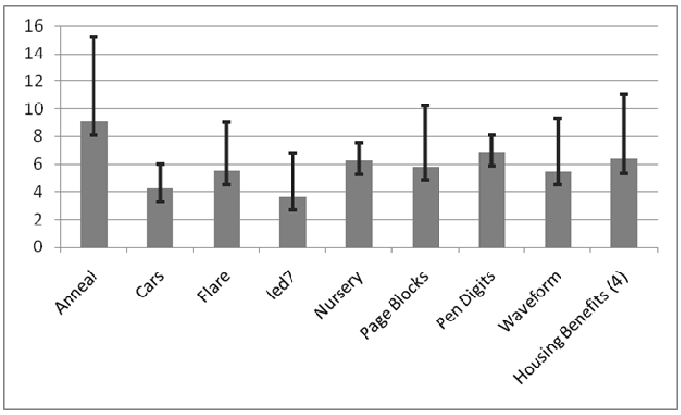


**Figure 8.3. Average number of rounds per domain.** *Error bars represent the standard deviation for each domain.*

The average percentage of the consequent end-results of the produced dialogues was provided for the analysis of the behaviour of PISA with respect to each dataset in the above TCV tests. Table 8.4 illustrates these results. A number of interesting observations can be made:

- The reported results indicate that the datasets associated with the largest portions of blue wins (Annealing and Nursery) are those with uneven distribution of the class values.
- 17.03% blue wins were detected in the Housing Benefit dataset. This is because the class values in this dataset are closely related to each other, and only minor differences in certain values promote each class.
- In contrast, the Led 7 dataset has produced the largest portions of ties (particularly "s*trong ties*"), because this dataset contains ten possible classes evenly distributed amongst its records. Thus, not only each participant is allocated one tenth of the dataset, but has to compete with nine other participants, each given a similar size dataset.

| End-results (%) | Green Wins | Blue Wins | Strong Ties | Weak Ties |
|---|---|---|---|---|
| **Annealing** | 79.1 | 18.54 | 1.24 | 1.12 |
| **Cars** | 91.76 | 6.12 | 1.59 | 0.53 |
| **Flare** | 87.59 | 9.49 | 1.168 | 1.75 |
| **led7** | 83.06 | 11.75 | 4.38 | 0.81 |
| **Nursery** | 77.65 | 20.52 | 1.06 | 0.77 |
| **Page Blocks** | 90.8 | 8.81 | 0.29 | 0.09 |
| **Pen Digits** | 88.63 | 9.63 | 0.78 | 0.97 |
| **Waveform** | 92.7 | 6.1 | 0.98 | 0.22 |
| **Housing Benefits (4 Classes)** | 81.29 | 17.03 | 1.15 | 0.65 |

**Table 8.4. Percentage of each end-result in each of the considered domains.**

One interesting question is whether ties are usually between adjacent classifications. The notion of adjacent classifications was given in Chapter 7, by which two class values are considered adjacent if they are related to each others. To clarify this issue, information was gathered with respect to the ties scored when PISA was applied with the Housing Benefits (4 Classes) dataset. As here the adjacency relations are pre-defined unlike the real-world datasets, which may require expert consultation to identify these relations. Table 8.5 illustrates the percentage of ties between adjacent classes in this dataset (the number between brackets indicates the percentage of the considered ties in the overall recorded ties of the given type). NE was omitted because it was assumed to be "*distant*" from the other classifications. PE was assumed adjacent to EN, and PR and EN were considered "*mutually*" adjacent. The reported results suggest that ties often take place between participants advocating adjacent classifications (87.82% of strong ties and 92.31% of weak ties took place between the three participants promoting EN, PE and PR in the given dataset).

| Ties (%) | EN and PE. | EN and PR, PR and EN. | Others |
|---|---|---|---|
| **Strong** | 0.45 (39.13%) | 0.56 (48.69%) | 0.14 (12.17%) |
| **Weak** | 0.24 (36.92%) | 0.36 (55.39%) | 0.05 (7.69%) |

**Table 8.5. Percentage of strong/weak ties between adjacent classes.**

Sub-section 6.3.1 discussed some mechanisms to resolve ties which involve reapplying PISA using only the undefeated parties. For weak ties an additional restriction was applied, aiming at forcing at least some of the involved parties to advance new arguments. Both mechanisms were applied to cases which resulted in ties after the first application of PISA in each of the domains included in the TCV tests described above. Table 8.6 illustrate the results of this second application. Note that the operation of PISA can benefit from applying the advocated mechanisms, in every case resolving the ties improved the accuracy. However, this comes with the cost of reapplying PISA to the cases in question.

| **Accuracy(%)** | **Before Resolution** | **After** | **% remained unclassified** |
|---|---|---|---|
| **Anneal** | 87.31 | 89.16 | 0.51 |
| **Cars** | 93.84 | 95.96 | 0.00 |
| **Flare** | 92.55 | 94.43 | 1.04 |
| **led7** | 90.16 | 93.04 | 2.3 |
| **Nursery** | 95.45 | 96.35 | 0.93 |
| **Page Blocks** | 97.33 | 97.52 | 0.20 |
| **Pen Digits** | 95.90 | 97.08 | 0.56 |
| **Waveform** | 97.84 | 99.04 | 0.00 |
| **Housing Benefits** | 95.60 | 97.00 | 0.40 |
| **Average** | **94.00** | **96.17** | |

**Table 8.6. Accuracy before/after tie resolution.**

### 8.2.3. Discussion about the Relation between the Participants' Experience and the Operation of PISA

The intuition behind PISA was to facilitate multiparty "*Arguing from Experience*" in which the amount of experience available to each party plays an important role in strengthening/weakening their arguments, and thus their contribution/position in the underlying debates. The empirical results reported thus far have shown that PISA works better with at least moderately large datasets, which enables the allocation of fairly decent amount of experience to each participant. These results merit further investigation. In order to isolate the size factor, an experiment was performed using a number of Housing Benefits (4 classes) datasets, generated so allowing any size of dataset required. Five datasets were generated for the purposes of this experiment, covering a wide size range; from as little as 100 records per participant (400 records in total) to

10000 records per participant (40000 records in total). TCV tests, using PISA, were then applies to these datasets, and the results were plotted in relation with the size of each set as illustrated in Figure 8.4.

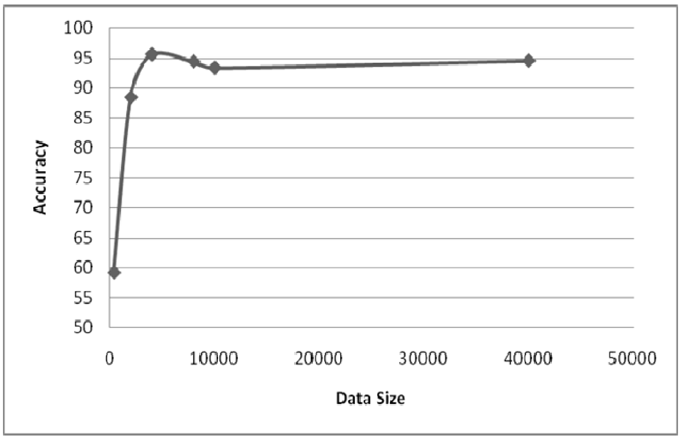


**Figure 8.4. The relation between data size and accuracy.**

As expected, when using very small datasets (100 records for each player) the accuracy of PISA is significantly lower than when using bigger dataset. This is because each participant was allocated a very small set of past examples which made it harder to draw valid arguments from. However, although the accuracy of the underlying dialogues initially increases when the size of the dataset increases, this is not the case for very large datasets. The given results suggest that the highest accuracy is obtainable when the size of the data available to each participant is in the range of 1000 to 2000 records. If the size of the data is much smaller or much large than this, the accuracy of the resulting dialogues may suffer.

## 8.3. Assessment of PISA as a Classifier

This section provides empirical evidence of the possibility of applying PISA as a classifier. The reported experiments compared the results obtained from applying PISA in the manner discussed in the previous section to those obtained from the identified classifiers, to assess the application of PISA as classifier. Four experiments were undertaken in order to investigate this application:

- The first compared the accuracies obtained from PISA to those obtained from the identified classifiers.
- The second experiment examined the behaviour differences between PISA and each of the applied classifiers using the McNemar's test.
- The third experiment evaluated PISA as classifiers ensemble.
- The fourth experiment was executed to assess the robustness of PISA towards random class noise in comparison with other classifiers.

The first experiment compared the results from the TCV tests reported in Figure 8.1 with the other classifiers. Figure 8.5 shows the average accuracy obtained from each classifier, for each dataset. Note that PISA performs consistently well: outperforming the other classifiers in four domains and giving comparable results in the others. These empirical observations merit further discussion. RDT outperformed the other classifiers in the Car Evaluations, Nursery and Pen Digits domains, mainly because of the nature of these domains. In particular, the Nursery datasets was originally produced using decision trees, and so it should be no surprise that RDT has surpassed the other classifiers in this domain. Also, the covering methods (FOIL, PRM and CPAR) worked well with the Annealing dataset as here the class distribution favours one class value with 76.17% of the cases, while the other class values are marginal. Thus, a cover algorithm that derives the most obvious class correlated to a rule, and ignores the other classes, will perform well when the class distribution is significantly biased toward one specific class. Note also that if the number of classes in such datasets is fairly large, then each participant will be given a small share of the original dataset. For instance the 898 records in the Annealing dataset provided only 150 records per participant, and here the performance of PISA was the worst when compared with other datasets. Another interesting observation is that PISA seems to be consistent with respect to the number of attributes in each record.

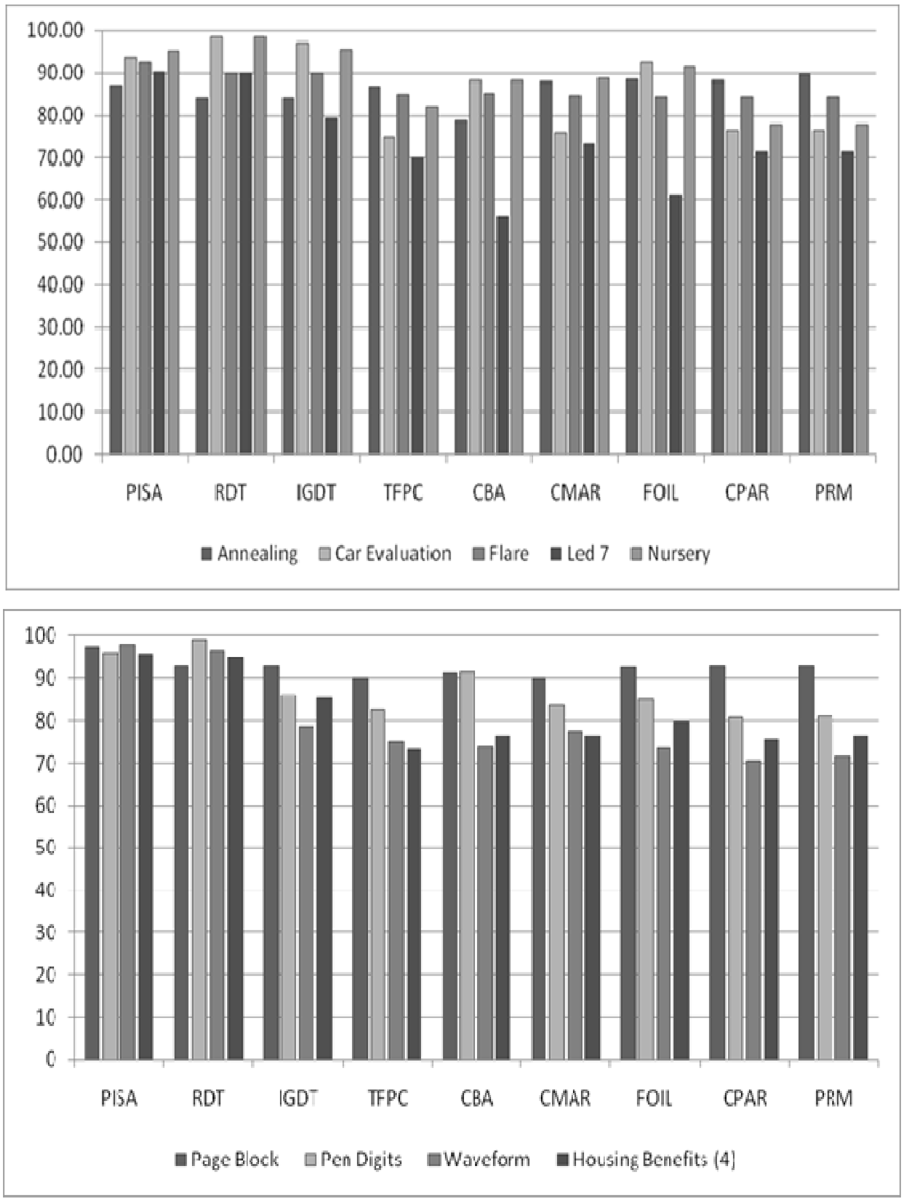


**Figure 8.5. Accuracy of the TCV tests for each dataset.**

The average accuracy across all the domains included in the previous TCV tests was also calculated. Figure 8.6 illustrates these results. Note that the average accuracy of PISA (94%) is better than the other classifiers (e.g. RDT = 93.82% and IGDT = 85.84%). This is due to the consistency of the performance of PISA when compared with the other classifiers. These results and the average accuracy obtained in each single domain, demonstrate that PISA provides a useful classification mechanism.

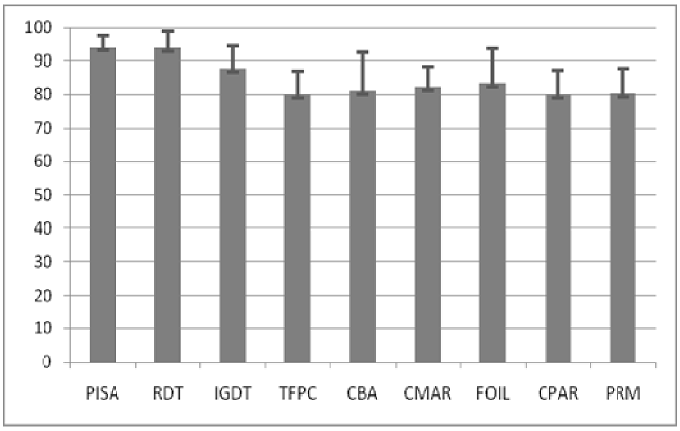


**Figure 8.6. Average accuracy across all multi-class domains.** *Error bars represent the standard deviation for each classifier.*

### 8.3.1. Analysis of the McNemar's Test

The McNemar's test was applied for each of the nine datasets included in the previous test, to explore the hypothesis that PISA is significantly better than the other eight classifiers, and to examine the differences in behaviour between PISA and each of the other classifiers in turn. The setup of this test was similar to the one outlined in Sub-section 5.2.1. However, here the datasets were equally divided according to the number of possible classes in each dataset. Table 8.7 shows the P-value associated with the executed McNemar's tests[39]. In general, PISA seems to operate better than other CARM classifiers, and as well as the decision tree methods, in most of the included domains.

As part of the McNemar's tests, detailed information as to which cases were misclassified by one or both of the two classifiers under consideration was also generated. Table 8.8 presents the detailed results obtained from the Housing Benefits dataset. Figure 8.7 illustrates these results comparing the operation of PISA with each of the other eight classifiers, for the real-world datasets. The results obtained from the performed McNemar's tests have revealed that both PISA and RDT have produced better accuracies than the other classifiers in

[39] Recall from Chapter 5 that the P-value indicates the probability of PISA producing results at least as good as the ones obtained from the other classifier, assuming that the null hypothesis is true. The *lower* the p-value, the *more* "*significant*" the differences are between the two approaches.

most of the cases; and that there are not any significance differences between the performance of PISA and the performance of RDT. Additionally, the mistakes made by both PISA and RDT are different (e.g. Table 8.8). Therefore, a joint application of PISA and RDT can potentially increases the overall accuracy of the classification process. The following sub-section discusses this point in some detail.

| **Dataset** | **CBA** | **CMAR** | **TFPC** | **RDT** | **IGDT** | **FOIL** | **CPAR** | **PRM** |
|---|---|---|---|---|---|---|---|---|
| **Annealing** | 0.0291 | 0.6767 | 1 | 0.013 | 0.013 | 0.8312 | 0.8312 | 0.8312 |
| **Cars Evaluation** | 0.4795 | 0.0412 | **<0.0001** | 0.1336 | 0.1336 | 0.4795 | 0.7728 | 0.7728 |
| **Flare** | 0.0953 | 0.0953 | 0.0953 | 1 | 1 | 0.0953 | 0.0953 | 0.0953 |
| **Led 7** | **<0.0001** | **0.0101** | **0.0039** | 0.6625 | 0.5224 | **<0.0001** | **<0.0001** | **<0.0001** |
| **Nursery** | **<0.0001** | 0.7728 | 0.0801 | 0.2207 | 0.3711 | 1 | **<0.0001** | **<0.0001** |
| **Page Blocks** | 0.0162 | 0.0162 | 0.0162 | 0.1138 | 0.1138 | 0.0162 | 0.0162 | 0.0162 |
| **Pen Digits** | 0.2278 | 0.3017 | **0.0044** | 0.1824 | 0.3865 | 0.0159 | **0.0098** | 0.0162 |
| **Waveform** | **0.0008** | **0.0002** | **0.0001** | 1 | **0.0007** | **<0.0001** | **<0.0001** | **<0.0001** |
| **Housing Benefits (4 Classes)** | 0.791 | 0.4227 | **0.0094** | 0.7237 | **<0.0001** | **1** | **0.0001** | **0.0008** |

**Table 8.7. The P-value associated with McNemar's Tests.** *Values in bold indicates extreme statistical differences between the two classifiers.*

| **Housing Benefits** | **CBA** | **CMAR** | **TFPC** | **RDT** | **IGDT** | **FOIL** | **CPAR** | **PRM** |
|---|---|---|---|---|---|---|---|---|
| **Both Failed** | 0 | 0 | 4 | 0 | 4 | 2 | 3 | 3 |
| **PISA Failed** | 5 | 5 | 1 | 5 | 1 | 3 | 2 | 2 |
| **Other Failed** | 8 | 9 | 11 | 3 | 39 | 4 | 22 | 18 |
| **Both Succeeded** | 87 | 86 | 84 | 92 | 56 | 91 | 73 | 77 |

**Table 8.8. Detailed McNemar's results for Housing Benefit (4 classes) dataset.**

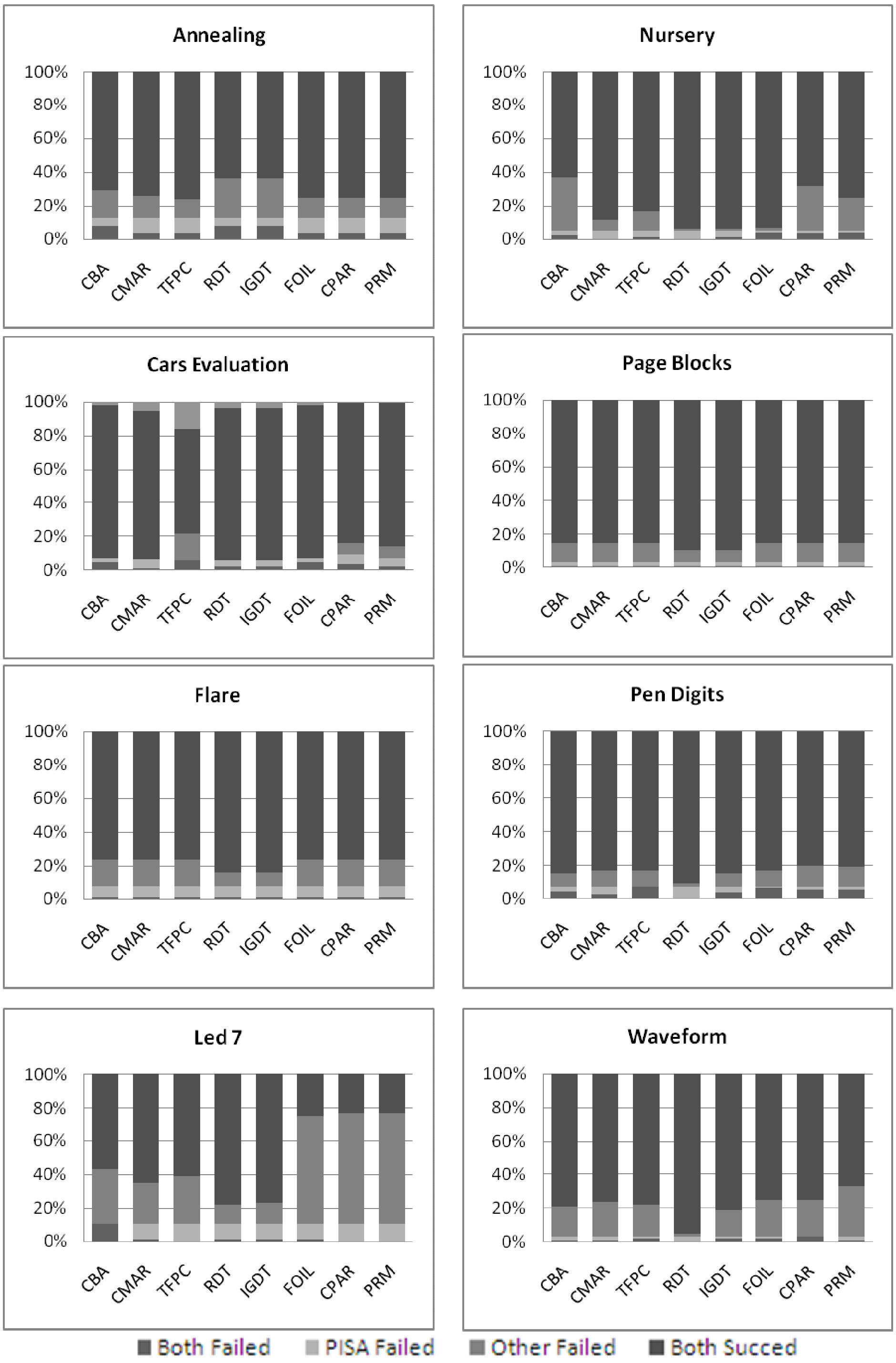


**Figure 8.7. Comparing PISA with the classifiers included in the McNemar's Test[40].**

[40] The results from CPAR and PRM were identical therefore only comparison with CPAR is illustrated in this table.

### 8.3.2. A Note about Combining the PISA with a Decision Tree Classifier

The analysis of both PADUA and PISA using the McNemar's test suggests that the process of "*Arguing from Experience*" can profitably used in conjunction with a decision tree method to produce better classifier. The joint approach can, potentially, succeed in predicting the class label for the cases which both PISA and RDT have failed to classify. One possible approach to such joint application is to run RDT first, and then apply PISA to the cases which RDT failed to classify correctly. Mozina et al. (2005) apply a similar approach by which they first run CN2 (Clark and Niblett, 1989) to generate a set of rules corresponding to the given dataset. After inducing these rules, the most problematic examples are identified by the means of a TCV test. An expert then gives arguments as to why each example should be associated with certain class. The method of (Mozina et al., 2005), ABCN2, is then applied to the "*argumented*" examples. However, combining PISA with RDT offers two advantages:

- The advocated approach does not involve expert consultation; once the hard cases are identified using RDT they can be piped directly to PISA.
- PISA and RDT have proved to work well with multi-class as well as two-class problems, unlike ABCN2 which works with two-class datasets only.

Also, the combination of RDT and PISA was motivated by the differences in their behaviours: when RDT was applied to the Housing Benefit (4 Classes) dataset it correctly classified most of the 100 test cases. However, three "*hard*" cases remained misclassified. These cases were identified and PISA was applied to each case. Interestingly, because of the differences in PISA and RDT classification behaviours, PISA correctly classified the three cases. For example in order to classify the first identified case of a 84 year old, UK resident, female applicant, who satisfies all conditions for entitlement to Housing Benefits, but not to priority entitlement, as she has not paid contributions in the last year nor five years ago (year5). PISA generated the following dialogue:

First Round: EN – Proposes a new Rule: The case has the following features: Age>80, 15%<Income<20% and 2000£<Capital<3000£. Therefore the case should be classified as: (entitled). With confidence = 58.33%.

Round 2:

PE – Distinguishes a Rule against a previously played move: The case has the following additional feature: Contribution Y5 = not paid. Therefore my confidence in this case being of class (entitled) is no more than 20.0% only.

PR – Counter Rule: The case has the following features: Gender = female, 15%<Income<20%, Contribution Y1 = not paid, Contribution Y2= paid, Contribution Y3 = paid and Contribution Y4 = paid. Therefore the case should be classified as: (priority entitled). With confidence = 60.67%.

NE – Distinguishes a Rule against a previously played move: The case has the following additional features: Gender=female and Contribution Y5 = not paid. Therefore my confidence in this case being of class: (entitled) is no more than 32.9% only.

Round 3:

EN – Proposes a new Rule against a move previously played by PR. The case has the following features: Residency = UK, 15%<Income<20% and 20003<Capital<3000£. Therefore the case should be classified as: (entitled). With confidence = 78.78%.

Round 4:

PR – Distinguishes a Rule against previously played move: The case has the following additional feature: Gender = female. Therefore my confidence in this case being of class: (entitled) is no more than 15.38% only. If it has the additional attribute: contribution year4 = paid.

NE – Distinguishes a Rule against previously played move: The case has the following feature: Gender= female, Contribution Y1 = not paid and Contribution Y5 = not paid. Therefore my confidence in this case being of class: (entitled) is no more than 22.78% only.

Round 5:

EN – Proposes a new Rule: The case has the following features: Residency = UK, 15%<Income< 20%, 2000£<Capital<3000£, Contribution Y2 = paid and Contribution Y4 = paid. Therefore the case should be classified as: (entitled). With confidence =

### 8.3.3. A Comparative Study of PISA as an Ensemble Method

Recall from Chapter 2 that the objective of ensemble methods is to build "*ensembles"* of "*weak*" classifiers which, when used in combination, produce a single strong classifier. Two ensemble methods were also discussed: (i) *Bagging* whereby a number of classifiers are generated from the sample taken from the input data, and (ii) *Boosting* whereby a series of weak classifiers are iteratively generated and compound into a single strong classifier. Following these definitions, PISA is argued to embody a bagging-like method, by which the dataset is equally divided amongst a number of participants corresponding to the number of class values in the dataset. Each participant applies the same algorithm to mine CARs supporting their advocated classifications. To this end, each participant corresponds to a single classifier. The argumentation process by which each participant advances moves to support its proposals corresponds to voting methods by which ensemble techniques assigns class labels to input cases. However, the argumentation process of PISA differs from voting in Bagging. While Bagging returns the class label that won the most votes, and all votes are equal, PISA applies a debate, whereby each participants aims at persuading the other participants of a particular classification of the given case. PISA also differs from Boosting techniques in that it does not generate series of classifiers; rather the classification decision is achieved via the argumentation process amongst the participants. Furthermore, PISA classifies unseen records "*on the fly*" by producing a limited number of CARs sufficient to reach a decision without the need to produce the full set of CARs. However, the resemblances between PISA and ensemble methods, in that both approaches divide the data amongst a number of classifiers, and apply "*Meta*" techniques to assign class labels to unseen records, merit further investigation.

The following presents the results of a number of TCV tests aimed at comparing the operation of PISA to that of well known ensemble methods. Three Ensemble techniques, ADABoost, MultiBoosting and Bagging were applied to a collection of 14 datasets. WEKA 3 (Hall et al., 2009) was used to carry out these ensemble experiments. ADABOOST/ADABoost.m1 and Multiboosting TCVs were executed using 10 irritations eight mass to build the default 100 classifiers.

Bagging was executed using the default number of irritations (10). The size of each bag was a 100 cases (default).

Figure 8.8 illustrates the average accuracy achieved using each of the ensemble techniques and compares it to the results obtained from PISA. It is evident that PISA produced results comparable to those produced by ADABoost and Bagging, and outperformed MultiBoosting in all the identified domains. Moreover, PISA scored an average accuracy (95.53%) higher than that obtained from any of the methods studied (e.g. bagging (90.70%)). Note that ensemble methods seem to perform worse than PISA in domains that contain a large number of classes. However, ensemble methods seem to outperform PISA in two-class domains. PISA, however, has shown consistent performance with both multi-class and two-class datasets.

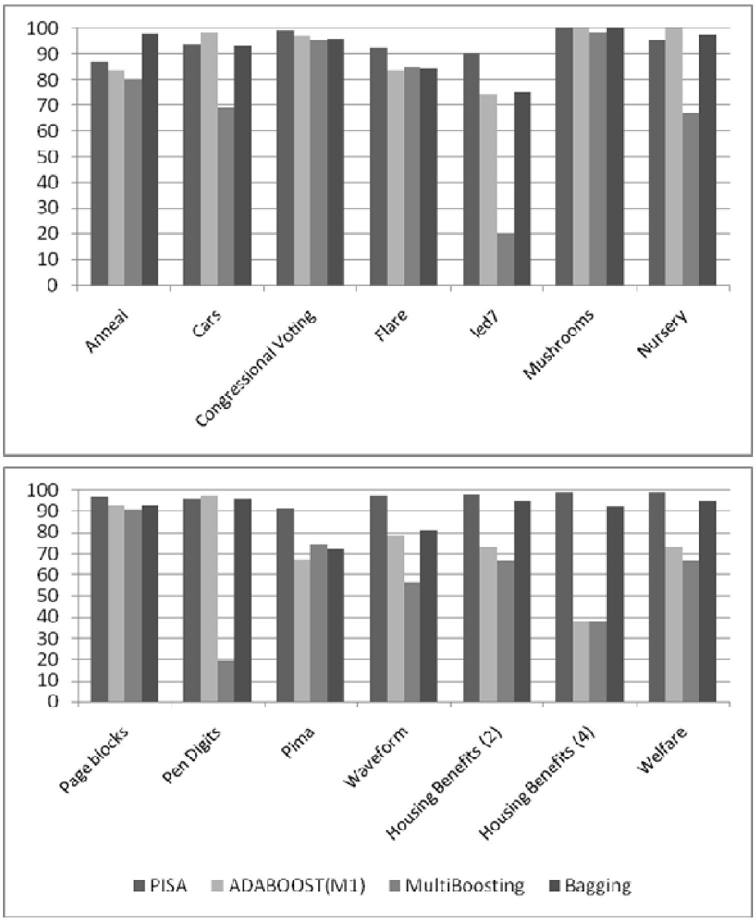


**Figure 8.8. An evaluation of PISA and ensemble methods using TCV.**

### 8.3.4. Classifying Noisy Data

One of the challenges of the classification problem is how to deal with (very) noisy data as most real world data is often infected with varying levels of noise. Sub-section 2.3.5 discussed a number of different noise types and their treatment in the literature on classification in KDD. An alternative approach is to apply "*Arguing from Experience*" to decrease the effect of noise on the overall accuracy of classification (e.g. Sections 5.4, 5.5). Handling noisy data is not only important to the utilisation of the proposed model to solve classification problems. It also touches upon a very important issue with respect to the process of arguing from past experience. If the promoted model is to be applied in real-life settings, it would be often the case that the agents' gathered experience contains some sort of noise, which may not be possible to clean. Therefore, the advocated model had to show some tolerance to noise, should we wish to make full use of it to support real-life dialogues from experience amongst a number of independent autonomous agents. An assessment of the effectiveness and robustness of PISA with respect to noise using a wide range of datasets is given below. A number of TCV tests were applied using each dataset. Random noise was introduced to the class label of the training set of each test, and not to the test set, using the following model[41]: for an N% noise level in a dataset of I instance ((N/100)*I) instances were randomly selected and the class label changed to some other randomly selected value (with equal probability) from the set of available classes. The noise levels used in this study were: 2%, 5%, 10%, 20%, 40% and 50%. The operation of PISA was also compared against the eight identified classifiers, in addition to a number of ensemble methods (ADABoost, Bagging and BrownBoost)[42]. BrownBoost was carried out using the Jboost package. 100 runs were excusted, and margin logs were generated for each iteration. The amount of error to be tolerated in the training

[41] This model is different from the one used in Sub-section 5.4.1 to test PADUA's robustness to noise which was motivated to provide means to compare the operation of PADUA to that of CN2 and ABCN2 as reported by Mozina et al (2005). For the purposes of testing PISA it was believed that CTV tests provide better assessment.

[42] Note that different ensemble techniques were used in this experiment rather than the ones used for healthy datasets in Section 8.2. This is because BrownBoost has shown some robustness against noise, while MultiBoosting often produces results very similar to those generated by ADABoost.

set was set according to the percentage of noise in this set (e.g. if the training set had 10% mislabelled examples then the booster was told to accept a 10% error rate).

First, PISA was evaluated using a Housing Benefit (4 Classes) from Section 8.2. The given levels of noise were applied to this dataset, in the manner described above. The training dataset used for each of the noise levels, was then split into four equal subsets; each subset was given to one player, and the four players argued to classify the 240 cases in the test set, for each of the ten runs of the TCV. Figure 8.9 shows the effects of adding noise to the Housing Benefit (4 classes) dataset on the performance of PISA and the identified approaches.

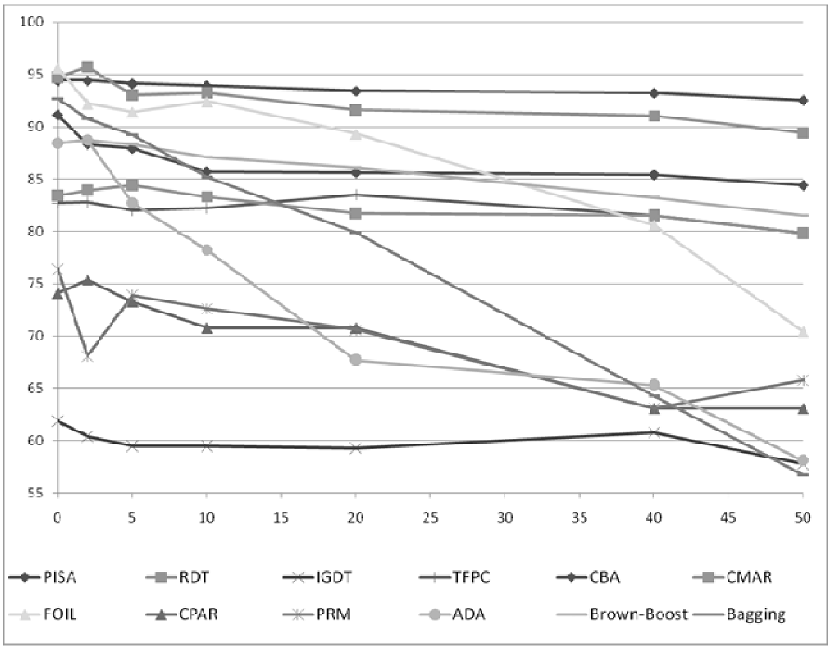


**Figure 8.9. Accuracy versus noise (Housing Benefits - PISA).**

From the above figure it is evident that the overall accuracy drops in relation to the increasing noise. The best overall classifier is PISA, with an accuracy level starting with 94.4% for clean data and dropping to 92.5% with 50% noise. This indicated that the PISA coped extremely well when noise was introduced to the data given to each participant. Also, it appears that Brown Boosting is indeed pretty good for noisy datasets. It starts from a lower accuracy but holds up quite well, when compared to the other included classifiers. The reported results suggest that the proposed model for "*Arguing from Experience*" is tolerant to noise in the "*experience*" gathered by the individuals taking part in the underlying debates. Consequently, these results suggest that the utilisation of PISA to classification shows better robustness to noise than the other classifiers (e.g. RDT=89.4% with 50% noise and BrownBoost = 81.5%). This latter point suggests that PISA can be made use of to correctly predict class labels even when using noisy or bad data.

The reported experiment demonstrated the utility of PISA when using noisy experience. However, in order to provide a comprehensive analysis of its robustness toward noise, PISA was also evaluated using a more variable collection of datasets. The operation of PISA was then assessed and compared to that of the classifiers mentioned above. Below only the comparison with decision trees and ensemble methods is reported because these methods were found to be the closest "*competitors*" to PISA. The results of the evaluation showed a similar pattern: the accuracy of almost all the test sets dropped when the noise percentage was increased. However, PISA maintained a good level of accuracy even with high noise, and did not display any severe drops in this accuracy in any of the included domains. Figure 8.10 illustrates these results (the percentage of noise is given on the X-axis and percentage accuracy on the Y-axis). From this figure it is evident that with the increase of noise levels, PISA (and RDT) starts outperforming all the other classifiers, when the noise level hits 50% the difference in performance between these two classifiers and the rest of the classifiers included in this study, becomes considerable. Moreover, with high levels of noise PISA produces the best results with most of the dataset. Note that the datasets where RDT outperformed PISA with high levels of noise tended to be those with a small number of records per class, so that each PISA player had only a limited number of cases from which to mine their arguments. Also, PISA produced consistent performance and shown the same levels of noise tolerance across the whole collection of datasets.

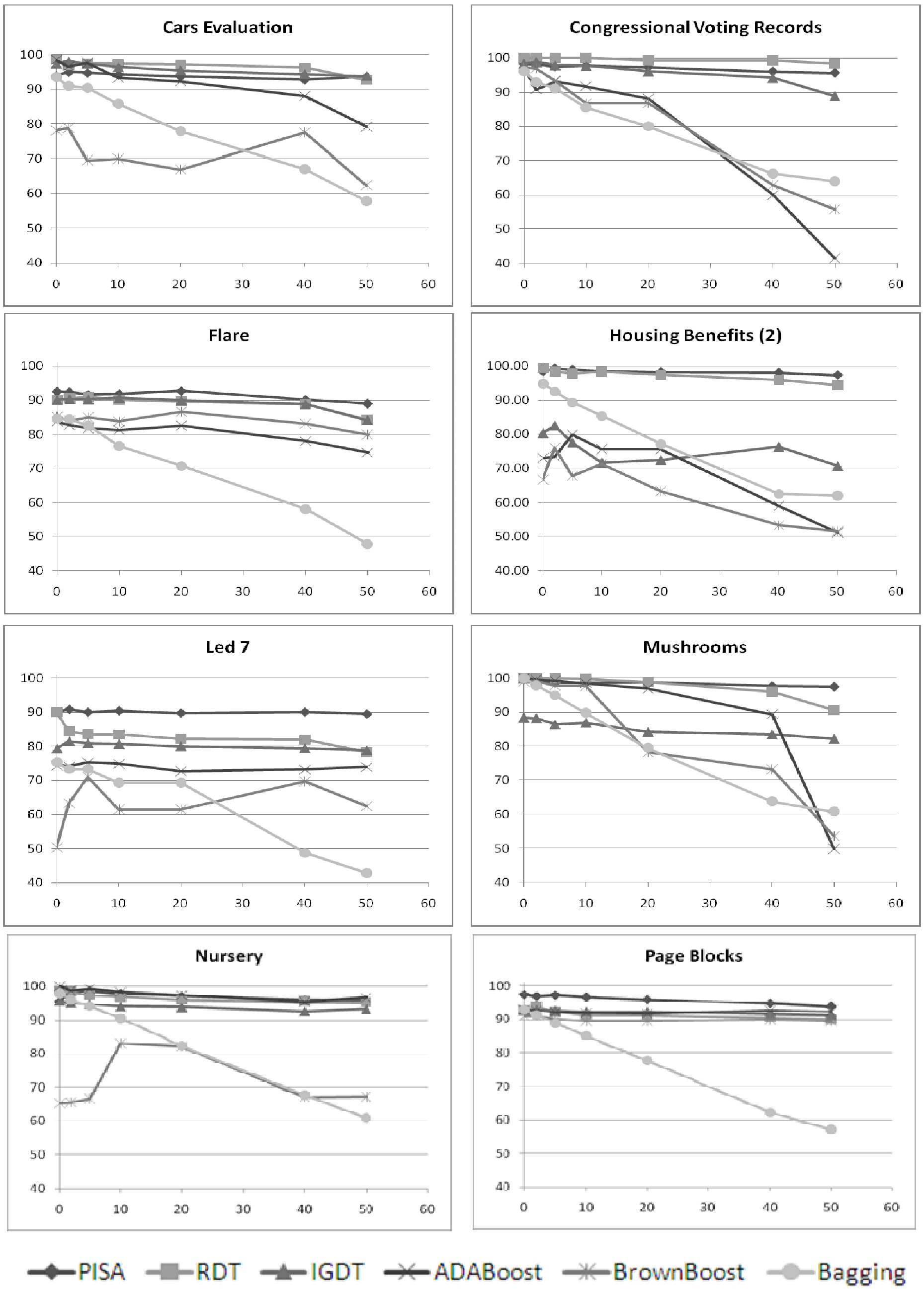


**Figure 8.10 (a). Accuracy versus noise (PISA).**

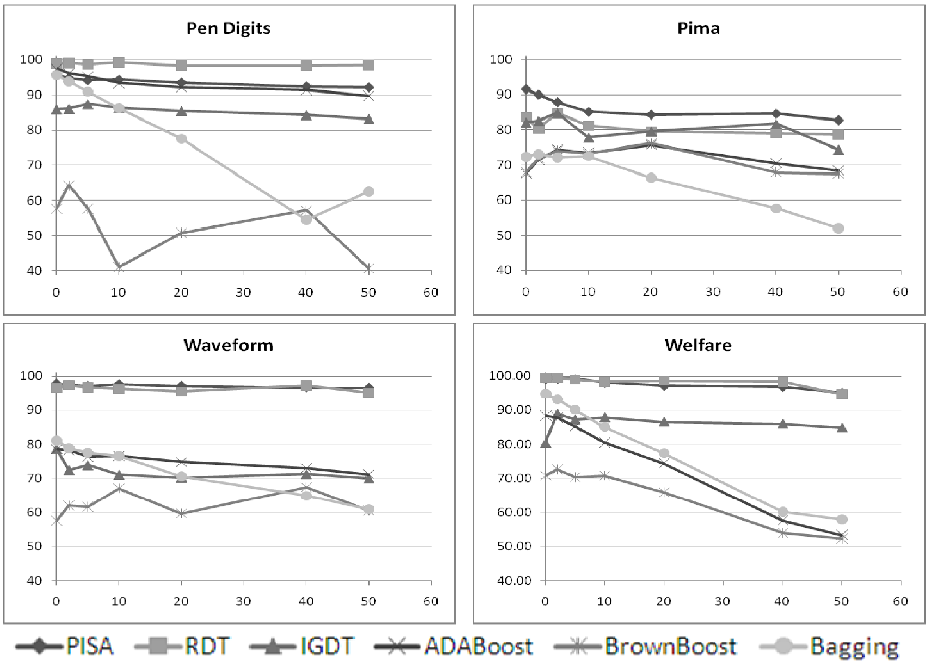


**Figure 8.10 (b). Accuracy versus noise (PISA) (continued).**

PISA was also applied using a different type of noise, by which a number of different *systematic errors* were introduced to a number of underlying Housing Benefits (4 Classes) datasets. The results of this assessment have shown that PISA performs well in the presence of these errors, outperforming all the other included classifiers. However, for reasons of space, the study of the effects of *systematic errors* on the operation of PISA is discussed in a Appendix D.

## 8.4. Investigating the Role of Groups in PISA

PISA allows for any number of agents to take part in "*Arguing from Experience*" dialogues. However, if two or more agents support the same classification, then they will "*join forces*" and form what was termed a "*group*" of players. The details of how groups are formed in PISA were discussed in the previous chapter. Two types of groups where distinguished according to the strategies of their members, and the decision making process within the group was also explained. This section provides an analysis of how the performance of PISA relates to the number of players in each group, and is concerned with homogenous groups only (where all the members apply the same strategy). Also, it is assumed that all the group members have the same amount of experience, so that the process of selecting a group leader is utterly random. The following section will return to the issue of groups and provide some discussion regarding heterogeneous groups. Two experiments were carried out to investigate the operation of groups in PISA:

- **Experiment with the same number of players per each group**: involved five TCV tests using PISA with four groups, each comprising the same number of players, using the Housing Benefit dataset from Section 8.1.
- **Experiment with variable number of players per each group**: comprised a set of TCV tests using the same dataset as above, but here each group was assigned a random number of players (2 to 8).

### 8.4.1. Same Number of Players Per Group

The first experiment provided evidence on the operation of groups within PISA. This experiment assumed that the amount of data available to each group was fixed and equally divided amongst its members, similar to an ensemble-like approach to classification. Consequently, if too many players joined one single group, each with a very small dataset, the group will not be able to defend its propositions, for its members will not be able to mine adequate rules. An alternative approach in which each player has accumulated its own experiences, independent from other members was investigated in the second experiment.

Five TCV tests were carried out using the same 8000 records Housing Benefits dataset from Section 8.1. PISA was run using four groups comprising (respectively) of 2, 4, 6, 8 and 10 players each. In each test the original dataset was divided equally amongst the players. Figure 8.11 presents the results of these tests, which indicate that, in general, PISA operates better when carried out using groups of players. As having more than one agent advocating the same classification, using the notion of groups, seems to have positive effects on the accuracy of the resulting dialogues, due to the fact that more arguments can now be mined from different datasets, each presenting different experience. Thus more options will be available to the group as a whole, from which the group's leader can select the best course of action. However, the increase in accuracy is proportional to the amount of data given to each agent in the dialogues. If the data is not sufficient to mine adequate ARs, the operation of PISA will not benefit from dividing the data any further. Note that the accuracy of the resulting dialogues starts to drop when the size of each dataset falls below 250 records. Another issue is the overhead cost resulting from the decision making process within each group. For each round of the dialogue, each member of the group attempts to suggest a move advancing the classification advocated by this group. Also, the group's leader has to choose one of these moves to place forward in the ongoing dialogue. The previous experiment provided information about the average number of moves suggested by the groups' member in each of the ten runs of each of the TCV tests. These results have shown that, on average, an overhead of 0.81*N (N is the number of players/group) is added to the overall lengths of dialogues when applying PISA using groups.

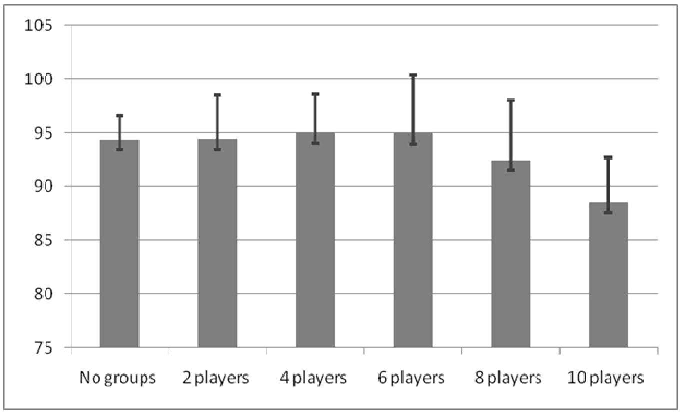


**Figure 8.11. Results of the TCV tests using same number of players per group.**

### 8.4.2. Variable Number of Players Per Group

The second experiment was conducted to investigate the role of groups in PISA expanded the above discussion by investigating two points: (i) the relation between the overall number of players and the accuracy of the subsequent dialogues; and (ii) the relation between the amount of data available to the group as a whole and the performance of the group. For these purposes, two sets of

TCV tests were performed, each attending to one of the above points. In the first one, the amount of data available to each group was equally divided among a random number of players, and then a TCV test was performed. The TCV test was repeated a number of times with the number of players in each group randomly generated. Figure 8.12 presents the results of these tests. As expected, the accuracy of the classifications obtained benefited from having more than one agent championing each of the possible classifications. However, once the amount of data available to each player dropped below a certain threshold, the overall accuracy started to fall. For instance, when each player was given 1000 records, the accuracy was 94.48% on average and when each player had only 200 records, the average accuracy fell to 88.58%.

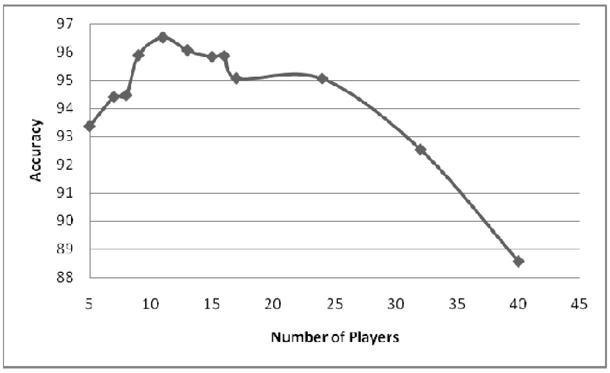


**Figure 8.12. The relation between the number of players and the accuracy of PISA.**

To address the second point, a number of Housing Benefits datasets, each containing 1000 records, were generated. PISA was run with four groups, each comprising the same number of players. But here, each player was given one of the generated datasets. Thus, the more players that join the same group, the more experience is available to the group as a whole. A number of TCV tests were then carried out. For each test, the number of players in each group was randomly generated. Figure 8.13 shows the result of these tests.

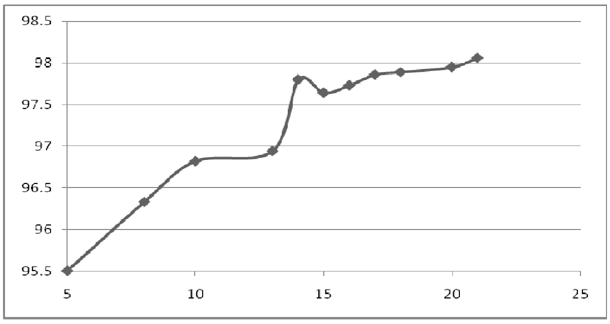


**Figure 8.13. The relation between the number of players and the accuracy of PISA when players are given fixed amount of data.**

Note that the overall accuracy benefits from the increase of the number of players in each group. For instance, the average accuracy when each group comprised five players was 95.5%. However as the number of players in each groups is increased, the accuracy gradually increase to reach 98.06% when each group contains 21 players. Note that the groups were assigned fairly large number of players. This was to establish that the drop in accuracy associated with the increase of the data divisions in the previous experiment was not related to the number of players in each group.

Another set of TCV tests was carried out. These tests focused on determining the two effects of group's size on accuracy: (i) a big group would win when it should lose, and a (ii) small group fails to win when it should win. For these purposes, the four groups in the previous test were assigned random number of players. Each player was given similar sized Housing Benefits dataset, like above. Four random players' allocations were generated, and the TCV test was repeated for each allocation (Table 8.9 (a)). The results of these tests suggest that the two effects highlighted above hold. In order to clarify this point, supplementary information was generated with respect to the winning groups for each dialogue in the four TCV tests. The results generated have shown that in TCV1, group G4 has failed in winning 30.16% of the cases that should classify according to its advocated class. Also, 94.25% of these cases were won by G1 (highest number of players). Similar results were reported with the other TCV tests. Table 8.9 (b) illustrate, for each group, the percentage of cases that this group has failed to classify correctly.

| Players | TCV1 | TCV2 | TCV3 | TCV4 |
|---|---|---|---|---|
| G1 | 8 | 5 | 1 | 3 |
| G2 | 4 | 8 | 7 | 8 |
| G3 | 4 | 1 | 4 | 2 |
| G4 | 1 | 3 | 5 | 3 |
| PISA Accuracy | 90.21 | 91.73% | 93.91% | 94.38% |

**Table 8.9 (a). Four random allocations for random number of players/group.**

| Test | G1 | G2 | G3 | G4 |
|---|---|---|---|---|
| G1 | 0.00% | 2.81 | 21.05% | 4.90% |
| G2 | 1.74% | 0.00% | 0.31% | 0.00% |
| G3 | 3.26% | 29.08% | 1.90% | 10.90% |
| G4 | 30.16 | 1.19% | 1.10% | 6.68% |

**Table 8.9 (b). Percentage of misclassified cases that should have been won by each group from Table 8.9(a).**

## 8.5. Investigating the Impact of Players' Strategies upon the Operation of PISA

Chapter 7 presented a two-tier strategy model to accommodate multiparty "*Arguing from Experience*" dialogues. Three basic strategies were derived from the proposed model: (i) attack whenever possible (S1), (ii) attack when needed (S2) and (iii) attack to prevent a forecasted threat (S3). Each interpreted the argumentation tree associated with the PISA Framework in a different manner. Examples were also given as to how applying different strategies lead to different dialogues. These examples assumed that the agents involved in a dialogue would increase their chances of winning the dialogue if they made the utmost use of the argumentation tree. In this section, the results of a number of experiments are analysed to investigate whether this claim stands or not. The experiments, reported below, compared the operation of PISA with a number of redefined possible strategy combinations. Table 8.10 gives the details of these strategies[43].

| **Strategy** | **Name** | **Strategy (S1, S2,S3)** | **Sub-Strategy** | **Mode** |
|---|---|---|---|---|
| **ES1 (best)** | S3 | S3 | - | - |
| **ES2 (Worst)** | S1-1-2 | S1 | Blind | Destroy |
| **ES3** | S2-3-2 | S2 | Tree Dependent - Full | - |
| **ES4** | S1-3-2 | S1 | Tree Dependent – Full | - |
| **ES5** | S2-3-1 | S | Tree Dependent - Leaves | - |
| **ES6** | S1-3-1 | S1 | Tree Dependent - Leaves | - |
| **ES7** | S2-2-1 | S2 | Focused | Build |
| **ES8** | S1-2-1 | S1 | Focused | Build |
| **ES9** | S1-1-1 | S1 | Blind | Build |
| **ES10** | S2-2-2 | S2 | Focused | Destroy |
| **ES11** | S2-1-2 | S2 | Blind | Destroy |
| **ES12** | S1-2-2 | S1 | Focused | Destroy |

**Table 8.10. Strategies used in evaluating PISA.**

A sequence of TCV tests were then carried out using different combinations of the predefined strategies. A total of 12 different strategies were predefined to cover a variety of situations including worst possible and best possible strategies. The rest of the strategies used were randomly generated. Table 8.11 illustrates these strategies. In order to establish the consequences of strategy on the resulting dialogues a total of three experiments were undertaken, each comprising a series of TCV tests. These tests were performed using the Housing Benefit dataset from Section 8.2 and were carried out as follows:

- **SE1**: Examined the role of strategies in dialogues involving individual players (no groups).
- **SE2**: In which PISA was applied using homogeneous groups of players each contained four individual players.
- **SE3**: Aimed at investigation of the operation of heterogeneous groups.

[43] In PISA, the number of all the possible strategy combinations could be very large. This is exacerbated when some (or all) of these participants are indeed groups of individual players. Therefore, PISA was experimented with using a number of predefined strategies, generated at random, as exemplified in Table 8.10.

| Strategy | P1 | P2 | P3 | P4 |
|---|---|---|---|---|
| **Random Allocation 1** | ES8 | ES12 | ES5 | ES10 |
| **Random Allocation 2** | ES5 | ES3 | ES6 | ES11 |
| **Random Allocation 3** | ES4 | ES7 | ES4 | ES7 |
| **Random Allocation 4** | ES10 | ES5 | ES9 | ES8 |

**Table 8.11. Four random strategy allocations.**

### 8.5.1. SE1 – No Groups

SE1 examined the role of strategies in dialogues involving individual players. In these tests PISA was applied using four individual agents. Then different strategies were assigned to these players as follows:

- **Test 1-1**: one player was given the best possible strategy (ES1) while the rest were given similar strategies (S1-2-1).
- **Test 1-2**: all players were given the best possible strategy (ES1).
- **Test 1-3**: all players were given the worst possible strategy (ES2).
- **Test 1-4**: the four players were assigned strategies randomly as illustrated in Table 8.11.

Figure 8.14 illustrates the accuracy obtained from applying PISA using the identified strategies. These results suggest that the highest accuracy (95.05%) is obtained when using the Random Allocation 3, by which two players applied S1-3-2 and the other two players used S2-2-1. The worst possible allocation (T1-3) results in an accuracy of 87.26%. However, the worst accuracy (86.21%) was obtained from Random Allocation 2. By this allocation, three of the players apply S2 sub-strategies, two of which are tree dependent, one with full tree inference and one with leaves-only inference. The last player applies S1-3-1.

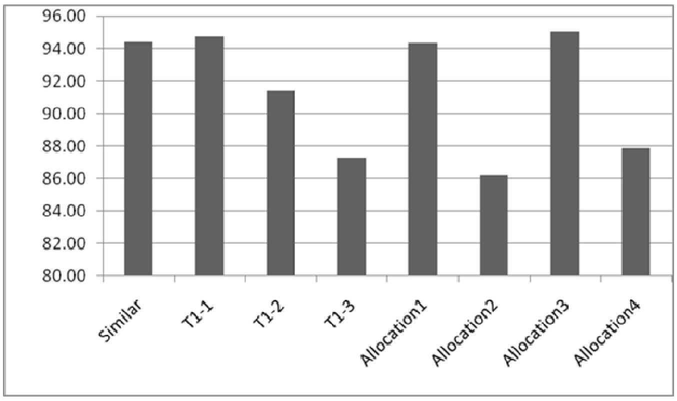


**Figure 8.14. Results of first PISA strategy test.**

Detailed information about the winning parties, for each of the above dialogues, was also generated. The obtained results suggest that equipping the players with better strategies, may enable them to win, even when they are advocating the wrong classification. For instance, in the case of Random Allocation 2, the player with strategy ES3 (the best strategy in this allocation) (Player2) has won the dialogue games over cases which should classify according to the class value it promotes. Additionally, the player with strategy SE11 (the worst in this allocation): Player 4 has failed to win 20.5% of the cases that should classify according to the class value it advocates (13.25% of the total misclassified cases). In most of these cases the winner was Player2. The rest of the misclassified cases, ended with a tie between the players with the right classifications and Player2. These results suggest that, when players are equipped with different strategies, those associated with the best strategies often win PISA dialogue games.

Additional information regarding the dialogues was also gathered. Interestingly, the shortest dialogues were observed when all players applied the best possible strategies (4.79 rounds ± 1.95 standard deviation) and the longest were observed when all players were given the worst possible strategy (7.10 ± 3.88 standard deviation). This is because the worst possible strategy enables the participants to prolong the game by attacking whenever possible and aiming at undermining their opponents' moves rather than building their own proposals. In contrast, with the best possible strategy, the players attack only when necessary, using the best possible attacking moves. As expected, when all or most players make use of build strategies the percentage of green wins/strong ties was high and vice versa, when all or most players applied destroy strategies the number of blue wins/weak ties was high. For instance, the highest percentage of blue wins/ties (53.5% and 8.75% respectively) was scored in T1-3 when all players applied the worst possible strategy.

The McNemar's test was also applied to examine if the behaviour of PISA changes when the strategy setup of its players was altered. The test compared the results from applying PISA with four players (no groups) applying a similar strategy, to the results obtained from each of the allocations described in Table 8.11. The results of this test revealed that the performance of PISA was significantly worse when applying Random Allocation 2 and when the four players make use of the worst possible strategy, than when all the four players apply the same strategy. Figure 8.15 illustrates the results of this test.

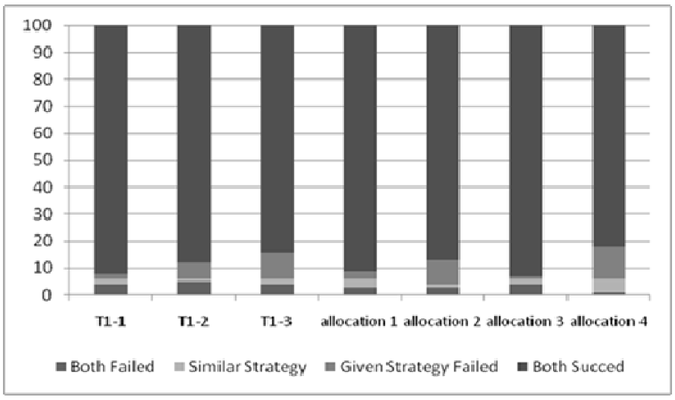


**Figure 8.15. McNemar's results for PISA strategy test.**

### 8.5.2. SE2–Homogenous Groups

Here, PISA was applied using four groups of players made of four individual players. It was also assumed that the members of each group made use of the same strategy. These tests were numbered T2-1 to T2-4 and they correspond to T1-1 to T1-4 but with groups instead of individual players. The results of these tests were then compared against applying PISA using four groups of players all applying the same strategy. Figure 8.16 presents these results. Here, Dark grey columns represent results using homogenous groups and light grey ones represent the results of the previous test. As expected, the results suggest that the average accuracy has risen for all the tests, and that the same allocations that produced best/worst results in the previous set of tests, have scored similar results when used with groups.

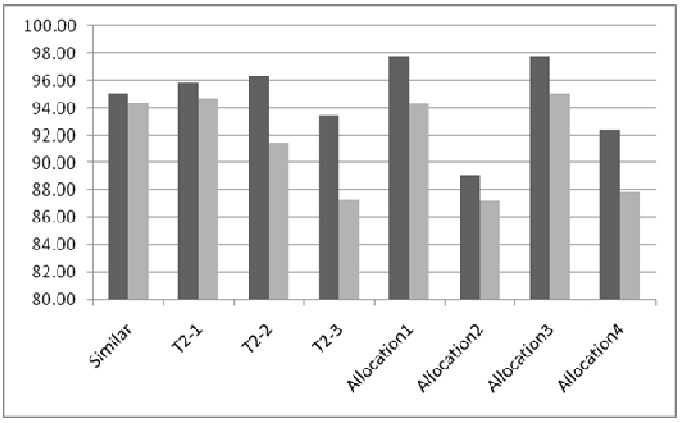


**Figure 8.16. Results of the second strategy experiment.**

### 8.5.3. SE3 – Heterogeneous Groups

Here, a set of TCV tests was designed to investigate what happens when group members apply different strategies. These tests made use of the same groups as in the previous experiment. However, here the focus was on heterogeneous groups rather than homogenous ones, whereby each member of the group was given a different strategy. Table 8.12 illustrates the new strategy allocations.

| | Group1 | | | | Group2 | | | | Group3 | | | | Group4 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **P1** | P2 | P3 | P4 | **P1** | P2 | P3 | P4 | **P1** | P2 | P3 | P4 | **P1** | P2 | P3 | P4 |
| R1 | S1 | S9 | S12 | S7 | S1 | S5 | S11 | S3 | S1 | S9 | S3 | S7 | S1 | S3 | S3 | S6 |
| R2 | S3 | S2 | S4 | S6 | S1 | S2 | S10 | S5 | S5 | S2 | S6 | S7 | S1 | S2 | S3 | S12 |
| R3 | S5 | S8 | S12 | S12 | S5 | S12 | S7 | S9 | S1 | S1 | S5 | S5 | S3 | S5 | S12 | S6 |
| R4 | S6 | S7 | S10 | S10 | S1 | S5 | S6 | S12 | S3 | S9 | S10 | S9 | S1 | S3 | S4 | S8 |
| R5 | S1 | S4 | S7 | S5 | S1 | S5 | S8 | S4 | S3 | S4 | S11 | S12 | S1 | S5 | S4 | S4 |
| R6 | S5 | S7 | S9 | S9 | S1 | S5 | S5 | S6 | S3 | S9 | S10 | S7 | S1 | S3 | S7 | S8 |

**Table 8.12. Heterogeneous groups random strategy allocations.** ***P1 refers to the group leader.***

The results are plotted in Figure 8.17 which shows that different results were obtained when players in each group had been assigned dissimilar strategies. These results indicated the importance of strategy in PISA, by applying different strategies one can either increase or decrease the accuracy, as well as altering the features of the resulting dialogues.

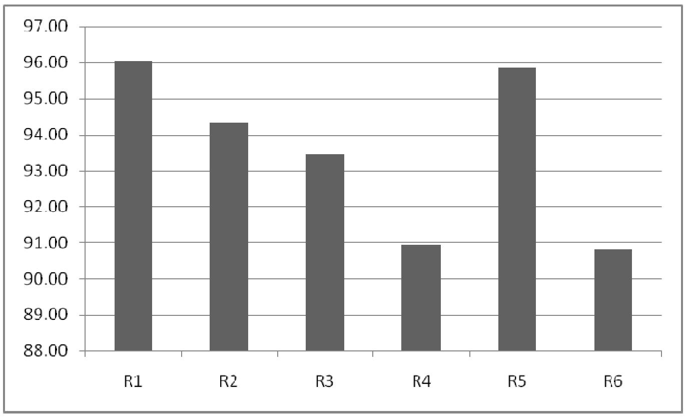


**Figure 8.17. Results of third strategy test using PISA.**

## 8.6. Summary

Implementing PISA, according to the discussion given in the previous two chapters, has proved to be valuable in evaluating the underlying model for multiparty "*Arguing from Experience*". This application provides a proof of concept by showing that reliable dialogues can be conducted using PISA. The analysis given in this chapter has revealed that the proposed model for multiparty "*Arguing from Experience*", as embodied in PISA, usually produces coherent dialogues, leading in most of the cases to a successful resolution of the underlying debates. The approach of providing evidence to the nature of these dialogues, focused on applying PISA to a number of different classification problems. A discussion of the results obtained from a collection of empirical experiments intended to assess the "*effectiveness*" and nature of the resulting dialogue was given. PISA was shown to provide a productive route to multiparty "*Arguing from Experience*" through dialogues capable of addressing debates regarding the classification of cases from variety of domains.

The operation of PISA as a classifier was also investigated. The given analysis suggested that PISA can be profitably exploited as a valuable means of classification. The advantageous features of PISA, identified in the foregoing, may be itemized as follows:

- It does not require a training phase. Furthermore, PISA produces a limited number of classification association rules sufficient to classify the given case without the need to generate all the possible rules.
- PISA may be applicable to large datasets, as well as small ones. However, it is advised to use PISA with moderately large sets so that the individual players have a reasonably sized "*experience (case) base*" to argue from. Very large datasets should be split and groups used.
- By dividing one dataset amongst reasonable number of agents, distributed amongst a number of groups, PISA can act as an ensemble technique that competes with other well-known ensemble methods.
- PISA is noise tolerant, in that it can cope well with high levels of class noise in the training set without failing to classify correctly the test cases.
- It has a wide range of configuration parameters, such as the number of players in each group and the strategy setup for each player; by changing some of these parameters the course of PISA dialogues can be modified to better fit the underlying datasets.

This concludes the analysis of PISA, and of the process of multiparty "*Arguing from Experience*". However, it can be argued that the particular PISA application examined in this chapter is not complete and that further development may be possible. Some possible additional developments will be discussed in the conclusion chapter of this thesis (Chapter 9).

# Chapter 9: Conclusions and Future Directions

*"Would you tell me, please, which way I ought to go from here?"*

*"That depends a good deal on where you want to get to," said the Cat.*

*"I don't much care where –" said Alice.*

*"Then it doesn't matter which way you go," said the Cat.*

*"– so long as I get somewhere," Alice added as an explanation.*

*"Oh, you're sure to do that," said the Cat, "if you only walk long enough."*

***Lewis Carroll, British author (1832- 1898).***
***Alice's Adventures in Wonderland. Pig and Pepper.***

This chapter provides a summary of the contributions made by the work presented in this thesis and discusses some areas for possible future work.

## 9.1. Summary of Contributions

The aim of this thesis, as stated in the introductory chapter, was to attempt to answer the following question:

> *By what means may a model, that enables software entities to make use of their accumulated experience to jointly reason about a given situation, be realised; and how might such a model be evaluated?*

Throughout the preceding chapters a number of issues were addressed that all contribute towards answering the above question and also attend to the more specific research goals set out in Section 1.2. The contributions of this thesis, and how they address the identified research goals, are summarised in this section.

In Chapter 3 a theory of "*Arguing from Experience*" was articulated for use in situations involving a number of software agents (entities) reasoning about coming to a "*view*" about some case from a given domain. Each agent was equipped with a separate collection of instances from the same domain, and this set was assumed to represent this agent's experience in the domain under consideration. One of the aims of the proposed theory was to provide a coherent, and easy to evaluate, means to represent and warrant "*Arguments from Experience*". The proposed theory catered for this objective by exploiting association rule mining techniques to discover associations between features of the case under consideration and a consequent "*view*" of this case proposed according to the previous experience. A "*view*" on a current example was identified as being akin to a classification of a given case, thus enabling a pragmatic application of the promoted theory. The theory advocated contains a number of features that enable it to deal with defeasible reasoning:

- It is represented as inductive reasoning whereby a justification for a proposed "*view*" can be structured into an argument scheme with associated critical questions.
- It borrows elements from a number of schemes and builds upon them to form a new scheme: the "*Argument from Experience based on Classification*" (AEC/AEC2) scheme. This scheme defeasibly justifies a desired claim relating to the case under discussion by the means of association rules linking some features in the case to the claim.
- It enables the agents to mine ARs from a collection of past examples embodying the agent's experience. These examples provide the backing for the arguments resulting from the AEC/AEC2 scheme.

The persuasive element is dealt with through a comprehensive list of speech acts corresponding to one version of the proposed scheme (AEC2) and the critical questions associated with it. These speech acts form the proposed dialogue model and are posed by the different parties in the dialogue (each of which is the advocate of a different "*view*") in an attempt to persuade the others that either the "*views*" they promoted do not hold in the given case, or that other "*views*" are more suited to the given case. The underlying association rules in each speech act reflect either a criticism that can be posed against a proposed "*view*" or an association rule supporting a given "*view*". A dialogue game protocol, for facilitating "*Arguing from Experience*", was also presented. Each agent, taking part in these games, can choose which speech act to utter at each stage of the game, provided that it follows the rules of the protocol. In defining this theory the first of the research goals of this thesis was addressed:

> *To provide a theory of persuasion within the setting of reasoning from experience that accounts for the defeasible nature this style of reasoning, and to provide the means by which the advocated theory can be implemented to enable different participants to draw arguments directly from their past experience.*

The proposed theory was intended to be applied in situations where participants have not analysed their experiences into rules in a knowledge (belief) base, but draw directly on their experience, presented by a set of examples, to find reasons for coming to a view on some current example. This approach was argued to have several advantages:

- It provides a natural means to represent arguments, we often argue from our experience by drawing on this experience and encapsulating it in statements such as "*every time we have done x, y happened*" or "*All Xs we have encountered thus far were Ys, therefore any new X is likely to be Y*" and then deploying these statements in an argument.
- It provides a means for avoiding the knowledge engineering bottleneck that occurs when knowledge bases are constructed. Additionally, there is no need to commit to a theory in advance of the discussion. The information can be deployed as best meets the need of the current situation. Moreover no revision of the knowledge base is required when new cases are added.
- It provides an invaluable mode of reasoning, especially where it is not possible to use other types of reasoning, such as proof or reasoning from beliefs. For instance, if a belief base cannot be manually constructed,

because it requires extensive consultation with experts, or is simply prohibitively expensive to hand-craft.

- It allows agents to benefit from the differences in their experiences, thus enhancing their overall knowledge of the world they inhabit.

In Chapter 4 the above theory was articulated for two-party dialogues. This manifestation was referred to as PADUA and it enables persuasive dialogues to be undertaken by two participants so as to come to a conclusion regarding the most suitable classification of a given case. Proponents of a possible classification may state and justify their proposals in the form of the AEC2 argument scheme, and the opponent may attack this position according to the speech acts presented in Chapter 3. The result of dialogue games of this form is the classification of the examples under consideration as proposed by the winning party. Additional issues were also addressed in Chapter 4. A four layer strategy model was presented and PADUA was extended to allow for nested dialogues to take place over intermediate precedents. Details were also given of how PADUA was implemented using Java. This implementation successfully encoded the protocol, as was shown by a number of examples. Developing PADUA presented a step towards the realisation of the second research goal:

> *To show how this theory can be transformed into a computational framework that can be effectively deployed in autonomous software systems to enable two-party dialogues for "Arguing from Experience".*

Chapters 6 and 7 gave details of how multiparty "*Arguing from Experience*" can be achieved through the PISA Framework. PISA allows for any number of agents to take part in dialogues regarding the classification of cases from some domain. The innovative contribution of PISA is the mechanisms whereby it answers some of the challenges found in multiparty dialogues. PISA embodies a number of notable features. In particular, the control structure, the turn taking policy, the approach to game termination and the definition of the roles of the participants allowing them to adopt differing strategies. The supporting argumentation tree data structure is also significant. In summary, PISA offers several advantages, in addition to those featuring in PADUA:

- It allows argumentation amongst any number of participants rather than the more usual two.
- It leads to a reasoned consensus increasing the acceptability of the outcome to all parties.

In Chapter 6 both the design and the structure of PISA were fully described. Details were also given of how PISA was implemented using Java. Chapter 7 complemented the account of PISA with a number of additional features. In particular, an advanced dialogue strategy design, which exploits a dynamic view of the history of the dialogue, was tailored for PISA. The potential for participants to form dynamic groups, and the decision making process within the group, were also considered. Consideration was also given as to how these features were incorporated in the Java application. Developing PISA was intended to complete the answer to the second research goal:

> *To show how this theory can be transformed into a computational framework that can be effectively deployed in autonomous software systems to enable* <u>*multiparty*</u> *dialogues for "Arguing from Experience".*

Examples were given in Chapter 4, produced using the *PADUA GUI Application,* to illustrate the style of dialogues generated by PADUA. These examples touched upon variety of issues, such as intermediate concepts and the role of strategy in the resulting dialogues. Examples were also given in Chapters 6 and 7, obtained using the *PISA Application*, to provide a proof of concept of the multiparty "*Arguing from Experience*" embodied in PISA. Chapters 5 and 8 turned toward the pragmatic application of "*Arguing from Experience*", and in particular, the use of this mode of reasoning to generate a decision with respect to classifying cases from given domains. Extensive empirical assessment of both PADUA and PISA has shown that the dialogues produced using either of them led to a successful resolution of the issue at hand (correct classification of the given case) in a very high proportion of cases. The given account fulfils the goal of articulating the theory of "*Arguing from Experience*" in terms of enabling its computational use. The effectiveness of the account was demonstrated through the analysed experiments and examples. The production of example dialogues and the reported empirical analysis were intended to contribute toward answering to another issue relating to the second research goal:

> *To evaluate the two instantiations of the framework by applying the concept of "Arguing from Experience" to classification problems. Thus, incorporating the process argumentation into the field of data mining.*

The results of evaluating the automated process of "*Arguing from Experience*" using a number of classification problems encouraged further investigation. This investigation was intended to establish both PADUA and PISA as worthy classifiers. The intuition behind using "*Arguing from Experience*" to classifying "*unseen*" instances was to provide a "Meta" method by which a number of software agents (entities) can reason about the classification of a given case by means of arguments. The utilisation of "*Arguing from Experience*" as a classification technique had two objectives:

- To provide a pragmatic use of this style of reasoning that can be applied to solve many real–life situations.
- To provide means to test and evaluate the proposed model and its applications. One possible way to assess "*Arguing from Experience*" has focused, as discussed above, on the coherence and comprehensiveness of the resulting dialogues. Other criteria, however, include: (i) the results obtained from these dialogues when applied to classification problems (as here there exist right or wrong answers), and (ii) the quality of these dialogues. The promoted utilisation addresses these criteria.

The analysis discussed in Chapters 5 and 8, in addition to demonstrating the operation of the "*Arguing from Experience*" concept, indicated that both PADUA and PISA perform well as classifiers. This was highlighted by a collection of experiments, in which both applications obtained classification's accuracy above 90% in both clean and noisy settings. The results of this empirical analysis suggested the possibility of exploiting "*Arguing from Experience*" as a classification technique that could compete with other well known classifiers. Moreover, unlike the other classifiers used for comparison, the advocated approach enjoys some desirable features such as:

- It does not require a training phase. Further, it produces a limited number of rules sufficient to classify the given case without the need to generate all the possible rules.
- It is noise tolerant and can cope with high levels of noise in the datasets without failing to classify correct cases.
- It provides a wide range of configuration parameters; by changing some of these parameters the user can modify the course of dialogues, in both systems, to better fit the underlying datasets.

Taking these advantages into consideration, as well as the results obtained from the reported empirical analysis, it is suggested that the fourth research goal of this thesis has also been addressed:

> *To assess the application of "Arguing from Experience" to classification problems by means of comparative empirical experiments; thus, providing means to evaluate the promoted incorporation of argumentation and data mining.*

The results and investigations presented in this thesis were founded upon a number of different research areas. The contributions that were made in addressing these issues have produced a comprehensive model for arguing on the basis of past experience and have shown how this can be computationally represented and implemented for use in two-party and multi-party dialogues. A by-product of the promoted model for "*Arguing from Experience*" was the successful application of this model to classification problems, which was proven to be competitive with other renowned classification methods covering a wide range of paradigms, when evaluated in a variety of domains and situations.

## 9.2. Future Directions

The results presented in this thesis have suggested a number of possible directions for future work. Firstly, one interesting avenue to pursue would be the potential use of the model for "*Arguing from Experience*" in domains other than tabular databases. One interesting field for providing experience to the arguing agents would be document and web page collections. Association rules (ARs), which were incorporated into the promoted model as means to warrant arguments can be mined from either texts or web pages. This may prove to be a rich field for further developing the work undertaken in this thesis:

- Experience can be represented by means of texts rather than the more straightforward tabular datasets. For instance, legal proceedings of past cases may provide extensive experience for those who read and analyse them. However, the textual representation of experience comes with a number of challenges, mainly concern the need for a method for extracting useful information to warrant arguments in the promoted model. One possible solution is the application of text mining techniques to extract various interesting, previously unknown and potentially useful knowledge, particularly in the form of ARs, from sets of collected textual data.
- The emergence of web technologies has led the World Wide Web to become the default platform for delivering interactive information systems to both professionals and the public. The work presented in this thesis may be extended by equipping the arguing parties with the means to undertake web content mining, which will in turn provide them with means to extract structured data from web pages.

The extension of the existing model to make use of either web pages or document collections is potentially of great benefit. In order to make an assessment of the most suitable techniques required to generate arguments from the documents and to successfully incorporate them in the promoted model a wide range of issues should be addressed. This is in itself a clear area for future work. Such an implementation and analysis would present a significant task and is thus outside the scope of this thesis.

A second area that may be addressed in future work is the analysis of potential new speech acts to be incorporated in the model for "*Arguing from Experience*" presented in Chapter 3. One possible speech act that could be added to the promoted model would be the "*weakest link attack*" briefly discussed in Section 4.3. This speech act will enable one participant to undermine a previously played move on the basis of the fact that this participant has different confidence in the content of this move, or in one of its components (if it was an accrual of arguments). The attacked party can then respond to this move by proposing a new move or withdraw the unwanted component from the original move. Another possible speech act would be to address the notion of online ARM (Aggarwal and Yu, 1998). For instance, new speech acts could be incorporated into the proposed model by which one participant asks another to change the support value used to generate the underlying data structure so that different set of rules can be mined. This request could be justified by the fact that the given participant applies a very low support value. Another speech act could be integrated such that the party which has been asked to change its support value could challenge this request by demanding the requesting parties to change their support values as well. These speech acts present but an example for further extension of the proposed model. However, the treatment of the notion of online ARM in the applications of "*Arguing from Experience*" given in this thesis remains incomplete should the issue of variable support remain unanswered.

Another potential area for future work would be to develop a dynamic strategy model for "*Arguing from Experience*" dialogues. The promoted layered strategy models have provided the agents with the means to select moves, such that agents can attempt to look for ARs that support certain types of speech acts from the theory of reasoning from experience according to their underlying strategies. This model also provided criteria for ARs selection. However, the application of the proposed strategy models is fixed: once the strategy for a given agent is chosen it cannot be changed, even if it produces unsatisfactory results, for this particular agent, in a given domain. A useful extension of this strategy model would be to allow the agents to dynamically select the best strategy to be applied when tackling cases from a given domain. One approach to this issue would be to enable the agents to change their strategies when appropriate. A second approach would be to supplement both PISA and PADUA by heuristics, obtained from a variety of domains, such that participants taking part in either system could select their strategies on the basis of these heuristics according to the nature of the domain. A *Meta* description of each domain may help the agents in selecting their strategies, even for new domains, for which there are no available heuristics. However, in order to make a thorough assessment of each possible strategy to obtain the required heuristics, a fairly large number of real-world applications involving implemented real-agents arguing over a wide range of domains would be required. Such an implementation and analysis would present a significant task and is thus outside the scope of this thesis.

A fourth area for further investigation would be to apply "*Arguing with Experience*" to cases where the involved agents happen to share some parts of their experience (or the whole experience is shared amongst these agents). Such application is interesting because it may lead to discover only the best rules in the shared experience, and thus will have a pragmatic appeal in both ARM and Machine Learning paradigms.

A fifth area to address in future work would be to extend elements of the propose account for "*Arguing from Experience*" that do not currently form the primary focus, but would provide useful extensions to it, were they to be developed further. One possibility would be to investigate treatments for situations in which each participant applies different representations of the same data. For instance, the order of the items in each dataset may differ, or the name of these items, their assigned values, or even the items held by different databases. Furthermore, some agents may use additional attributes, possibly resulting from each agent applying a different discretisation mechanism. Here, one potential treatment would be to make use of Meta descriptions, such as ontology of the underlying datasets, such that each agent could match their own datasets against these descriptions. This area of future development would need a thorough analysis, and may require expert consultation. Other extensions could also prove beneficial. Another interesting extension would be to investigate multi-objective aspects of "*Arguing from Experience*" in rules discovery. For instance, applying the promoted model to maximise both the support and confidence of the discovered rules is one possible way to apply "*Arguing from Experience*" to multi-objective association rules mining (e.g. (Ishibuchi and Nojima, 2005)). This can be done by modifying the current model slightly so that the discovered rules should satisfy more conditions in order for them to be accepted as *legal* rules in the argumentation process.

Thus far general extensions to the proposed model have been discussed. However, this thesis has developed two separate applications to embody "*Arguing from Experience*" theory: PADUA for two-party dialogues and PISA for multi-party dialogues. Each application could benefit from further improvements, and each offers different possibilities for future work. For instance, the analysis in Chapter 4 has shown that PADUA could be applied to generate deliberation rather than persuasion dialogues. This area merits further examination, the results of which may confirm the potential of using PADUA to treat different types of dialogues, such as information seeking or inquiry dialogues. Another instructive area for future investigation, with respect to PADUA, would be to examine extensions for the control layer. The intention behind this layer was to provide a means to control the instantiation of each dialogue. However, the current application of PADUA makes a very little use of this layer. A comprehensive extension and proper implementation of the notion of this layer would prove beneficial to PADUA, and could make it more applicable to real-life situations.

In the case of PISA, a number of possibilities for future improvements were highlighted in Chapter 7. The current application of PISA does not accommodate for a number of features, such as *biased agreeable profiles* and *temporary coalitions*, associated with the strategy model described in Section 7.1. An obvious extension of PISA would be to consider implementing these features. One important requirement would be an analysis of the underlying domain to identify which classifications are closer to one another, thus following the notion of adjacency as explained in Chapter 7. Once the relation among the classes has been identified, an appropriate representation would be required, so that the included agents can make use of it. In order to facilitate temporary coalitions amongst a number of participants within PISA a number of issues need to be addressed: (i) the formation process of coalitions between a number of participants; (ii) whether the coalitions should be dynamic or not; and (iii) the process by which an existing coalition is dismantled.

Unlike the *biased agreeable profiles*, coalition requires mutual agreement among a number of participants, thus a preparation step is necessary. The suggested process is as follows: when one participant requests an alliance with another participant, the receiving participant assesses the situation and replies with either yes or no. If the answer is yes, then the two participants form a coalition. If another participant asks to join with any of the participants in the coalition, then a similar process to the above is repeated, but this time the participant receiving the request, will also pass this request to the other parties in the coalition, which will then make a decision whether to accept the new party or not. This process also addresses the second issue: coalitions are assumed to be dynamic in PISA such that participants may join and leave at any time. Also, the process whereby an existing coalition is dismantled relates to the initial objective of forming coalitions. Coalitions are formed for strategic reasons. In particular, if one participant emerges as a strong opponent to some of the other participants, then these participants may form a coalition against this participant. The objective of this coalition will be, for instance, to attempt to remove this participant from the dialogue. In this case the coalition would be dismantled once the participant in question leaves the game. However, a number of questions require attending to for a successful termination of coalitions. For instance: are the termination conditions agreed when the coalition is formed? Does everyone have to agree to the coalition ending? Further consideration may be given to identifying strong opponents in a given dialogue. The criteria for seeking coalition with other participants should also be determined. Once these two concepts are decided upon, they can be integrated in the control layer of PISA, either through the chairperson agent or by other means. It is anticipated that a successful implementation of both concepts would prove beneficial to applying PISA to real life problems. However, the above description does not attend to the following issues: Will other players know about which coalitions are in effect? Will they know when other players are trying to form coalitions?

And can there be competition with respect to coalitions' formation? These questions should be answered, should a successful integration of coalitions within PISA be sought.

The areas that have been identified here for possible future directions are just some of the options presented by the work detailed in this thesis. There are also a number of interesting sub-issues that would benefit from further investigation. In particular those related to the weakness of the promoted model for "*Arguing from Experience*" and the associated applications (PADUA and PISA). Some of these weaknesses are summarised as follows:

- The promoted model lacks a precise notion of agency. For instance the model does not take into consideration the internal state of the involved agents, and how this state may change during the course of "*Arguing from Experience Dialogues*".
- The model (in particular the AEC2 scheme (Section 3.1)) is tailored for the support/confidence framework (Agrawal et al., 1993). While this interest measure have proven to be popular and reliable in mining association rules. Other measures of interests may also prove useful.
- The promoted model generates certain type of association rules (rules with class attribute in their conclusion). Expanding the model to generate all types of association rules may be necessary to tackle some situations, and to enable more sophisticated types of arguments (accruals for example).
- The central structure in PISA could prove to be an obstacle if PISA was to be used in real-life multi agent environment. Future work needs to be done in order to enable the distribution of the central argumentation tree such that it can be accessed by a number different agents residing on different machines.
- PISA lacks the sufficient means to enable fully open dialogues. The current implementation of PISA requires the user to determine the agents that would be involved in dialogues over cases from particular domains. Some of these agents may be discarded by the chairperson if they remain *idle* for a given number of rounds. However, new agents are not allowed to join an ongoing

dialogue. This particular point merits further consideration, if PISA was to function as a real multi agent application.

- Additionally, in order for the advocated account to be deployed for classification problems; a proper study as to how classifiers could be generated using "*Arguing from Experience*" is essential. In particular, with relation to the "considerable" runtime required to classify each case, when compared with other classification and machine learning algorithms.

As for the last point, the promoted approach generate a limited number of rules sufficient to classify a given case, by subjecting each rule to critique from the other parties in the dialogue, so that only the strongest rules emerge undefeated. One interesting extension would be to store the undefeated rules from each dialogue and make use of them to classifying future instances. For instance, in PISA, such rules can be maintained by the chairperson, which will then trigger a dialogue if only no adequate rule, to match the new case, was found. This area of future work will enable the effective usage of PISA and PADUA, and may to reducing the execution time required to classify each case. Once a case is agreed upon, it could be added to the participants' corresponding experience, so they can benefit from the persuasion process to grow their own experience, and thus providing more reasons to believe that similar cases should be treated in the same manner as the cases previously discussed.

The results presented throughout this thesis are intended only to provide details of how "*Arguing from Experience*" can be dealt with in software entities such as autonomous agents. For agents to be fully autonomous in dealing with reasoning from experience the theory presented here also needs to be complemented by the different aspects of agency (Wooldridge, 2001), successful completion and evaluation of the proposed applications and the potential extensions discussed above. Nonetheless, the findings reported in this thesis are believed to provide sufficient analysis of the concept of argumentation from experience, and its application to one particular problem, that of classification in data mining. The given account is intended to provide a treatment for this particular mode of reasoning, by means borrowed from a variety of fields, such that it can be built upon in the quest for the effective design and construction of realistic automated computer systems. Additionally, this thesis has shown possibilities for potentially beneficial application of the proposed treatment in the fields of classification and association rule mining.

# Bibliography

(Aamodt and Plaza, 1994) A. Aamodt and E. Plaza. Case-based reasoning; Foundational issues, methodological variations, and system approaches. In AICom - Artificial Intelligence Communications, vol.7(1),. IOS press (1994). pp: 39-59.

(Aggarwal and YU, 1998) C.C. Aggarwal and P.S. Yu. Online Generation of Association Rules. In: Proc. 4th Int. Conf. on Data Engineering (ICDE'98), IEEE, (1998). pp: 402-411.

(Agrawal and Srikant, 2000) R. Agrawal and R. Srikant. Privacy-preserving data mining. In Proc. ACM SIGMOD Conf. on Management of Data (SIGMOD'00). ACM Press, (2000). pp: 439 – 450.

(Agrawal et al., 1993) R. Agrawal, T. Imielinski and A. Swami. Mining association rules between sets of items in large databases. In Proc. ACM SIGMOD Conf. on Management of Data (SIGMOD'93). ACM Press, (1993). pp: 207 – 216.

(Ahmed, 2004) S. Ahmed. Strategies for partitioning data in association rule mining. Ph.D. Thesis, The University of Liverpool, UK, (2004).

(Ahuja et al., 1986) S. Ahuja, N. Carriero and D. Gelernter. Linda and Friends. In IEEE Computer, vol.19(8). IEEE Computer Society Press (1986). pp: 26–34.

(Allwwod, 1995) J. Allwood. Reasons for management in dialog. In R. J. Beun, M Baker, and M Reiner (editors), Dialogue and Instruction, Springer-Verlag, (1995). pp: 241–250.

(Aleven, 1997) V. Aleven. Teaching Case Based Argumentation Through an Example and Models. PhD thesis, University of Pittsburgh, Pittsburgh, PA, USA. (1997).

(Aleven, 2003) V. Aleven. Using background knowledge in case-based legal reasoning: a computational model and an intelligent learning environment. In Artif. Intell., vol. 150 (1-2), (2003). pp: 183-237.

(Ambroszkiewicz et al, 1998) S. Ambroszkiewicz, O. Matyja and W. Penczek. Team Formation by Self-Interested Mobile Agents. In Multi-Agent Systems, Lecutre notes in computer science, Springer (1998). pp: 1-15.

(Amgoud and Cayrol, 1998) L. Amgoud and C. Cayrol. On the acceptability of arguments in preferencebased argumentation. In Proc. 14th Conf. on Uncertainty in Artif. Intell.. Morgan-Kaufmann, (1998). pp: 1–7.

(Amgoud and Maudet, 2002) L. Amgoud and N. Maudet. Strategical considerations for argumentative agents (preliminary report). In Proc. 9th Int. Workshop on Non-Monotonic Reasoning (NMR'02). Toulouse, France, (2002). pp:. 409-417.

(Amgoud and Parsons, 2001) L. Amgoud and S. Parsons. Agent dialogues with conflicting preferences. In Proc. 8th Int. Workshop on Agent Theories, Architectures and Languages. Seattle, Washington, (2001). pp: 1-15.

(Amgoud et al., 2000a) L. Amgoud, N. Maudet, and S. Parsons. Modelling dialogues using argumentation. In Proc. 4th Int. Conf. on Multiagent Systems (ICMAS'00). Boston, MA. IEEE Press. (2000). pp: 31– 38.

(Amgoud et al., 2000b) L. Amgoud, S. Parsons, and N. Maudet. Arguments, dialogue, and negotiation. In Proc. Europ. Conf. on Aritf. Intell. (ECAI'00). Berlin, Germany, (2000). IOS Press. pp: 338–342.

(Amgoud et al., 2006) L. Amgoud, S. Belabbès, and H. Prade. A formal general setting for dialogue protocols. In: Proc. 12th Int. Conf. on Artif. Intell. (AIMSA'06). Varna, Bulgaria, (2006). pp: 13 - 15.

(Amir et al., 1997) A. Amir, R. Feldman and R. Kashi. A New and Versatile Method for Association Generation. In Proc. PKDD'97. Springer LNCS, (1997). pp: 221-231.

(Aristotle ,1938) Aristotle. Prior Analytics. Categories, On Interpretation, Prior Analytics. Loeb Classical Library, vol. I. Cambridge, MA: Harvard University Press, (1938).

(Aristotle, 1997) Aristotle. Topics. Clarendon Press, Oxford, UK, 1997. Translated by R. Smith.

(Ashley, 1990) K. D. Ashley. Modelling Legal Argument. MIT Press, Cambridge, MA, USA, (1990).

(Ashley and Aleven, 1991) K. D. Ashley and V. Aleven. Toward an Intelligent Tutoring System for Teaching Law Students to Argue with Cases. Tin Proc. 3rd Int. Conf. on AI and Law (ICAIL'91). ACM Press (1991). pp: 42 – 52.

(Ashley and Brüninghaus, 2003) K. D. Ashley and S. Brüninghaus. A Predictive Role for Intermediate Legal Concepts. In Proc. 16th Annual Conf. on Legal Knowledge and Information Systems (JURIX'03). IOS Press: Amsterdam (2003). pp: 153-162.

(Ashley and Rissland, 2003) K. D. Ashley and E. L. Rissland. Law, learning and representation. In Artif. Intell. vol.150(1-2), (2003). pp: 17-58.

(Atkinson, 2006) K. Atkinson. Value-based argumentation for democratic decision support. In Proc. 1st Conf. on Computational Models of Argument (COMMA '06). Liverpool, UK. IOS press (2006), pp: 47-58.

(Atkinson and Bench-Capon, 2005) Atkinson K., Bench-Capon T. J. M. Legal Case-based Reasoning as Practical Reasoning. In Journal of Artif. Intell. Law. vol. 13(1), (2005). pp: 93-131.

(Bel-Enguix and López, 2006) G. Bel-Enguix and D. J. López. Membranes as Multi-agent Systems: an Application to Dialogue Modelling. In Professional Practice in AI. Springer (2006). pp: 31 -40.

(Bench-Capon, 1991) T.J.M. Bench-Capon. Knowledge Based Systems Applied To Law: A Framework for Discussion. In Knowledge Based Systems and Legal Applications. Academic Press, (1991). pp: 329-342.

(Bench-Capon, 1993) T.J.M. Bench-Capon. Neural Nets and Open Texture. In Proc. 4th Int. Conf. on AI and Law (ICAIL'94). ACM Press: Amsterdam, (1993). pp: 292–297.

(Bench-Capon, 1997) T. J. M. Bench-Capon. Arguing with Cases. In Proc. 10th Annual Conf. on Legal Knowledge and Information Systems (JURIX'97). GNI, Nijmegen (1997). pp: 85-100.

(Bench-Capon, 1998) T. J. M. Bench-Capon. Specification and Implementation of Toulmin Dialogue Game. In Proc. 11th Annual Conf. on Legal Knowledge and Information Systems (JURIX'97). GNI, Nijmegen (1997). pp: 5-20.

(Bench-Capon, 2003) T. Bench-Capon. Persuasion in practical argument using value based argumentation frameworks. In Logic and Computation, vol.13(3). Oxford University Press (2003). pp: 429–48.

(Bench-Capon and Coenen, 2000) T.J.M. Bench-Capon and F. Coenen. An Experiment in Discovering Association Rules in the Legal Domain. In Proc. 11th Int. Workshop on Database and Expert Systems Applications. IEEE Computer Society: Los Alamitos, (2000). pp: 1056–1060.

(Bench-Capon and Prakken, 2006). T. Bench-Capon and H. Prakken. Argumentation. In Information Technology and Lawyers, Springer (2006). pp: 61-80.

(Bench-Capon and Sergot, 1989) T. Bench-Capon and M. Sergot. Towards a Rule Based Representation of Open Texture in Law. In Computing Power and Legal Reasoning, Greenwood Press (1989). pp: 39-60.

(Bench-Capon and Staniford, 1995) J. T. M. Bench-Capon and G. Staniford. PLAID - Proactive legal assistance. In Proc. In Proc. 5h Int. Conf. on AI and Law (ICAIL'95). ACM Press: New York, (1995). pp: 81-88.

(Bergenti, and Ricci, 2002) Bergenti, F., Ricci, A.: Three approaches to the coordination of multiagent systems. In Proc. ACM symposium on Applied computing, ACM Press, Madrid, Spain (2002).

(Black and Hunter, 2007) E. Black and A. Hunter. A generative inquiry dialogue system. In Proc. AAMAS'07. Honolulu, Hawaii, (2007).

(Blake and Merz, 1998) C.L. Blake and C.J. Merz. UCI Repository of machine learning databases http://www.ics.uci.edu/~mlearn/MLRepository.html, Irvine, CA: University of California, Department of Information and Computer Science, (1998).

(Bohanec and Rajkovic, 1990) M. Bohanec and V. Rajkovic. Expert system for decision making. In Sistemica, Vol 1(1) (1990). pp:145–57.

(Breiman, 1996 ) L. Brieman. Bagging predictors. In Machine Learning, vol. 24 (1996). pp: 123-140.

(Brin et al., 1997) S. Brin, R. Motwani, J. D. Ullman and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'97). Tucson, Arizona. ACM Press (1997). pp: 255 - 264.

(Brodley and Friedl, 1999) C. E. Brodley and M.A Friedl. Identifying Mislabeled Training Data. In Artif. Intell. Research vol.11, (1999). pp: 131–167.

(Bruninghaus and Ashley, 2003) S. Bruninghaus and K. D. Ashley. Predicting Outcomes of Case-based Legal Arguments. In Proc. 9th Int. Conf. on AI and Law (ICAIL'03). ACM Press. New York, (2003). pp: 233–242.

(Burdick et al., 2001) D. Burdick, M. Calimlim and J. Gehrke. MAFIA: A maximal frequent itemset algorithm for transactional databases. In Proc. 17th Int. Conf. on Data Engineering (ICDE'01). Heidelberg, Germany, IEEE (2001). pp: 443 – 452.

(Carlson, 1983) L. Carlson. Dialogue games: an approach to discourse analysis. Reidel Publishing Company, Dordrecht, (1983).

(Carnap, 1952) R. Carnap. The Continuum of Inductive Methods, Chicago: The University of Chicago Press (1952).

(Carriero and Gelernter) N. Carriero and D. Gelernter. The Linda alternative to message-passing systems. In Parallel Computing vol.20(4). Elsevier Science Publishers B. V. Amsterdam, The Netherlands (1994). pp: 632–655.

(Cayrol et al., 2003) C. Cayrol, S. Doutre and J. Mengin. On Decision Problems Related to the Preferred Semantics for Argumentation Frameworks. In Logic and Computation vol.13(3), Oxford University Press, (2003). pp: 377- 403.

(Chang et al., 2006) G. Chang, M. Healey, J. A. M. Mchugh and T. L. Wang. Mining the world wide web – an information search approach. Kluwer Academic Publishers, Norwell, MA, USA, (2006).

(Chesñevar et al., 2000) C. Chesñevar, A. Maguitman, R. Loui and R. Prescott Loui. Logical Models of Argument. In ACM Computing Surveys, vol.32, (2000). pp: 337 – 383.

(Chorley and Bench-Capon, 2005) A. Chorley and T.J.M. Bench-Capon. AGATHA: Using heuristic search to automate the construction of case law theories. In Journal of AI and Law. vol. 13(1), Springer (2005). pp: 9-51.

(Clark and Boswell, 1991) P. Clark and R. Boswell. Rule induction with CN2: Some recent improvements. In Proc. 6th Euro. Working Session on Learning. Porto, Portugal. Springer-Verlag (1991). pp. 151-163.

(Clark and Niblett, 1989) P. Clark and T. Niblett. The CN2 induction algorithm. In Machine Learning, vol. 3(4), Springer (1989). pp: 261-283.

Bibliography.

(Coenen, 2003) Coenen, F. The LUCS-KDD Discretised/normalised ARM and CARM Data Library, http://www.csc.liv.ac.uk/~frans/KDD/Software/LUCS_KDD_DN/, Department of Computer Science, The University of Liverpool, UK. (2003).

(Coenen, 2004a) F. Coenen. The LUCS-KDD TFPC classification association rule mining algorithm. Department of Computer Science, University of Liverpool, 2004.

(Coenen, 2004b) Coenen, F. The LUCS-KDD Association Rule Mining Algorithm, http://www.csc.liv.ac.uk/~frans/KDD/Software/Apriori_TFP/aprioriTFP.html, Department of Computer Science, The University of Liverpool, UK. (2004).

(Coenen, 2004c) F. Coenen. LUCS-KDD implementations of the FOIL, PTM and CPAR algorithms, http://www.cxc.liv.ac.uk/~frans/KDD/Software/FOIL_PRM_CPAR/, Department of Computer Science, The University of Liverpool, UK, (2004).

(Coenen and Leng, 2002) F. Coenen, and P. Leng. Finding association rules with some very frequent attributes. In Proc. 6th Euro. Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'02). Helsinki, Finland, (2002). pp: 99 – 111.

(Coenen and Leng, 2004) F. Coenen, and P. Leng. An evaluation of approaches to classification rule selection. In Proc. 4th Int. Conf. on Data Mining (ICDM’04). IEEE (2004). pp 359 – 362.

(Coenen and Leng, 2005) F. Coenen and P. Leng. Obtaining Best Parameter Values for Accurate Classification. In Proc. 5th Int. Conf. on Data Mining (ICDM'05), IEEE(2005). pp: 597-600.

(Coenen and Leng, 2006) F. Coenen, and P. Leng. Partitioning Strategies for Association Rule Mining. In The Knowledge Engineering Review, vol. 21(1), Cambridge University Press (2006). pp: 25-47.

(Coenen et al., 2001) F. Coenen, G. Goulbourne and P. Leng. Computing association rules using partial totals, In Proc. 5th Euro. Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'01). Freiburg, Germany, (2001). pp: 54 – 66.

(Coenen et al., 2004a) F. Coenen, P. Leng and S. Ahmed. Data structure for association rule mining: T-trees and p-trees. In IEEE Transactions on Knowledge and Data Engineering, vol.16(6) (2004). pp: 774 – 778.

(Coenen et al., 2004b) F. Coenen, P. Leng and G. Goulbourne. Tree structures for mining association rules. In Data Mining and Knowledge Discovery, vol 8(1). Springer (2004). pp: 25 – 51.

(Coenen et al., 2005) F. Coenen, P. Leng and P. and L. Zhang. Threshold Tuning for Improved Classification Association Rule Mining. In Proc. 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining (*PAKDD’05*). Springer, (2005). pp: 216-225.

(Cogan et al., 2006) E. Cogan, S. Parsons, and P. McBurney. New types of inter-agent dialogs. In Proc. 3rd Int. Workshop on Argumentation in Multiagent Systems (ArgMAS'06). Springer (2006). pp: 154–168.

(Cohen and Levesque, 1990) P. Cohen and H. Levesque. Intention is choice with commitment. In Artif. Intell., vol. 42(2–3), Springer (1990). pp:213–261.

(Cohen et al, 1999) P. Cohen, H. Levesque and I. Smith. On Team Formation. In Contemporary Action Theory. Synthese, Kluwer Academic Publishers (1999). pp: 87-114.

(Dignum and Vreeswijk, 2004) F. Dignum and G. Vreeswijk. Towards a testbed for multi-party dialogues. In Advances in Agent Communication, Int. Workshop on Agent Communication Languages, (ACL'03), Melbourne, Australia, (2003). Lecture Notes in Computer Science. Springer vol. 2922, (2004). pp: 212-230.

(Dignum et al., 2001) F. Dignum, B. DuninKeplicz, and R. Verbrugge. Agent theory for team formation by dialogue. In Intelligent Agents VII. Springer (2001). pp: 141–156.

(Doutre et al., 2005) S. Doutre, P. McBurney and M. Wooldridge. Law-governed Linda as a semantics fo ragent dialogue protocols. In Proc. 4th Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS'05). Utrecht, the Netherlands. ACM Press, New York. pp: 1257–1258.

(Dung, 1995) P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. In Artif. Intell., vol. 77 Springer (1995). pp:321–357.

(Efron and Tibshirani, 1993) B. Efron and R. Tibshirani. An Introduction to the Bootstrap. Chapman and Hall, New York, (1993).

(Eggermont et al., 2004) J. Eggermont, J. Kok and W. A. Kosters. Genetic Programming for data classification: partitioning the search space. In *Proc. ACM Symposium on Applied Computing (*SAC '04). Nicosia, Cyprus. ACM, New York, NY(2004). pp: 1001-1005.

(Eschrich et al, 2002) S. Eschrich, N. V. Chawla and L. O. Hall. Generalization Methods in Bioinformatics. In Proc. 2nd ACM SIGKDD Workshop on Data Mining in Bioinformatics (BIOKDD'02), Edmonton, Alberta, Canada (2002). pp: 25 – 33.

(Feldman and Sanger, 2006) R. Feldman and J. Sanger. The text mining handbook: Advanced approaches in analyzing unstructured data. Cambridge University Press, (2006).

(Frawley et al. 1991) W.J. Frawley, G. Piatetsky-Shapiro and C.J. Matheus, C.J. Knowledge discovery in databases: An overview. In Knowledge Discovery in Databases, AAAI/MIT Press, (1991). pp: 1 – 27.

(Freund, 1999) Y. Freund. An adaptive version of the boost by majority algorithm. In Proc. COLT '99: the 12th annual Conf. on Computational learning theory. Santa Cruz, California, United States (1999). pp: 102-113.

(Freund and Schapire, 1997) Y. Freund R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In Journal of Comput. Syst. Sci., vol 55(1). Academic Press, Inc. Orlando, FL, USA (1997). pp:119-139.

(Fuller, 1985) L . Fuller. Positivism and Fidelity to Law - A Reply to Professor Hart. 71 Harv. L. Rev. 630, (1958).

(Gamberger, 1999) D. Gamberger, N. Lavrac and C Groselj. Experiments with Noise Filtering in a Medical Domain. In Proc. 16th Int. Conf. on Machine Learning (ICML'99). San Francisco, CA, (1999). pp:143–151.

(Gardner, 2001) Gardner M. A skeptical look at Karl Popper. In the Skeptical Inquirer, vol.25, (2001). pp: 13-14, 72.

(Gionis et al., 2007) A. Gionis, H. Mannila and P. Tsaparas. Clustering aggregation. In ACM Trans. Knowl. Discov. Data. vol.1(1). ACM(2007). pp: 4, DOI= http://doi.acm.org/10.1145/1217299.1217303.

(Gómez and Chesñevar, 2004 ) S.A. Gómez and C. I. Chesñevar. Integrating defeasible argumentation and machine learning techniques. Technical report, Universidad Nacional del Sur, Bahia Blanca (2004).

(Goffman, 1981) E. Goffman. Forms of Talk. University of Pennsylvania Press, (1981).

(Gordon, 1991) T. F. Gordon. An abductive theory of legal issues. In Man- Machine Studies, vol. 35, (1991). pp: 95-118.

(Gordon, 1995) T. F. Gordon. The Pleadings Game. An Artif. Intell. Model of Procedural Justice. Dordrecht/Boston/London: Kluwer Academic Publishers, (1995).

(Gorman, 2005) R. Gorman. The Socratic method in the dialogues of Cicero. Published by Steiner Verlag (2005). pp: 43 – 50.

(Goulbourne et al., 2000) G. Goulbourne, F. Coenen, and P. Leng. Algorithms for computing association rules using a partial-support tree. In Knowledge-Based Systems, vol.13 Elsevier (2000). pp:141 – 149.

(Governatori and Stranieri, 2001) G. Governatori and A. Stranieri. Towards the App:lication of Association Rules for Defeasible Rules Discovery. In Proc. 14th annual Conf. on Legal Knowledge and Information Systems (JURIX'01). IOS Press: Amsterdam (2001). pp: 63-75.

(Groothius and Svensson, 2000) M. Groothius and J. Svensson. Expert System Support and Juridical Quality. In Proc. 13th annual Conf. on Legal Knowledge and Information Systems (JURIX'00). IOS Press: Amsterdam (2000). pp: 1–10.

(Grahne and Zhu, 2003) G. Grahne and J. Zhu. Efficiently using prefix-trees in mining frequent itemsets. In Proc. Worksh0p on Frequent Itemsets MIning (FIMI'03). Melbourne. IEEE Press, (2003).

(Hage, 1996) J. Hage. A theory of legal reasoning and a logic to match. In AI and Law vol. 4, Springer (1996). pp: 199–273.

(Hall et al., 2009) M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten. The WEKA Data Mining Software: An Update. In SIGKDD Explorations, Vol.11(1), (2009).

(Hamblin, 1970) C. L. Hamblin. Fallacies. Methuen, London, UK, (1970).

(Han and Kamber, 2006) J. Han and M. Kamber. Data mining: Concepts and techniques (2nd Edition). Morgan Kaufmann, (2006).

(Han et al., 2000) J. Han, J. Pei and Y. Yin. Mining Frequent Patterns Without Candidate Generation. In Proc. ACM SIGMOD Int. conf. on Management of Data (SIGMOD'00). ACM Press, (2000). pp: 1 – 12.

(Hart, 1985) H L A. Hart . Positivism and the Separation of Law and Morals. 71 Harv. L. Rev. 593, (1958).

(Hayes and Roth, 1985) B. Hayes-Roth et al. Blackboard architecture for control. In Artif. Intell., vol 26, (1985). pp:251–321.

(Hickey, 1996) R. Hickey. Noise Modeling and Evaluating Learning from Examples. In Artif. Intell., vol.82(1–2), (1996). pp: 157–179.

(Hidber, 1999) C. Hidber (1999). Online association rule mining. In Proc ACM SIGMOD Int. Conf. on Management of data, (SIGMOD'1999). pp: 145-156.

(Hilderman and Hamilton, 1999) R. J. Hilderman and H. J. Hamilton. Heuristic measures of interestingness. In Proc. 3rd Euro. Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'99). Springer, Berlin. (1999). pp: 232–241.

(Hotho et al., 2005) A. Hotho, A. Nürnberger and G. Paaß, A brief survey of text mining. LDV Forum. In GLDV Computational Linguistics and Language Technology, vol. 20(1), (2005). pp: 19 – 62.

(Hume, 1902) D. Hume. An Enquiry Concerning Human Understanding. Oxford, (1902). First published 1751.

(Hunter, 2006) A. Hunter. Presentation of Arguments and Counterarguments for Tentative Scientific Knowledge. In Proc. 2nd Int. Workshop on Argumentation in Multiagent Systems (ArgMAS'05). Springer, (2006). pp: 245 – 236.

(Ishibuchi and Nojima, 2005) H. Ishibuchi and Y. Nojima. Accuracy-Complexity Tradeoff Analysis by Multiobjective Rule Selection. In: Proc. Workshop on Computational Intelligence in Data Mining (ICDM'05). (2005) pp: 39–48.

(Jeffrey, 2004) R. Jeffrey. Subjective Probability (The Real Thing). Cambridge, England: Cambridge University Press, (2004).

(Johnston and Governatori, 2003) B. Johnston and G. Governatori. Induction of Defeasible Logic Theories in the Legal Domain. In Proc. In Proc. 9th Int. Conf. on AI and Law (ICAIL'03). ACM Press, (2003). pp: 204–213.

(Kakas et al., 2004) A. C. Kakas, N. Maudet and P. Moraitis. Layered Strategies and Protocols for Argument-based Interactions. In Proc. 1st Int. Workshop on Argumentation in Multiagent Systems (ArgMAS'04). pp: 64-77.

(Katzav and Reed, 2004) J. Katzav and C. Reed. On argumentation schemes and the natural classification of arguments. In Argumentation, vol.8(2), Springer (2004). pp:239–259.

(Kienpointner, 1986) M. Kienpointner. Towards a typology of argument schemes. In Proc. Int. Conf. of the Society for the Study of Argument (ISSA'86). Amsterdam University Press: Amsterdam, (1986).

(Kim and Park, 2004) H. Kim and H. Park: Data Reduction in Support Vector Machines by a Kernelized Ionic Interaction Model. In Proc. 4th SIAM Int. Conf. on Data Mining (SDM'04). Vista, Florida (2004). pp: 507-512.

(Kinny et al, 1992) D. Kenny, M. Ljungberg, A. Rao, G. Tidhar, E. Werner and E. Sonenberg. Planned Team Activity. In Proc. 4th Euro. Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW '92). Springer, London, (1994). pp: 227-256.

(Kneale, 1949) W. Kneale. Probability and Induction. Clarendon Press, (1949).

(Leung and Parker, 2003) K. T. Leung and S. D. Parker. Empirical comparisons of various voting methods in bagging. In Proc. 9th ACM SIGKDD Int. Conf. on Knowledge discovery and data mining (KDD'03). Washington D.C. ACM(2003). pp: 595-600.

(Li et al, 2001) W. Li, J. Han and J. Pei. CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules. In Proc. In Proc. 1st Int. Conf. on Data Mining (ICDM'01). San Jose, California, USA. (2001). pp: 369-376.

(Li et al., 2004) J. Li, G. Dong, K. Ramamohanarao and L. Wong. DeEPs: A New Instance-based Discovery and Classification System. In Machine Learning. Springer (2004), vol.54(2). pp: 99-124.

(Lindahl and Odelstad, 2005) L. Lindahl and J. Odelstad. Normative positions within an algebraic approach to normative systems. In Applied Logic, vol. 17(2), (2005). pp: 63-91.

(Liu et al, 1998) B. Liu, W. Hsu and Y. Ma. Integrating Classification and Association Rule Mining. In Proc. 4th Int. Conf on Knowledge Discovery and Data mining (KDD'98). New York, AAAI (1998). pp: 80-86.

(MacKenzie, 1979) J. D. MacKenzie. Question-begging in non-cumulative systems. In Philosophical Logic, vol.8, (1979). pp:117–133.

(McBurney and Parsons, 2001) P. McBurney and S. Parsons. Representing epistemic uncertainty by means of dialectical argumentation. In Annals of Mathematics and Artif. Intell., vol. 32(1–4), Springer: Amestredam(2001). pp:125–169.

(McBurney and Parsons, 2002) P. McBurney and S. Parsons. Games That Agents Play: A Formal Framework for Dialogues between Autonomous Agents. In logic, language and information, vol. 11(3), (2002). pp: 315-334.

(McBurney et al., 2003) P. McBurney, R. Eijk, S. Parsons, and L. Amgoud. A dialogue-game protocol for agent purchase negotiations. In Journal of Autonomous Agents and Mutilagent Systems (AAMAS), vol. 7(3), Springer (2003). pp:235– 273.

(McCarty and Sridharan, 1982) L. T. McCarty and M. S. Sridharan. A computational theory of legal argument. Technical Report LRP-TR-13, Computer Science Department, Rutgers University, (1982).

(McDonald et al, 2003) R.A. McDonald, D.J. Hand and I. A. Eckley. An Empirical Comparison of Three Boosting Algorithms on Real Datasets with Artificial Class Noise. In Multiple Classifier Systems, Springer (2003). pp: 161 - 170.

(Minsky and Papert, 1969) M. Minsky and S. Papert. Perceptrons: An Introduction to Computational Geometry. MIT Press, Cambridge MA. (1969).

(Miller and McBurney, 2007) T. Miller and P. McBurney. Using constraints and process algebra for specification of first class agent interaction protocols. In Engineering Societies in the Agents World VII, Lecture Notes in Artif. Intell. vol.4457. Springer, Berlin, (2007). pp: 245–264.

(Mohammadi and Gharehpetian, 2008) M. Mohammadi and G.B. Gharehpetian. Power System On-Line Static Security Assessment by Using Multi-Class Support Vector Machines. In Journal of Applied Sciences. vol. 8(12). (2008). pp: 2226 – 2233.

(Moore 1993) D. Moore. Dialogue game theory for intelligent tutoring systems. PhD thesis, Leeds Metropolitan University (1993).

(Mozina et al, 2005) M. Mozina, J. Zabkar, T. Bench-Capon and I. Bratko. Argument based machine learning applied to law. In Artif. Intell., vol. 13 (1), Springer (2005). pp: 53–73.

(Musgrave 2004) A. Musgrave. How Popper (might have) solved the problem of Induction. In Karl Popper: A Critical Appraisal. Roultedge (2004). pp: 16-27.

(Ogston et al, 2005) E. Ogston, B. Overeinder, M. van Steen and F. Brazier. Group Formation Among Peer-to-Peer Agents: Learning Group Characteristics. In Agents and Peer-to-Peer Computing, Springer (2005). pp: 59-70.

(Olave et al, 1989) M. Olave, V. Rajkovi, and M. Bohanec. An application for admission in public school systems. In Expert Systems in Public Administration (1989). pp: 145–160.

(Oliva et al, 2008a) E. Oliva, M. Viroli, A. Omicini and P. McBurney. Argumentation and artifact for dialogue support. In Proc. 5th Int. Workshop on Argumentation in Multiagent Systems (ArgMAS'08). Lisbon, Portugal, (2008). pp: 24-39.

(Oliva et al, 2008b) E. Oliva, P. McBurney and A. Omicini. Co-argumentation Artifact for Agent Societies.In Argumentation in Multi-Agent Systems Springer (2008). pp: 31-46.

(Omicini and Denti, 2001) A. Omicini and E. Denti: From tuple spaces to tuple centres. In Science of Computer Programming vol. 41(3), (2001). pp: 277–294.

(Ontañón and Plaza, 2006) S. Ontañón and E. Plaza. Arguments and Counterexamples in Case-Based Joint Deliberation. In Proc. 3rd Int. Workshop on Argumentation in Multiagent Systems (ArgMAS'06). Hakodate, Japan (2006). pp: 36-53.

(Optiz and Maclin, 1999) D. Opitz and R. Maclin. Popular Ensemble Methods: An Empirical Study, In Artif. Intell. Research, vol. 11, (1999). pp: 169-198.

(Oren et al., 2006) N. Oren, T. J. Norman and A. Preece. Loose Lips Sink Ships: a Heuristic for Argumentation. In Proc. 3rd Int. Workshop on Argumentation in Multiagent Systems (ArgMAS'06). Hakodate, Japan (2006). pp: 121 - 134.

(Oren et al., 2008) N. Oren, M. Luck, and T. J. Norman. Argumentation for normative reasoning. In Proc. Symp. Behaviour Regulation in Multi-Agent Systems, (2008). pp: 55-60.

(Parsons et al., 1998) S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. In Logic and Computation, vol. 8(3). Oxford University Press (1998). pp:261–292.

(Pei et al., 2000) J. Pei, R. Mao, K. HU and H. Zhu. Towards data mining benchmarking: a test bed for performance study of frequent pattern mining. In Proc. . In Proc. ACM SIGMOD Conf. on Management of Data (SIGMOD'00). Dallas, TX. ACM Press (2000). pp: 592.

(Pham et al, 2008) D. H. Pham, S. Thakur, and G. Governatori. Settling on the group's goals: An n-person argumentation game approach. In Proc. 11th Pacific Rim International Conference on Multi-Agents (PRIMA'08). Springer, (2008). pp: 328-339.

(Piatetsky-Shapiro, 2000) G. Piatetsky-Shapiro. Knowledge discovery in databases: 10 years after. In ACM SIGKDD Explorations, vol.1(2), (200). pp: 59 – 61.

(Pollock, 1995) J. Pollock. Cognitive Carpentry: A Blueprint for How to Build a Person. MIT Press, MA, USA, (1995).

(Prakken, 1993) H. Prakken. A logical framework for modelling legal argument. In Proc. 4th Int. Conf. on AI and Law (ICAIL'93). ACM Press, New Yotrk, (1993). pp: 1-10.

(Prakken, 2000) H.Prakken. On dialogue systems with speech acts, arguemnts and counter arguments. In Proc of the 7th Euro. Workshop on Logic for AI. Springer, Berlin (2000). pp: 224-238.

(Prakken, 2005a) H. Prakken. A study of accrual of arguments, with applications to evidential reasoning. In Proc. 8th Int. Conf. on AI and Law (ICAIL'05). Bologna, Italy. ACM Press, (2005). pp: 85-94.

(Prakken, 2005b) H. Prakken. Coherence and flexibility in dialogue games for argumentation. In Logic and Computation, vol. 15. Oxford University Press (2005). pp: 1009-1040.

(Prakken, 2006) H. Prakken. Formal systems for persuasion dialogue. In The Knowledge Engineering Review, vol. 21. Cambridge University Press(2006). pp: 163-188.

(Prakken, 2008) H. Prakken. Formalising ordinary legal disputes: a case study. In AI and Law, vol.16, (2008). pp: 333-359.

(Prakken and Sartor, 1997) H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. In Applied Non-Classical Logics, vol.7(1), (1997). pp: 25-75.

(Prakken and Sartor, 1998) H. Prakken and G. Sartor. Modelling reasoning with precedents in a formal dialogue game. In AI and Law, vol.6, (1998). pp:231–287.

(Prakken et al., 2005) H. Prakken, C. Reed and D. Walton. Dialogues about burden of proof. In Proc. 8th Int. Conf. on AI and Law (ICAIL'05). Bologna, Italy. ACM Press, (2005). pp: 115 – 124.

(Pollock, 1995) J. Pollock. Cognitive Carpentry: A Blueprint for How to Build a Person. MIT Press, MA, (1995).

(Quinlan, 1986) J. R. Quinlan. Induction of decision trees. In Machine Learning, Springer, Amesterdam, (1986). pp: 81-106.

(Quinlan, 1987) J. R. Quinlan. Simplifying decision trees. In. Man-Machine Studies, vol. 27(3). Academic Press (1987). pp: 221-234.

(Quinlan, 1989) J. R. Quinlan. Unknown Attribute Values in Induction. In Proc. 6th Int. Workshop on Machine Learning, (1989). pp:164–168.

(Quinlan, 1993) J. R. Quinlan. Combining instance-based and model-based learning. In Proc. 10th Int. Conf. on Machine Learning. Amherst, MA. Morgan Kaufmann, (1993). pp: 236-243.

(Quinlan, 1996) J. R. Quinlan. Bagging, Boosting, and C4.5. In Proc. 13th National Conference on Artif. Intell., AAAI Press, (1996). pp:725 – 730.

(Quinlan, 1998) J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, (1998).

(Quinlan and Cameron-Jones, 1993) J. R. Quinlan and R. M. Cameron-Jones. FOIL: A Midterm Report. In Proc. ECML'93. Austria, (1993). pp3-20.

(Rahwan et a., 2004) I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons, and L. Sonenberg. Argument-based negotiation. In Knowledge Engineering Review. Cambridge University Press (2004). pp: 343 - 375.

(Reed, 1998) C. Reed. Dialogue frames in agent communications. In Proc. Int. Conf. on Multiagent Systems (ICMAS'98). IEEE Press, (1998). pp: 246–253.

(Raymon, 1992) R. Raymon. Search through systematic set enumeration. In Proc. 3rd Int. Conf. on the Principles of Knowledge Representation and Reasoning. Cambridge, MA. (1992). pp: 539–550.

(Reichenbach , 1949) H. Reichenbach. The Theory of Probability. Berkeley: University of California Press, (1949).

(Rissland and Ashley, 2002) E.L Rissland and K.D. Ashley. A note on dimensions and factors. In AI and Law, vol. 10(1), Springer (2002). pp: 65-77.

(Rissland and Skalak, 1992) E. L. Rissland and D. B. Skalak. CABARET: Statutory Interpretation in a Hybrid Architecture. In Man-Machine Studies, vol.34. Academic Press (1991). pp: 839-887.

(Rissland et al., 1996) E.L. Rissland, D. B. Skalak, D. B and M. T. Friedman. BankXX: Supporting Legal Arguments through Heuristic Retrieval. In AI and Law, vol. 4(1), Springer (1996). pp: 1–71.

(Roddick and Rice, 2001) J. F. Roddick and S. P. Rice. What's interesting about cricket?—on thresholds and anticipation in discovered rules. In SIGKDD Explorations, vol. 3(1), (2001). pp: 1–5.

(Russell, 1974) B. Russell. On Induction. In R. Swinburne (ed.) The Justification of Induction. Oxford Readings in Philosophy series, Oxford University Press, (1974). pp: 19 – 25.

(Salzberg, 1997) S.L. Salzberg. On comparing classifiers: Pitfalls to avoid and a recommended approach. In Data Mining and Knowledge Discovery, vol.1. Springer (1997). pp: 317–328.

(Sarle, 1998) W. Sarle. Prediction with missing inputs. In Proc. 4th Joint Conf. on Information Sciences (JCIS'98), North Carloina, USA. (1998). pp: 399–402.

(Savage, 1954) L. J. Savage. The Foundations of Statistics. New York: Wiley, (1954).

(Savasere et al., 1995) A. Savasere, E. Omiecinski and S. Navathe. An Efficient Algorithm for Mining Association Rules in Large Databases. In Proc. 21st Int. Conf. on Very Large Data Bases (VLDB'95). Switzerland, (1995). pp: 432 – 444.

(Shcank, 1982) R. C. Schank. Dynamic Memory. A Theory of Reminding and Learning in Computers and People. New York: Cambridge University Press, (1982).

(Schapire, 1990) R. Schapire. The strength of weak learnability In Machine Learning, vol.5(2). Springer (1990). pp: 197-227.

(Sloman and Lagando, 2005) S.A. Sloman and D.A. Lagnado. The Problem of Induction. In Cambridge handbook of thinking and reasoning, Cambridge University Press (2005). p: 95.

(Suppes, 1998) P. Suppes. Review of Kevin Kelly, The Logic of Reliable Inquiry. In British Journal for the Philosophy of Science, vol. 49, (1998). pp: 351–354.

(Swinburne, 1974) R. Swinburne. Introduction. In R. Swinburne (ed.) The Justification of Induction. In Oxford Readings in Philosophy series, Oxford University Press, (1974). pp: 1 – 18.

(Tan and Dowe, 2003) P. Tan and D. Dowe. MML Inference of Decision Graphs with Multi-way Joins and Dynamic Attributes. In Proc. Australian Conference on Artificial Intelligence (2003). pp: 269 – 312.

(Toulmin, 1979) S Toulmin, An introduction to Reasoning. Macmillan Publishing C., Inc New York, (1979).

(Toivonen , 1996) H. Toivonen. Sampling large databases for association rules. In Proc. 22nd Int. Conf. on Very Large Data Bases (VLDB'96). Mumbai, India. Morgan Kaufmann (1996). pp: 134–141.

(Traum, 2004) D. R. Traum. Issues in multiparty dialogues. In Proc. 41st Annual Meeting on Association for Computational Linguistics (ACL'03). Melbourne, Australia, (2003). Springer, (2004). pp: 201–211.

(Vreeswijk and Prakken, 2000) G. A. W. Vreeswijk and H. Prakken. Credulous and sceptical argument games for preferred semantics. In Lecture Notes in Artif. Intell. vol.1919, Springer, Berlin (2000). pp: 239–253.

(Walton, 1985) D. N. Walton, Arguer's Position: A Pragmatic Study of Ad Hominem Attack, Criticism, Refutation, and Fallacy, contribution in philosophy, GreenWood Press, (1985).

(Walton, 1996) D. N. Walton. Argument Schemes for Presumptive Reasoning. Lawrence Erlbaum Associates, Mahwah, NJ, (1996).

(Walton, 1998) D.N. Walton. The New Dialectic, Toronto, University of Toronto Press, (1998).

(Walton, 2009) D. N. Walton. Invited Talk at 6th Int. Workshop on Argumentation in Multiagent Systems (ArgMAS'09). Budapest, Hungary, 12th May, (2009).

(Walton et al., 2008) D. Walton, C. Reed and F. Macagno. Argumentation Schemes. Cambridge University Press, (2008). pp: 43 -86.

(Walton and Macagno, 2009) D. Walton and F. Macagno. Reasoning from Classifications and Definitions. In Argumentation, vol.23(1), Springer (2009). pp:81-107.

(Walton and Krabbe, 1995) D. N. Walton and E. C. W. Krabbe. Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. SUNY Press, Albany, NY, (1995).

(Wardeh et al., 2009a) M. Wardeh, T. Bench-Capon and F.P. Coenen . PADUA: a protocol for argumentation dialogue using association rules. In AI and Law. Springer, vol. 17(3), (2009). pp: 183 – 215.

(Wardeh et al., 2009b) M. Wardeh, F.P. Coenen and T. Bench-Capon. An Arguing From Experience Approach to Classifying Noisy Data. In Proc. 11th Int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK'09). Springer LNCS 5691, pp: 354-365.

(Wardeh et al., 2009c) M. Wardeh, T. Bench-Capon and F.P. Coenen . *Multi-Party Argument from Experience*. To appear in Proc. 6th Int. Workshop on Argumentation in Multiagent Systems (ArgMAS'09). Budapest, Hungary (2009).

(Wardeh et al., 2008a) M. Wardeh, T. Bench-Capon and F.P. Coenen. *Arguments from Experience: The PADUA Protocol*. In Proc. 2nd Conf. on *Computational Models of Argument* (COMMA'08). Toulouse, France. IOS press, (2008). pp: 405- 416.

(Wardeh et al., 2008b) M. Wardeh, T. Bench-Capon and F.P. Coenen. *Argument Based Moderation of Benefit Assessment*. In Proc. 21st Annual Conf. on Legal Knowledge and Information Systems (JURIX'08). IOS Press, (2008). pp: 128-137.

(Wardeh et al., 2008c) M. Wardeh, T. Bench-Capon and F.P. Coenen. PISA - Pooling Information from Several Agents: Multiplayer Argumentation From Experience. In Proc. 28th SGAI Int. Conf. on AI (AI'08). Cambridge, UK. Springer, London, (2008). pp: 133-146.

(Wardeh et al., 2007a) M. Wardeh, T. Bench-Capon and F.P. Coenen. *Dynamic Rule Mining for Argumentation Based Systems.* In Proc. 27th SGAI Int. Conf. on AI (AI'07). Cambridge, UK. Springer, London, (2007).. pp: 65-78.

(Wardeh et al., 2007b) M. Wardeh, T. Bench-Capon and F.P. Coenen. PADUA Protocol: Strategies and Tactics. Proc. 9th Euro. Conf. on Symbolic and Quantative Approaches to Reasoning with Uncertainity (ECSQARU'07). Hammamet, Tunisia (2007). pp: 465-476.

(Webb, 2000) G. I. Webb. MultiBoosting: A Technique for Combining Boosting and Wagging. In Machine Learning. vol.40(2). Springer (2000). pp: 159 -196.

(Wellls and Reed, 2006) S. Wells, C. Reed. Knowing When To Bargain - The roles of negotiation and persuasion in dialogue. In Proc. 1st Conf. on Computational Models of Argument (COMMA '06). Liverpool, UK (2006). pp:235-246.

(Witten and Frank, 2005) I. H. Witten and E. Frank. Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, (2005).

(Wooldridge, 2001) M. Wooldridge. An Introduction to MultiAgent Systems. John Wiley and Sons, New York, NY, (2001).

(Yang et al., 1999) J. Yang, R. Parekh and V. Konava. DistAl: an inter-pattern distance-based constructive learning algorithm. In Intell. Data Anal, vol.3. IEEE(1999). pp:55-96.

(Yin and Han, 2003) X. Yin and J. Han. CPAR: Classification based on Predictive Association Rules. In Proc. SIAM Int. Conf. on Data Mining (SDM'03), San Francisco, CA, (2003). pp: 331-335.

(Yuan, T., 2004). Human Computer Debate, a Computational Dialectics Approach. Ph.D. Thesis, Leeds Metropolitan University, (2004).

(Yin and Han, 2003) X. Yin and J. Han. CPAR: Classification based on Predictive Association Rules. In Proc. SIAM Int. Conf. on Data Mining (SDM'03), San Francisco, CA, (2003). pp 331-335.

(Zaki , 2000) M.J. Zaki. Scalable algorithms for association mining. In IEEE Trans. Knowl. Data Eng. Vol 12(3), (2000). pp: 372–390.

(Zeleznikow and Stranieri, 1995) J. Zeleznikow and A. Stranieri. The Split-Up system. In Proc. 5th Int. Conf. on AI and Law (ICAIL'95). ACM Press, New York, (1995). pp: 185-195.

(Zhu and Wu, 2004) X. Zhu and X. Wu. Class noise vs. attribute noise: A quantitative study of their impacts. In Artif. Intell. Rev., vol. 22 (3/4), (2004). Pp: 177–210.

(Zhu and Wu, 2005) X. Zhu and X. Wu. Cost-constrained data acquisition for intelligent data preparation. In IEEE Trans. Knowl. Data Eng., vol. 17(11), (2005). pp: 1542– 1556.

# Appendix A: Design Documentation for the Java Implementation of PADUA

This appendix provides the design documentation for the implementation of the Java application that is based upon the PADUA Protocol (as described in Sub-section 4.1.1). Section A.1 discusses the analysis and design documentation for the implementation. Section A.2 provides the reader with a description of how to use the advocated application.

## A.1. PADUA Analysis and Design

Firstly, the analysis of the Java classes that are required to encode the PADUA protocol is discussed. These classes concern the basic operation of PADUA: implementing two-party "*Arguing from Experience*" dialogues. Then the details of how these classes were incorporated into the final PADUA GUI Application (Sub-section 4.1.1) are given. Of note here, the empirical analysis of Chapter 5 was undertaken using the basic PADUA classes (without the user interfaces). Figure A.1 presents a primitive class diagram showing the main classes that are needed for the dialogue game implementation. The code actually makes use of many more pre-defined classes from the Java Applications Programming Interface (API) but they have been omitted from this design documentation because although they are necessary for the program to function correctly, they are not the main focus point of the implementation presented here. The classes are all represented in the form of simplified UML style diagrams. In the given design class DialogueGame is assigned the operation of dialogue games between two players, each is an object of the class Player. Additionally it makes use of the class ActiveHistory to keep track of the ongoing dialogue. The majority of the other classes (other than Move, MoveNode and Strategy) are required for facilitating mining the requested association rules from given datasets. The original encoding of these classes was obtained from anonymous download from (Coenen, 2004b). However, the operation of the three argumentation queries (Section 3.3) was integrated into these classes.

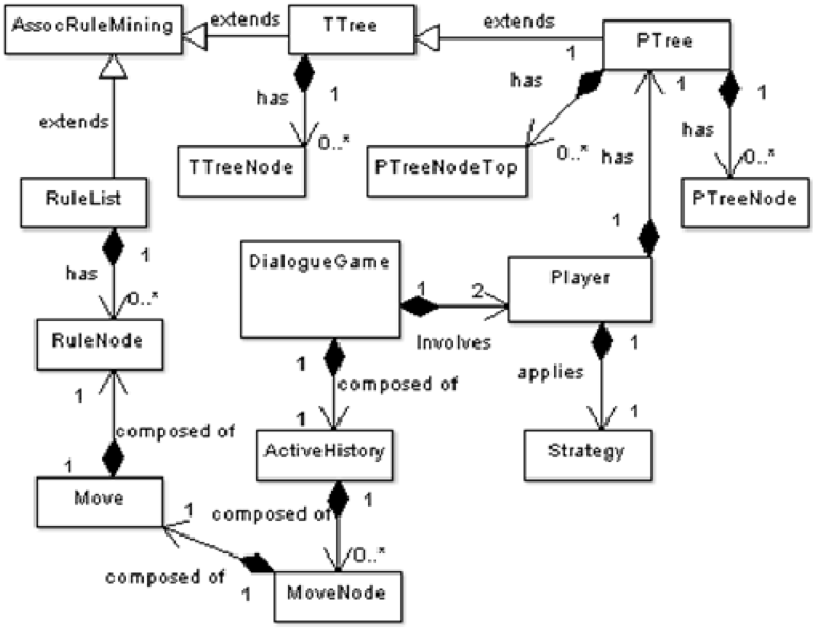


**Figure A.1. Primitive class diagram for PADUA.**

Given below in Figure A.2 is an analysis for the individual classes shown in Figure A.1. Here the attributes and the operations embodied in each class are highlighted. Only the essential visible operations of each class are given. Each of these classes requires a number of "*private*" operations to achieve their assigned design goals. However, these operations were not included in Figure A.2. For details about these operations please refer to the following webpage: www.csc.liv.ac.uk/~maya where a copy of the implementation classes can be downloaded. Additionally, where the constructor of a given class is default, the description of this constructor was omitted from the UML design in Figure A.2. The classes in Figure A.2 were implemented using Java, as intended, to operate the PADUA protocol along the outlines provided in Chapter 4. This implementation was tested thoroughly to ensure it does not contain any bugs. The above classes provide the basic operation of the PADUA protocol, on the assumption that all the input parameters are given. Chapter 4 presented the PADUA GUI Application as means to facilitate this input process, and to provide better means to visualise the resulting dialogues.

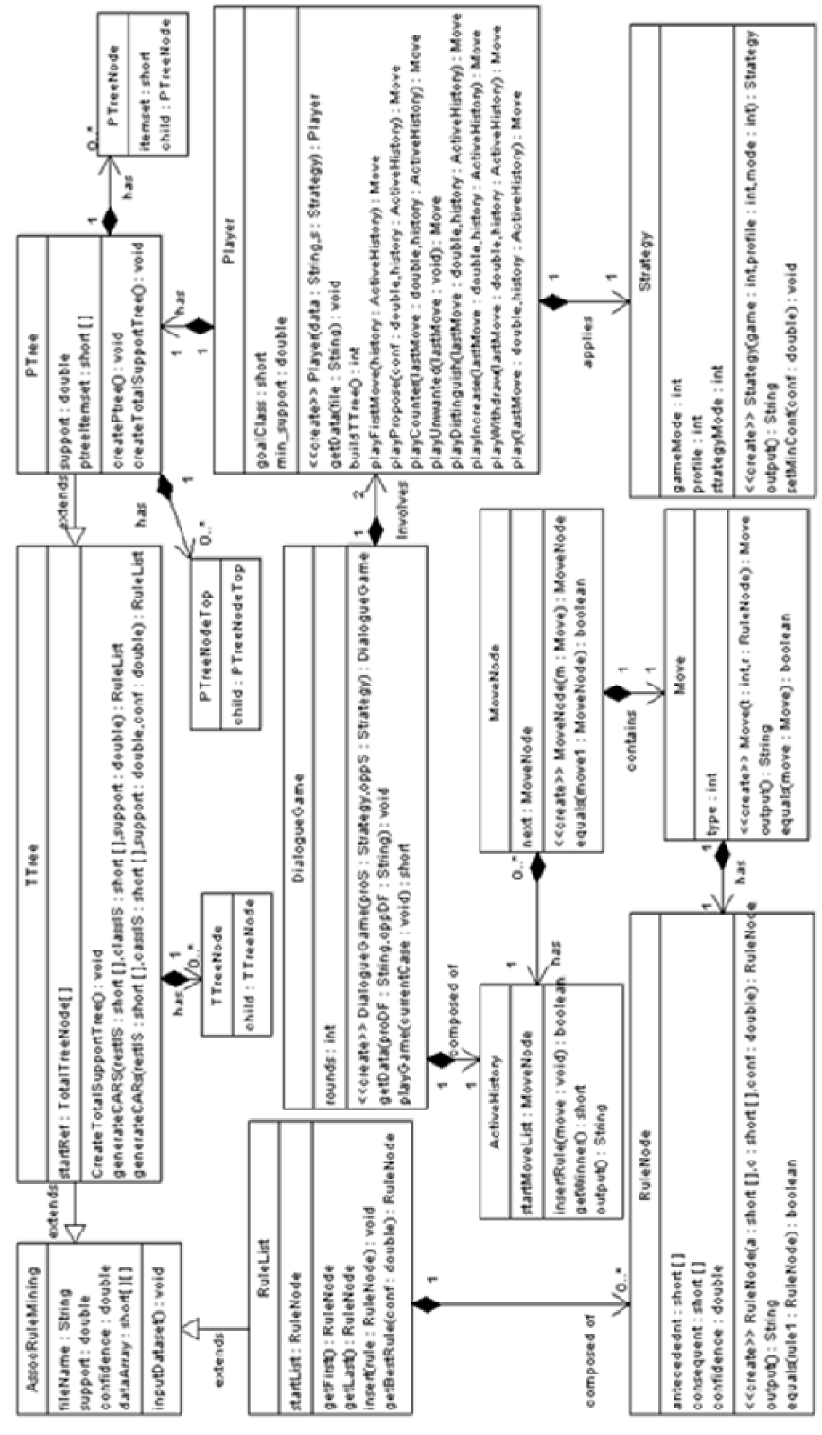

PTreeNode
itemset : short
child : PTreeNode
PTree
support : double
ptreeItemset : short [ ]
createPtree() : void
createTotalSupportTree() : void
Player
goalClass : short
min_support : double
<<create>> Player(data : String,s : Strategy) : Player
getData(file : String) : void
buildTTree() : int
playFirstMove(history : ActiveHistory) : Move
playPropose(conf : double,history : ActiveHistory) : Move
playCounter(lastMove : double,history : ActiveHistory) : Move
playUnwanted(lastMove : void) : Move
playDistinguish(lastMove : double,history : ActiveHistory) : Move
playIncrease(lastMove : double,history : ActiveHistory) : Move
playWithdraw(lastMove : double,history : ActiveHistory) : Move
play(lastMove : double,history : ActiveHistory) : Move
Strategy
gameMode : int
profile : int
strategyMode : int
<<create>> Strategy(game : int,profile : int,mode : int) : Strategy
output() : String
setMinConf(conf : double) : void
TTree
startRef : TotalTreeNode[ ]
CreateTotalSupportTree() : void
generateCARS(restIS : short [ ],classIS : short [ ],support : double) : RuleList
generateCARs(restIS : short [ ],classIS : short [ ],support : double,conf : double) : RuleList
PTreeNodeTop
child : PTreeNodeTop
TTreeNode
child : TTreeNode
DialogueGame
rounds : int
<<create>> DialogueGame(proS : Strategy,oppS : Strategy) : DialogueGame
getData(proDF : String,oppDF : String) : void
playGame(currentCase : void) : short
MoveNode
next : MoveNode
<<create>> MoveNode(m : Move) : MoveNode
equals(move1 : MoveNode) : boolean
Move
type : int
<<create>> Move(t : int,r : RuleNode) : Move
output() : String
equals(move : Move) : boolean
ActiveHistory
startMoveList : MoveNode
insertRule(move : void) : boolean
getWinner() : short
output() : String
RuleNode
antecedent : short [ ]
consequent : short [ ]
confidence : double
<<create>> RuleNode(a : short [ ],c : short [ ],conf : double) : RuleNode
output() : String
equals(rule1 : RuleNode) : boolean
RuleList
startList : RuleNode
getFirst() : RuleNode
getLast() : RuleNode
insert(rule : RuleNode) : void
getBestRule(conf : double) : RuleNode
AssocRuleMining
fileName : String
support : double
confidence : double
dataArray : short[ ][ ]
inputDataset() : void
has
has
has
has
has
has
extends
extends
extends
involves
applies
contains
composed of
composed of
1
0..*
2

**Figure A.2. Detailed UML class diagram for PADUA.**

The diagram in Figure A.3 exemplifies the design of the PADUA GUI Application. Please note that in this diagram, all the classes from the above UML diagrams, other than the DialogueGame class were omitted to simplify the diagram, as the GUI Application aims at providing this class with sufficient input to start the game, then displaying the results of this game. The diagram given in Figure A.3 aims at providing an insight to the design of the PADUA GUI Application. The following section will give the details of how to operate this GUI application to produce the intended dialogues.

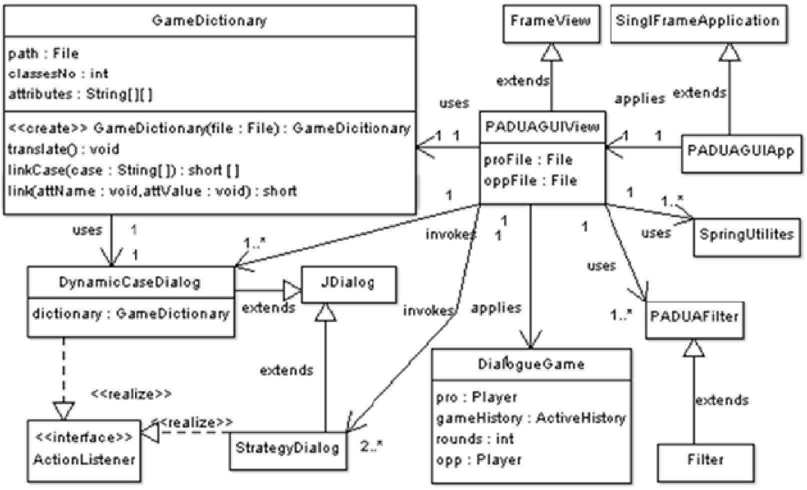
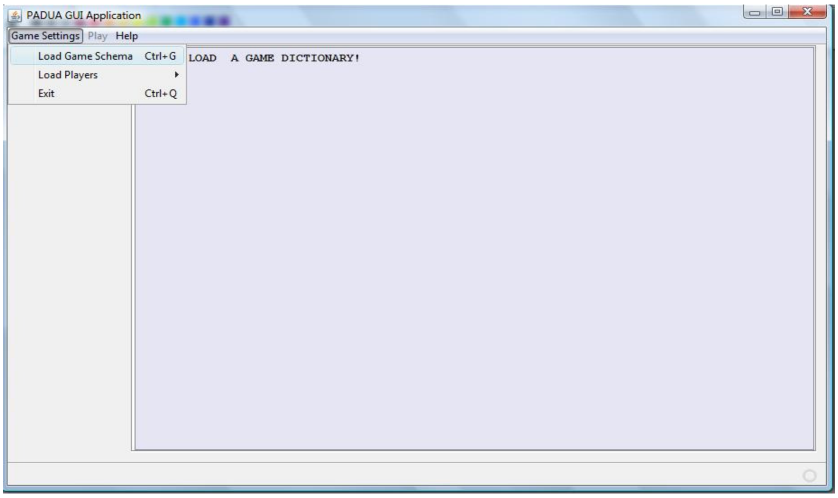


**Figure A.3. Class diagram of the PADUA GUI Application.**

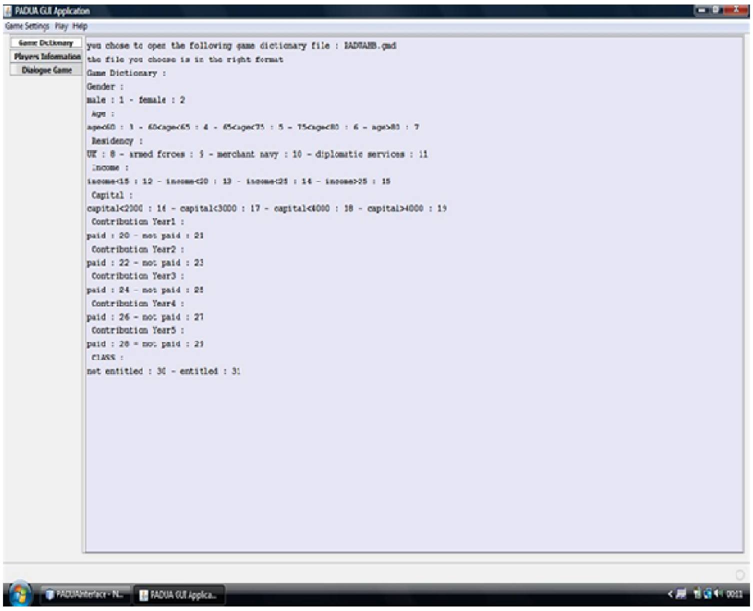
## A.2. Simple User Manual for the PADUA GUI Application

Sub-Section 4.1.2 presented an example produced using the advocated PADUA application. However, that example was intended to provide evidence as to how this application can be exploited to effectively construct meaningful dialogues, and to provide explanation of the underlying classification process of each input case. Given below in Figures A.4 – A.8 is a description of how to operate the given application, from starting it to producing dialogues. These figures are intended to explain the steps taken to produce the dialogue shown in Figure 4.2.



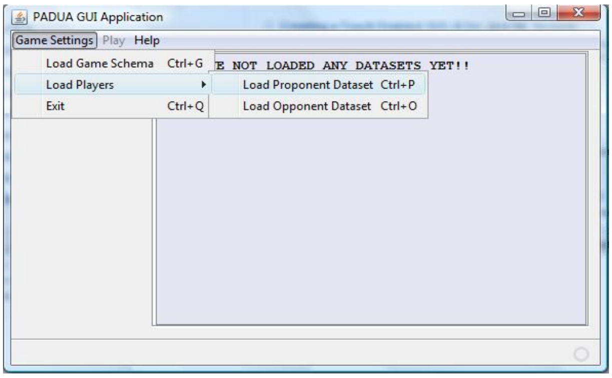
```
LOAD  A GAME DICTIONARY!
```

**Figure A.4. Introductory Screen to the PADUA GUI Application.**



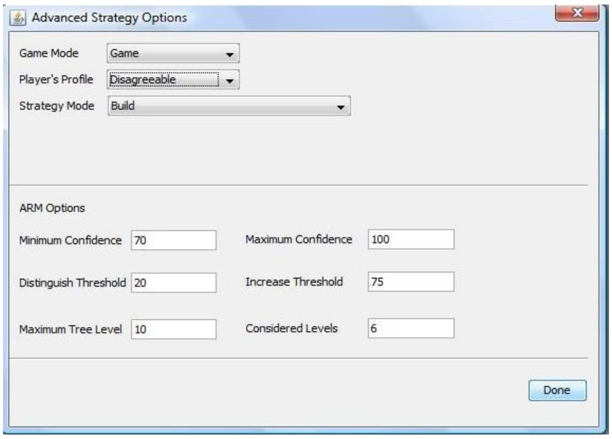
```
you chose to open the following game dictionary file : PADUAHB.gmd
the file you choose is in the right format
Game Dictionary :
Gender :
male : 1 - female : 2
 Age :
age<60 : 3 - 60<age<65 : 4 - 65<age<75 : 5 - 75<age<80 : 6 - age>80 : 7
 Residency :
UK : 8 - armed forces : 9 - merchant navy : 10 - diplomatic services : 11
 Income :
income<15 : 12 - income<20 : 13 - income<25 : 14 - income>25 : 15
 Capital :
capital<2000 : 16 - capital<3000 : 17 - capital<4000 : 18 - capital>4000 : 19
 Contribution Year1 :
paid : 20 - not paid : 21
 Contribution Year2 :
paid : 22 - not paid : 23
 Contribution Year3 :
paid : 24 - not paid : 25
 Contribution Year4 :
paid : 26 - not paid : 27
 Contribution Year5 :
paid : 28 - not paid : 29
 CLASS :
not entitled : 30 - entitled : 31
```

**Figure A.5. Results of successfully uploading a game dictionary.**



```
E NOT  LOADED  ANY  DATASETS  YET!!
```

**Figure A.6. Loading the players' datasets.**



Advanced Strategy Options

Game Mode: Game

Player's Profile: Disagreeable

Strategy Mode: Build

ARM Options

Minimum Confidence: 70

Maximum Confidence: 100

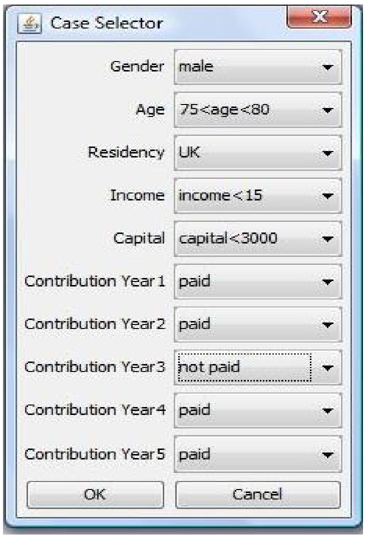
Distinguish Threshold: 20

Increase Threshold: 75

Maximum Tree Level: 10

Considered Levels: 6

**Figure A.7. Loading strategies.**



Case Selector

Gender: male

Age: 75<age<80

Residency: UK

Income: income<15

Capital: capital<3000

Contribution Year1: paid

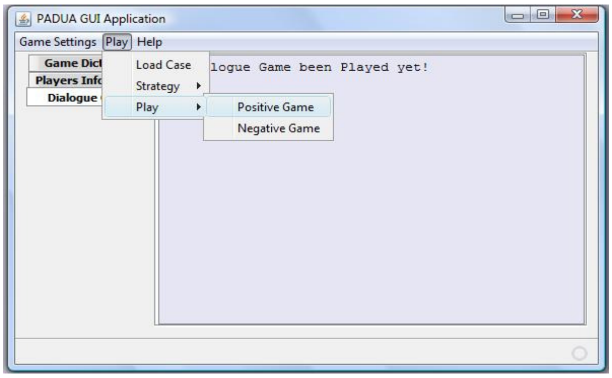
Contribution Year2: paid

Contribution Year3: not paid

Contribution Year4: paid

Contribution Year5: paid

**Figure A.8. Loading an input case.**



PADUA GUI Application

```
logue Game been Played yet!
```

**Figure A.9. Starting the dialogue.**

The above figures exemplify the steps required for the successful usage of the PADUA GUI Application. The first step is to upload a proper game dictionary file, containing the description of a given domain (dataset), to which the user intends to apply PADUA. The PADUA GUI Application assumes that the game dictionary files have the extension *.gmd*. Table A.1 presents schema for .gmd files. The second step is to load the data files (datasets) of each of the proponent and the opponent. After which a new case can be uploaded to the application using the case selector (Figure A.8). Should they wish, the users can manipulate the strategy of each of the two players using the strategy option dialog window (as shown in Figure A.7). Once all the game parameters are set, a new PADUA dialogue game can be instantiated between the proponent and the opponent. The given application provides two options, with regard to the dialogue games, positive and negative games – the first assumes that the proponent is an advocate of the positive classification (e.g. entitled) and the second assumes the opposite. However, in cases where there is no positive and negative classifications (e.g. white or red), the first options stands for the proponent defending the first class value (e.g. white), and the second stands for the opposite (proponent advocating the red class label).

| .GMD |
|---|
| Classes number |
| Attributes number |
| ATTRIBUTE |
| name |
| VALUES |
| value1/value2/.../value$_n$ |
| . |
| . |
| . |
| ATTRIBUTE |
| Class |
| VALUES |
| class$_1$/.../class$_c$ |

**Table A.1. Schema for gmd Files.**

# Appendix B: Design Documentation for the PISA Application

This appendix presents the design documentation upon which the Java implementation of the PISA Application (Section 6.5) is based. A discussion of the analysis and design steps for this application is given in Section B.1. Section B.2 explains the steps required, on the behalf of the user, to generate dialogues using the given application. In particular the process of generating the example given in Section 6.5.1 is exemplified.

## B.1. PISA Analysis and Design

This section provides some insight to the analysis step upon which the PISA Application was based. In particular, the JAVA classes that are required to encode this application are given. These classes embody the basic operation of the PISA application – allowing any number of participants to take part in "*Arguing from Experience*". Of note here, the empirical analysis of Chapter 8 was undertaken using another application, comprising the basic PISA classes, but does not implement any user interface; thus saving on the input time. Figure B.1 presents a primitive class diagram showing the main classes that are needed for implementing PISA. The given design assigns the operation of the system to the Chairperson class, which comprises of three units (components), each acting as a server for a number of tasks that would form one logical unit. The design promoted here structures the chairperson into three units:

- *Dialogue Management Unit (DMU)*: Manages and maintains the flow of PISA dialogues.
- *Participants Management Unit (PMU)*: Maintains a list of all the participants in a PISA dialogue game. PMU also updates this list throughout the dialogue and keeps track of the activities of each participant.

Additionally, this unit discards the participants who failed to contribute for a predefined number of rounds.

- *Argumentation Tree Management Unit (TMU)*: Examines the moves received from PMU, and decides which of these moves can be added to the Argumentation Tree and which should be discarded as illegal moves. TMU then adds the legitimate moves to the tree and adjust its colouring.

A few classes in this diagram are identical to ones given in Figure B.1. These classes concern the operation of mining adequate ARs from a given dataset.

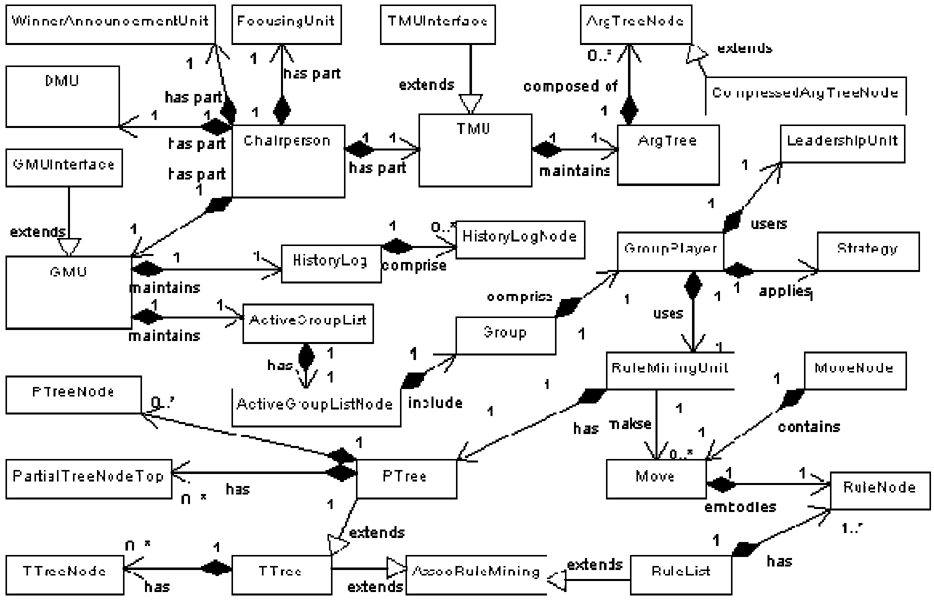


**Figure B.1. Primitive class diagram for PISA.**

The above class diagram is extended to provide details with respect to the attributes and operations embodied in each class. However, due to the complex nature of the promoted PISA Application a separate description is given for each of the chairperson and the three basic units – GMU, DMU and TMU. Figure B.2 illustrate the class diagrams of the chairperson agent. The classes described were implemented using Java, as intended, to operate PISA as outlined in Chapter 6 and 7. This implementation was tested thoroughly to ensure it is bug-free. The given classes provide the basic operation of PISA, on the assumption that all the input parameters are given.

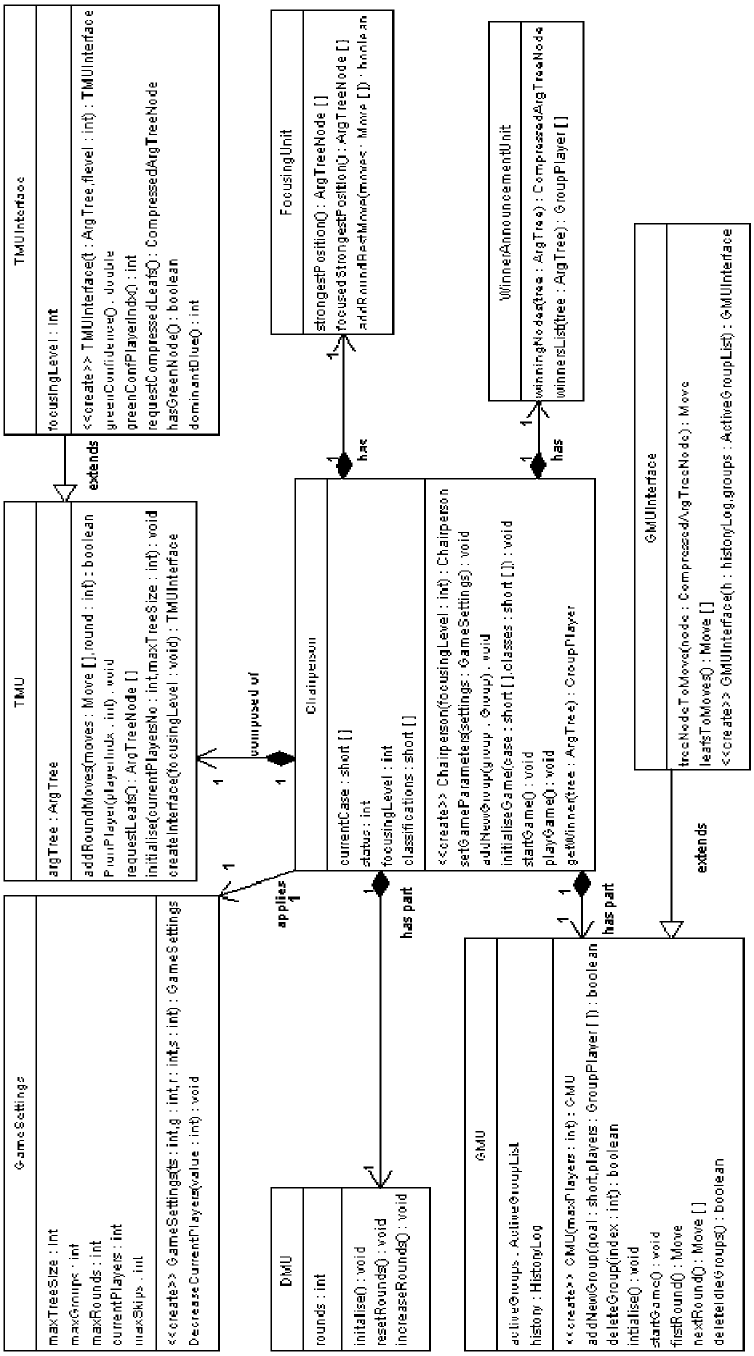

GameSettings
maxTreeSize : int
maxGroups : int
maxRounds : int
currentPlayers : int
maxSkips : int
<<create>> GameSettings(ts : int,g : int,r : int,s : int) : GameSettings
DecreaseCurrentPlayers(value : int) : void
TMU
argTree : ArgTree
addRoundMoves(moves : Move [],round : int) : boolean
PrunPlayer(playerIndx : int) : void
requestLeafs() : ArgTreeNode []
initialise(currentPlayersNo : int,maxTreeSize : int) : void
createInterface(focusingLevel : void) : TMUInterface
TMUInterface
focusingLevel : int
<<create>> TMUInterface(t : ArgTree,flevel : int) : TMUInterface
greenConfidence() : double
greenConfPlayerIndx() : int
requestCompressedLeafs() : CompressedArgTreeNode
hasGreenNode() : boolean
dominantDlue() : int
extends
1
composed of
1
applies
1
1
DMU
rounds : int
initalise() : void
resetRounds() : void
increaseRounds() : void
1
has part
1
Chairperson
currentCase : short []
status : int
focusingLevel : int
classifications : short []
<<create>> Chairperson(focusingLevel : int) : Chairperson
setGameParameters(settings : GameSettings) : void
addNewGroup(group : Group) : void
initialiseGame(case : short [],classes : short []) : void
startGame() : void
playGame() : void
getWinner(tree : ArgTree) : GroupPlayer
1
has
1
FocusingUnit
strongestPosition() : ArgTreeNode []
focusedStrongestPosition() : ArgTreeNode []
addRoundBestMove(moves : Move []) : boolean
1
has
1
WinnerAnnouncementUnit
winningNodes(tree : ArgTree) : CompressedArgTreeNode
winnersList(tree : ArgTree) : GroupPlayer []
GMU
activeGroups : ActiveGroupList
history : HistoryLog
<<create>> GMU(maxPlayers : int) : GMU
addNewGroup(goal : short,players : GroupPlayer [ ]) : boolean
deleteGroup(index : int) : boolean
intialise() : void
startGame() : void
firstRound() : Move
nextRound() : Move []
deleteIdleGroups() : boolean
1
1
has part
extends
GMUInterface
treeNodeToMove(node : CompressedArgTreeNode) : Move
leafsToMoves() : Move []
<<create>> GMUInterface(h : historyLog,groups : ActiveGroupList) : GMUInterface

**Figure B.2. Detailed class diagram for the chairperson and the associated classes.**

Chapter 6 presented a special GUI interface intended to provide means to facilitate this input process, on the behalf of the user. The given interface aims at providing the user with a variety of output, thus the operation of PISA could be assessed and investigated thoroughly. The diagram in Figure B.3 exemplifies the incorporation of the GUI interface within the PISA Application. Note that in this diagram, all the classes from the above UML diagram are omitted for reasons of space, only the Chairperson class remains in this diagram; as the GUI operation is intended to provide this class with sufficient input to start the dialogue. Additionally, the chairperson will provide the given GUI classes with sufficient information to produce the required output.

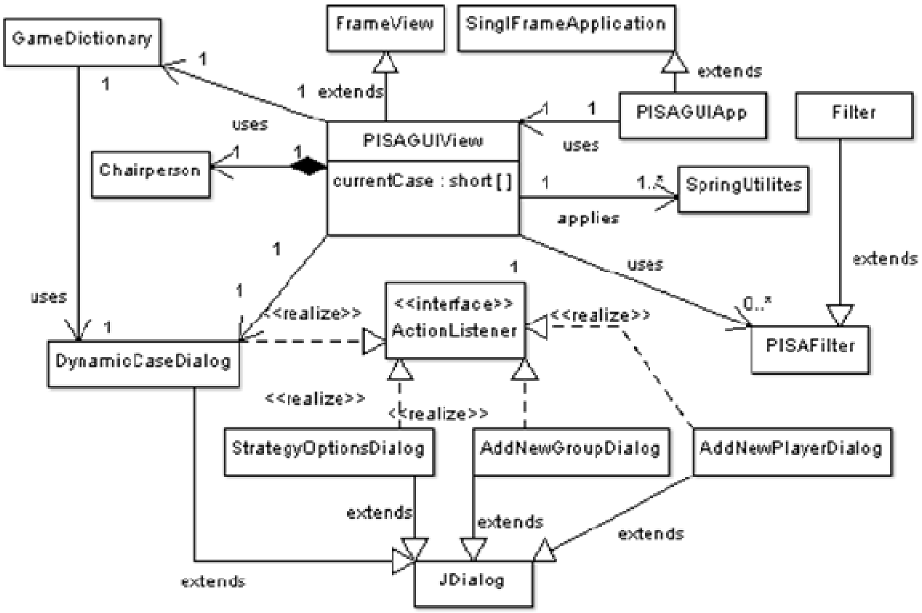
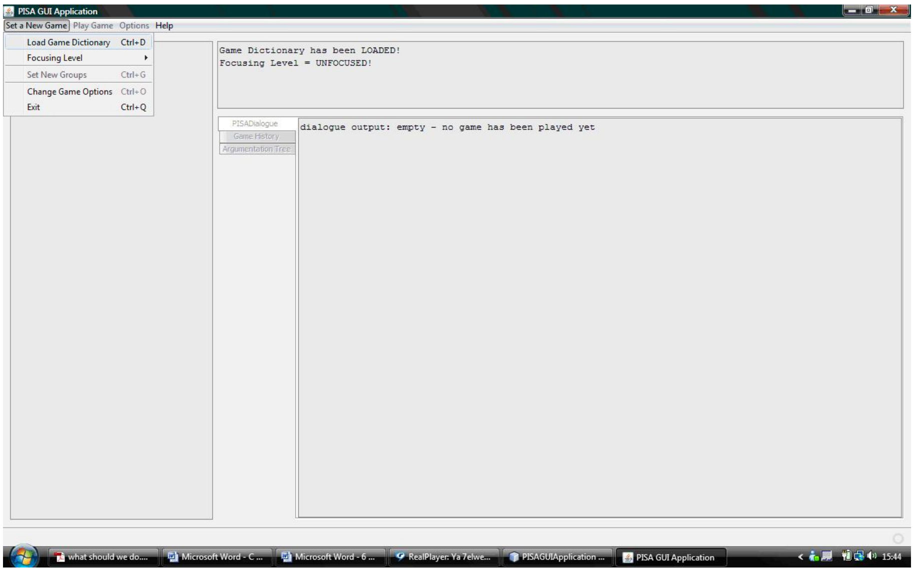


**Figure B.3. Class design of the PISA Application.**

## B.2. Simple User Manual for the PISA Application

Sub-Section 6.5.1 presented an example produced using the advocated PISA Application. However, that example was intended to provide an insight to the dialogues produced using PISA. Given below in Figures B.4 – B.7 is a description of how to operate the given application, from starting it to producing dialogues. These figures are intended to explain the steps that were taken to produce the dialogue shown in Figures 6.7, 6.8 and 6.9. These figures are intended to illustrate the essential steps required for the successful usage of the PISA Application. The first step is to upload an adequate game dictionary file

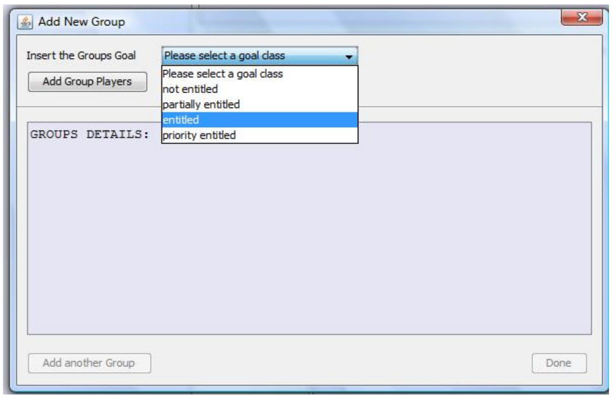
(Figure B.4). Once this file is uploaded the user has to create a number of groups equal to the number of classifications given in the game dictionary file, for each group the user is free to add as many individual player as she wishes (Figure B.5, B.6 and B.7). The user then can insert an input case and start the dialogue game, and then examine the resulting dialogue. The promoted application provides the user with additional output illustrating the Argumentation Tree (Figure 6.8) and the History Log (Figure 6.9) data structures. The operation of selecting a new input case is provided by a case selector identical to the one shown in Figure A.8.



```
Game Dictionary has been LOADED!
Focusing Level = UNFOCUSED!
```

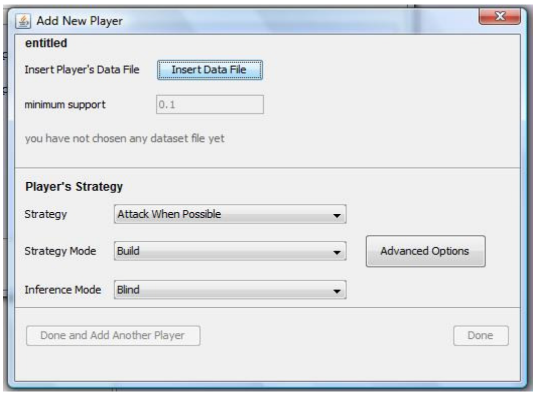
```
dialogue output: empty - no game has been played yet
```

Figure B.4. Introductory screen to PISA.



```
GROUPS DETAILS:
```

Figure B.5. Create a new group.



entitled

Insert Player's Data File

minimum support: 0.1

you have not chosen any dataset file yet

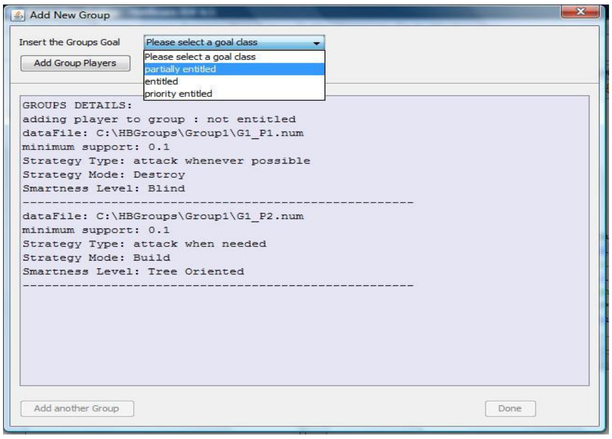
Player's Strategy

Strategy: Attack When Possible

Strategy Mode: Build

Inference Mode: Blind

**Figure B.6. Add a new player (in PISA).**

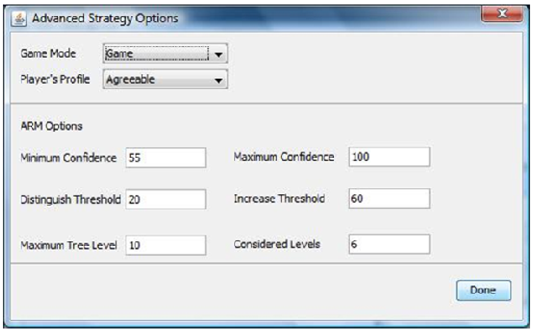


Insert the Groups Goal: Please select a goal class

```
GROUPS DETAILS:
adding player to group : not entitled
dataFile: C:\HBGroups\Group1\G1_P1.num
minimum support: 0.1
Strategy Type: attack whenever possible
Strategy Mode: Destroy
Smartness Level: Blind
-----------------------------------------------------
dataFile: C:\HBGroups\Group1\G1_P2.num
minimum support: 0.1
Strategy Type: attack when needed
Strategy Mode: Build
Smartness Level: Tree Oriented
-----------------------------------------------------
```

**Figure B.7. The group information display at the formation level.**

Having described the basic operation of PISA, the additional features of this application can now be discussed: Figure B.80 exemplifies the strategy selection process, by which the optional strategy parameters are and how the user may change the game setting parameters.



Game Mode: Game

Player's Profile: Agreeable

ARM Options

Minimum Confidence: 55

Maximum Confidence: 100

Distinguish Threshold: 20

Increase Threshold: 60

Maximum Tree Level: 10

Considered Levels: 6

**Figure B.8. Advanced Strategy Options.**

# Appendix C: Extending the Role of the Chairperson.

As described in the body of the thesis, the role of the chairperson agent has been that of a mediator administering the flow of the moves from the participants to the argumentation tree, and monitoring the behaviour of each participant. Thus, the chairperson does not influence the way participants direct their chosen moves. The participants are free to act in the way they find most suitable to realise their own strategies. However, the proposed turn taking policy (Section 6.2.2) lays the burden of maintaining the focus of the game on the participants themselves: they have to show higher levels of strategic planning to keep the dialogue on track, such that they do not place irrelevant moves against random opponents. Moreover, the larger the number of agents taking part in a given PISA dialogue, the larger the size of the auxiliary data structures, such as the argumentation tree and the history log, required to facilitate the dialogue. Thus, the cost of supervising these structures, by the chairperson, also gets larger. Also, the cost of operating these structures increases with the increase in the number of participants. All of this may reflect adversely on the performance of the PISA Framework. The above problems can be solved by imposing some restrictions on the moves that can be played in each round. This is achieved by giving the chairperson the authority to restrain the moves allowed throughout the dialogue game through the notion of levels of dialogue.

Three levels of PISA dialogues are now introduced, each level allowing for lesser degree of participants' freedom than the previous levels:

- **Unfocused Dialogues (Level 1)** are the ones which have been discussed at length in the body of the thesis. In these dialogues the participant can play any *legal* move.
- **Focused Dialogues (Level 2)**: The chairperson forces the participants to focus their attacks, so that in each round, the attacks are made against the participants holding the strongest position(s) of the previous rounds.

- **Strictly Focused Dialogues (Level 3)**: The chairperson practices even more control over the participants, allowing them to attack one and only one of the previously undefeated moves from the strongest position.

## C.1. Level 2: Focused Dialogues

At Level 2 participants are forced, in each round, to focus their attacks against the strongest position that has emerged from the previous rounds. Such enforcement means that at the beginning of each round, the chairperson informs the participants of the best position at this round. The permission to play in this round is given only to those participants who can, while any other move is considered illegal in this type of dialogue. The strongest position represented in a PISA argumentation tree, prior to the start of a round R is defined as follows:

- If the tree contains at least one green leaf node then the strongest position comprises the green nodes sharing the highest confidence (i.e. share the value of green confidence).
- If there are no green leaf nodes then the strongest position comprises the blue leaf nodes belonging to the player(s) owning the largest number of blue nodes in the current tree.

By focusing their attacks on the strongest position, the participants will effectively join forces against the strongest link(s) amongst the moves (propositions) made so far in the dialogue game. The strongest position changes in every round. Hence, the focus of the participants also changes. Eventually the participant(s) which has successfully defended itself against this type of joint attacks wins the game. Of course, focused dialogues can still end with a draw (tie), if two or more participants share the strongest position at the end of the game. In the case of two or more leaf nodes (previous moves) share the strongest position, the players can choose either one to attack. In the worst case scenario, if all the participants engaging in every round have played moves with similar strength, the focus of the game will be no better than the unfocused dialogues (Level 1).

Focusing the dialogues in the above manner requires certain changes to the control layer from the previous chapter. In particular, the chairperson is now assigned the additional task of focusing the participants' attacks in each round R against the strongest position computed prior to the start of that round $SP_R$. The chairperson, in the advocated PISA model, handles this task by firstly calculating $SP_R$ then informing the participants of this position, following the two rules discussed above. Upon receiving the move from participants, the chairperson filters any move that is directed against any leaf node outside the $SP_R$, and updates the dialogue game records accordingly. The remaining moves are added to the argumentation tree in the manner discussed in the previous chapter.

## C.2. Level 3: Strictly Focused Dialogues

This level avoids the emergence of more than one node in the strongest position. This is achieved as follows: Upon calculating the strongest position $\mathrm{SP_R}$ if this position consists of two or more leaf nodes from the argumentation tree, then the chairperson will choose one of these nodes, according to some criteria, and require the participants to attack this selected argument. By maintaining the focus of the dialogue game in this manner, the chairperson avoids the shortcomings of both focused and unfocused dialogues, in situations where the focus of the game gets diffused. In order to select one and only one of the previously undefeated moves (arguments) to be the focus of a new round R, the strongest position $\mathrm{SP_R}$ is processed according to some criterion. The criterion suggested here is referred to as *Random Strictly Focused dialogues*. Here the chairperson randomly chooses one argument from $\mathrm{SP_R}$ and demands that the participants should attack this argument in round R. The random criteria guarantees that the selection process is not be biased toward any participant. Note that this type of dialogue maintains the focus of PISA participants, in each round, against one strong argument. If this particular argument could not be defeated in one round, the chairperson attempts to keep the focus against this argument for a second round. Now, if none of the participants takes part in this second round, the chairperson will not terminate the dialogue. Rather, the chairperson will choose another argument from the strongest position and ask the participants to try and attack the new argument. The chairperson terminates the game only if all the arguments in the strongest position remained undefeated.

## C.3. Discussion of the three levels in PISA

Each level of the PISA framework discussed above, simplifies the argumentation tree, and speeds up the dialogue by focusing it (level 2) or further restricting it (level 3). However, levels 2 and 3 demands the chairperson to perform additional calculations at the beginning of every round in the dialogue; thus a bottleneck may emerge from these additional calculations as participants have to wait for the chairperson to inform them, at the beginning of every round, of the strongest position (level 2) or the strongest argument (level 3). Level 1 has no such bottleneck as the calculations are distributed amongst the participants, and not centralised within the chairperson.

The application of the above levels requires a delicate balance between the role of the chairperson and the role of the participants taking part in the dialogue. On one hand participants applying simple strategies may lead to unfocused dialogue games, consequently the resulting dialogues may be lengthy and/or include irrelevant arguments (moves) directed against random opponents. On the other hand, focused and strictly focused levels lead to dialogues where the new moves are directed only against the strongest arguments from previous rounds with the additional cost laid on the chairperson. This cost depends on two factors: (i) the strategies of the participants taking part in the game, and (ii) the number of these participants. The more participants there are in a game, the higher this cost. Additionally, if all (or most of) the participants adopt smart strategies (or at least strategies that are better than blind ones), then the chairperson will have to perform fewer calculations, as the participants would have already done some of these calculations on their own.

## C.4. An Empirical Study of the Effect of the Focusing Levels on PISA

This section presents the result of an empirical test intended to investigate the effects of allocating more control in the hands of the chairperson on both the operation of PISA and the characteristics of the produced dialogues. To address these issues two TCV tests were carried out using a 4000 records Housing Benefit (4 Classes) dataset, like the one used in testing the effects of the dataset size on the performance of PISA. In the first test, PISA was applied using *Focused* dialogues (level 2) and in the second using *Strictly Focused* dialogues (level 3). The results were then compared to the ones achieved without any focusing, as reported in Sub-section 8.2.4. Figure C.1 illustrates these results.

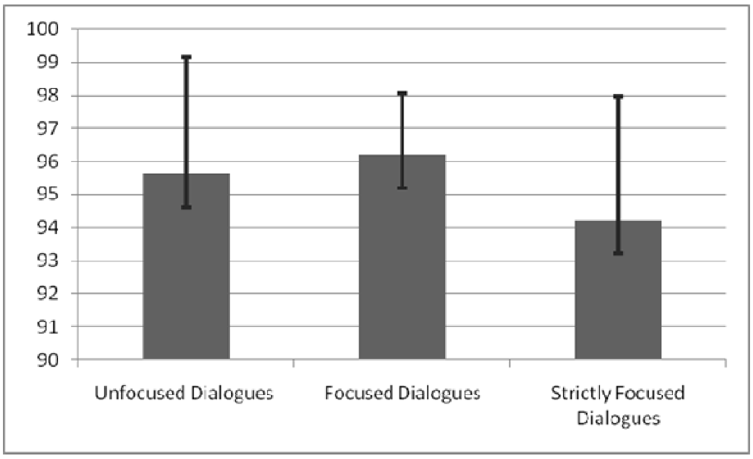


**Figure C.1. The results of the TCV test using different levels of focusing.**

From Figure C.1, a number of observations can be made:

- The best accuracy (96.2%) was achieved using focused dialogues in which the chairperson obliged the participants to attack, in each round, the strongest position(s) on the argumentation tree, rather than freely attack any undefeated previous moves (presented by the tree leaf nodes). This increase in accuracy, when compared with Unfocused dialogues (95.6%), was due to the fact that this type of dialogue did not allow for any unnecessary moves that did not benefit the overall goal of the dialogue, and thus incorrect classifications were less likely to emerge from situations where one participant agent wins the dialogue game simply because this agent has placed many distinguishing (or other blue) attacks against marginal targets. Focused dialogues also produce the most consistent performance across the ten folds.

- The worst accuracy occurred when applying strictly focused dialogues (94.2%), as here participants had to focus their attacks against one and only one previous move in each round. Thus, other moves with similar strength would be left unattacked, resulting in a drop in the accuracy of the classifications produced under this level of focusing.
- The implications of applying different focusing levels do not only affect the accuracy of the resulting classifications, but they also touch upon other dialogue features; especially: The length of the dialogues and the type of end-results.

With respect to the length of the produced dialogues, the shortest dialogues, as expected, were generated using the strictly focused level, with an average number of rounds equal to 4.97 (with 3.12 standard deviation). The focused level result in (on average) 5.78 rounds per dialogue (with 4.19 standard deviation), while the unfocused level produced the longest dialogues (6.37 rounds on average (with 4.73 standard deviation)). As for the end results, Table C.1 shows the average number of green wins/strong ties, and blue wins/weak ties of each level. Note that the highest percentage of Green wins was scored using the focused level. Additionally, the lowest percentage of ties was reported when applying focused games. However, strictly focused games led to sharp increase (almost double) in the percentage of ties amongst the participant, when compared with focused games. This was for the same reason as caused strictly focused dialogues to result in lower accuracy than focused and unfocused games. Interestingly, unfocused games produce the greatest number of blue wins, as here participants may attack freely, and thus apply more blue attacks (i.e. attacks that do not result in a new AR) than when they are forced to focus their attacks against the strongest positions.

| **Focusing Level** | **Green Wins** | **Blue Wins** | **Strong Ties** | **Weak Ties** |
|---|---|---|---|---|
| **Unfocused** | 270.6 (67.65%) | 121.5 (30.38%) | 3.9 (0.98%) | 4 (1%) |
| **Focused** | 291.2 (72.8%) | 102.9 (25.73%) | 2.8 (0.7%) | 3.1 (0.78%) |
| **Strictly Focused** | 290.4 (72.6%) | 98.1 (24.53%) | 5.4 (1.35%) | 6.1 (1.53%) |

**Table C.1. The average end results produced under each focusing level (/ 400).**

To sum up, the three different levels of focusing lead to different styles of dialogues which vary in: the accuracy of their classifications their end-results and their lengths. Focused dialogues seem to produce better results than the other two levels, whereas strictly focused ones produced the shortest dialogue.

# Appendix D: Applying PISA to Misinterpreted Data

Recall from Chapter 5 that PADUA was shown to perform well with the presence of systematic noise, a detailed account of which was given in Section 5.4 PISA was also subjected to similar tests using misinterpretations such that its robustness to "*systematic errors*" in input data could be evaluated. The results are reported in this appendix. The remainder of this appendix will address the misinterpretation problem associated with the process of awarding Retired Persons Housing Allowance (RPHA) benefits in multi-class settings[1]. It is assumed that the Housing Benefits applications are assessed in a number of different offices located in different areas, thus each can be subject to different regional differences arising from misinterpreting one of the conditions associated with the RPHA decision making process (Section 6.5.1). Multiparty "*Arguing from Experience*" dialogues were enforced between a number of participants, each representing one office, for the purposes of classifying given cases, such that each party in disagreement may argue for their positions with the other parties, and by exploiting the differences in their experiences, and in their decision making processes, the right conclusion may emerge.

Three experiments were performed in order to examine when the disagreement between the participants of these dialogues is a result of each party misinterpreting the input data. Here, PISA was applied to the fictional multi-class RPHA process described above. It was assumed that the RPHA benefit was assessed in four different offices, each located in different regional areas. These experiments are described as follows:

- The first evaluated the extent to which classification would be improved by moderation using PISA. This was done using a TCV test. A number of other classifiers were also applied to the data to provide a comparison.

[1] The outlines of this process were previously given in Sections 5.5, 5.1 and Sub-section 6.5.1.

- In the second McNemar's tests were performed to show the significance of the differences between classifiers.
- A more detailed analysis of the performance of PISA was also carried out in order to discover interesting properties of the underlying dialogues.

For the purposes of these experiments, four large datasets were generated such that one dataset (DS1) comprised 10,000 rows of correctly interrelated RPHA records (2500 rows for each class). The other three datasets, each also containing 10000 records, included 25% (2500) misinterpreted records, and the remaining records were equally distributed among the four possible classifications. These datasets are described as follows:

- **DS2**: contained 2500 records which were misinterpreted such that female aged between 60 and 64 were considered illegible to benefit.
- **DS3**: contained 2500 records which were misinterpreted such that candidates with capital larger than 2000£ were considered illegible to full benefit, rather to partial benefit only.
- **DS4**: contained 2500 records which were misinterpreted such that armed forces candidates who paid contribution in 4 out of the last 5 years were considered illegible to priority benefit (instead of normal benefit).

## D.1. Cross Validation using Misinterpreted Data

In the first experiment, eight other classifiers (TFPC, CBA, CMAR, RDT, IGDT, FOIL, PRM and CPAR) were used, operating on the union of the four datasets described above (DS = DS1 $\cup$DS2$\cup$DS3$\cup$DS4), and a TCV test was performed using each of these classifiers, and PISA. For PISA, each of the above datasets was given to one PISA Participant Agent. For each run of the TCV, one tenth of the dataset was set aside, and PISA was applied to classify a test set comprises of 3000 cases divided as follows: 600 cases for each of the four classes (2400 in total)[2] in addition to:

- 200 cases that should be classified as entitled but players with age misinterpreted data may classify it as not entitled.
- 200 cases that should be classified as entitled but players with capital misinterpreted data may classify it as partially entitled.
- 200 cases that should be classified as entitled but players with contribution misinterpreted data may classify it as priority entitled.

For the other classifiers, randomly selected 10% of DS was excluded from each run of the TCV, and the classifiers were applied to the same test set described above. Additionally, for PISA, two runs were performed: The first run comprised four players each making use of one of the four datasets described above. In the second run, each dataset was further divided into four equal subsets, and then PISA was applied with four groups, each comprising four players. Figure D.1 presents the results of these TCV tests. From this figure, it is clear that the CAR algorithms (TFPC, CBA, and CMAR) performed less well than PISA and the two decision trees methods, which performed significantly better. PISA also outperformed all the other classifiers, scoring 93.6% on average when using four individual players and 97.2% when using four groups of four players. Note that the decision tree classifiers seem to perform consistently throughout the test, whereas PISA showed more variation in its performance, in particular when using groups. This suggests that when dividing the amount of information available to one group amongst its members, the operation of the group becomes more sensitive towards the exact sample available to each of its individual members. This is because each member has to mine the required rules (arguments) from its own dataset, which in this particular test equalled a quarter the size of the one available to individual players when applying PISA without any groups. However, the application of groups seemed to benefit the overall operation of PISA. This is considered to be

[2] The 600 cases failing to entitle to any benefit were distributed equally amongst the 5 conditions (thus each 120 fail to meet one condition. Same applies for the partial benefit cases (200 for each condition).

because the greater the number of separate datasets available for groups members the greater the number of arguments found, thus enabling a more thorough exploration of the problem.

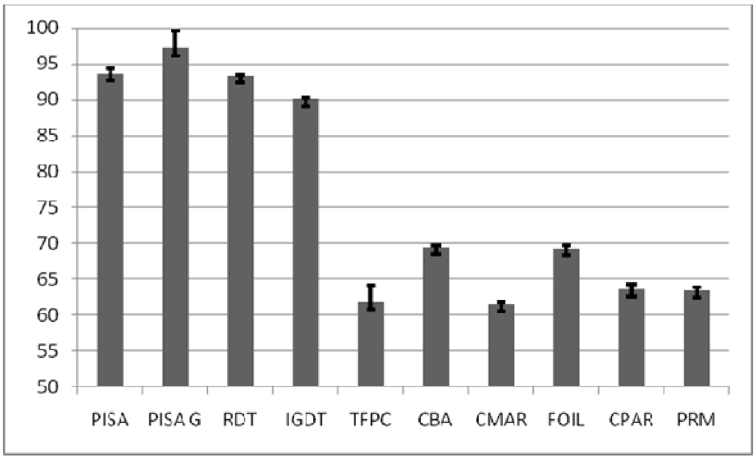


**Figure D.1. The results of performing cross-validation test using data with systematic errors.** ***Error bars represent the standard deviation for each classifier.***

## D.2. The McNemar's Test using Misinterpreted Data

Overall, PISA outperformed all the other classifiers when classifying misinterpreted RPHA records and this level of performance is encouraging and merits further analysis as to why PISA copes with systematic noise. For this purpose a McNemar's test was also carried out to investigate whether PISA is better or worse than any of the other classifiers used in the previous test. For this test PISA operated using a newly generated set of test cases comprising of 700 cases (100 cases of each class and 100 cases that could be wrongly decided for each of the misinterpretations described above).This data was then used as a test set for the other classifiers, the original data supplying the training set. As might be anticipated from interpreting Figure D.2, PISA (with/without groups) significantly outperformed the CARM classifiers. Additionally, there were no significant differences in the behaviour of PISA compared with the two decision trees classifiers.

Table D.1 illustrates these results. As part of the McNemar's test detailed information was generated as to which cases were misclassified by one or both of the classifiers under consideration. The comparison between PISA and the two decision tree classifiers is presented in Table D.2.

| **Test** | | **CBA** | **CMAR** | **TFPC** | **RDT** | **IGDT** | **FOIL** | **CPAR** | **PRM** |
|---|---|---|---|---|---|---|---|---|---|
| **No Group** | **M** | 216 | 137.15 | 354.06 | 1.4 | 0 | 175 | 141.13 | 175 |
| | **P** | **<0.0001** | **<0.0001** | **<0.0001** | 0.31 | 0.62 | **<0.0001** | **<0.0001** | **<0.0001** |
| **Group** | **M** | 259.68 | 175.97 | 381.82 | 1.72 | 0.04 | 202.3 | 168.2 | 202.3 |
| | **P** | **<0.0001** | **<0.001** | **<0.0001** | 0.24 | 1 | **<0.0001** | **<0.0001** | **<0.0001** |

**Table D.1. McNemar's and P-Value for systematic noise test using PISA.**

| **Dataset** | **No Groups** | **Groups** |
|---|---|---|
| **Both Failed** | 14 | 2 |
| **PISA Failed** | 14 | 19 |
| **PADUA Failed** | 21 | 28 |
| **Both Succeeded** | 651 | 651 |

**Table D.2. McNemar's Tests results for the Systematic Noise Experiment.**

## D.3. Further Discussion

Here, the TCV trials for PISA (with and without groups) are considered in more detail. Figure D.2 illustrates the detailed results of this test. From this figure it can be observed that, overall, PISA performs well in classifying the input cases. However, some types of cases seem to be easier to classify than others. For instance, PISA (without groups) scores high accuracy when dealing with cases that should entitle "*partial benefits*". One the other hand, the same application seems to "*suffer*" when classifying cases failing to meet one of the required conditions. This issue arises from the high number of misinterpreted records (25% of the data size) which impairs the ability to form correct rules. Therefore, PISA rarely scores 100% accuracy with any set of cases other than applicants who should be given partial benefits due to not meeting the capital condition ("*partial benefits*" class). However, when using groups, the overall behaviour seems to become more consistent. The previous discussion shows that PISA provides an approach to the problem of systematic errors. The experimental results reported here have demonstrated that multiparty "*Arguing from Experience*" dialogues result in reducing the number of misclassifications when using datasets with such errors.

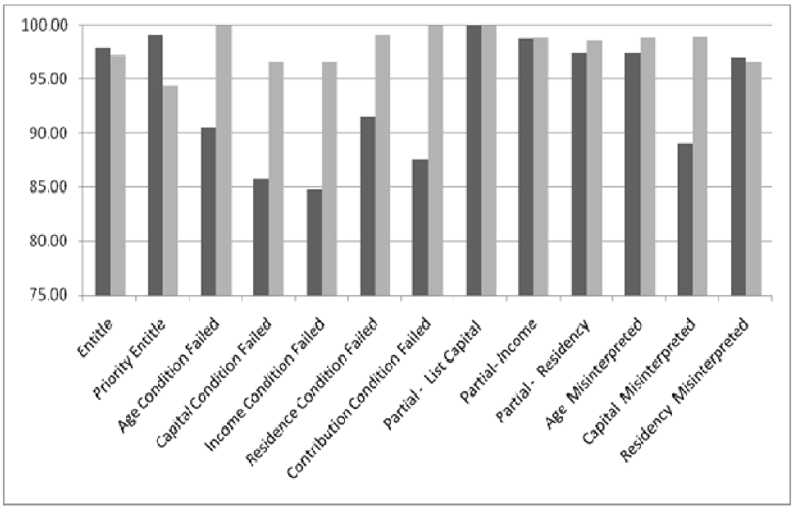


**Figure D.2. The results of the TCV tests for PISA with systematic noise.** *Dark grey columns represent the results obtained from PISA without groups and the light ones are for PISA with groups.*