PADUA: A PROTOCOL FOR ARGUMENTATION DIALOGUE USING ASSOCIATION RULES[1]

Abstract: We describe PADUA, a protocol designed to support two agents debating a classification by offering arguments based on association rules mined from individual datasets. We motivate the style of argumentation supported by PADUA, and describe the protocol. We discuss the strategies and tactics that can be employed by agents participating in a PADUA dialogue. PADUA is applied to a typical problem in the classification of routine claims for a hypothetical welfare benefit. We particularly address the problems that arise from the extensive number of misclassified examples typically found in such domains, where the high error rate is a widely recognised problem. We give examples of the use of PADUA in this domain, and explore in particular the effect of intermediate predicates. We have also done a large scale evaluation designed to test the effectiveness of using PADUA to detect misclassified examples, and to provide a comparison with other classification systems.

Keywords*: Argumentation Dialogue, Dialogue Games, Classification, Association Rules.*

1. Introduction

In this paper we will describe PADUA (*Protocol for Argumentation Dialogue Using Association Rules*), a system designed to enable two agents to engage in a persuasion dialogue regarding the classification of a new example, and its application to examples in the legal domain. PADUA models *argument from experience*: that is the agents will have considerable experience of classifying examples in the particular domain and will draw on this experience to offer reasons for classifying the new example. For our legal example we will consider the type of lay adjudication found in many routine cases, such as the award of welfare benefits in the UK. Such adjudicators will deal with many cases, and will develop particular habits of classification. Error rates in such decision making are high, and this is often because rarely encountered exceptions are overlooked, and because some bad habits can become ingrained. Such benefits are typically decided by a range of adjudicators working in several different offices, and different adjudicators and different offices will tend to encounter different types of case (e.g. some particular lung diseases are much more common in former mining areas, some occupations requiring special treatment, such as fishermen, will be rarely encountered in inland areas, etc) and so tend to develop different bad habits and blind spots. PADUA is intended to ameliorate this by allowing a dialogue between two agents representing different offices, with a view to moderating their decisions. Our idea is that this will enable mistakes to be corrected.

The high error rate encountered in the assessment of claims to welfare benefit is a significant problem, certainly not well addressed by current procedures. Groothuis and Svensson [1] drew attention to this 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 of US Disability Insurance [2]:

> ‘‘The multistage process for determining eligibility for Social Security Disability Insurance (DI) benefits has come under scrutiny for the length of time the process can take – 1153 days to move through the entire appeals process, according to a recent Social Security Administration (SSA) analysis – and for inconsistencies that suggest a potentially high rate of errors. One inconsistency is the high reversal rate during the appeals process – for example, administrative law judges, who represent the second level of appeal, award benefits in 59% of cases. Another inconsistency is the variation in the award rates across states – from a high of 65% in New Hampshire to a low of 31% in Texas in 2000 – and over time – from a high of 52% in 1998 to a low of 29% in 1982.’’

[1] This paper is a revised and consolidated version drawing on work presented at ECSQUARU 2007, COMMA 2008 (Wardeh, M., Bench-Capon, T. and Coenen, F.P. (2008). Arguments from Experience: The PADUA Protocol. Proc. COMputational Models of Argument, COMMA'2008, IOS press, pp 405-416.) and JURIX 2008 (Wardeh, M., Bench-Capon, T. and Coenen, F.P. (2008). Argument Based Moderation of Benefit Assessment. Proc. JURIX'08 (21st International Conference on Legal Knowledge and Information Systems), pp128-137.).

Similar observations are made of the UK. An official UK Publication produced by the Committee of Public Accounts [3]: ‘‘Finds that the complexity of the benefits system remains a major problem and is a key factor affecting performance. Skills of decision makers need to be enhanced through better training and wider experience. Too few decisions are right first time, with an error rate of 50% for Disability Living Allowance. There are also regional differences in decision making practices that may lead to payments to people who are not eligible for benefits.’’

A 2006 report from the UK national Audit Office [4] estimated losses from error in Social Security benefits at around £1 billion per annum, and stated that “Errors by officials arise mainly because of the sheer complexity of benefit rules and regulations.”

Although here it is the complexity of the rules that is emphasized, significant aspects of this complexity include the existence of numerous special cases and rarely encountered exceptions, and some easily misunderstood interpretations of the legal conditions. Our example will include these aspects.

In section 2 we will give some general motivation for, and description of, argument from experience, and show why it is a distinctive approach to persuasion dialogues in AI. In particular we will identify some dialogue moves typical of that style of persuasion. Section 3 will present the PADUA protocol, while section 4 will describe some of the strategies and tactics that can be used in the protocol. Section 5 illustrates PADUA with some examples applied to a particular (fictional but characteristic) welfare benefit scenario. Section 6 gives a detailed empirical study designed to explore the effectiveness of PADUA in moderating decisions containing a large number of errors. Section 7 concludes the paper.

2. Arguing from Experience

PADUA is intended to offer a particular form of persuasion dialogue. Thus far, both in AI and Law, and in AI in general, the majority of such 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 [5]. In this work Prakken identifies the speech acts typically used in such dialogues:

- *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 he is 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).
- *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, for example, 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 [6] 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 Mackenzie [7] who was interested in exploring a particular logical fallacy. The take up in Computer Science has largely been by people 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. On 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, they will have systemized their experience into what might be termed knowledge, or have been taught a theory. While persuasion dialogues of this form do take place in practice, others take a different form, 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.

In AI and Law the closest style of argumentation to this is found in case based approaches to common law, especially as practiced in the US, where arguments about a case are typically backed by precedents, such as [9,10]. Even where decisions on past cases are putatively encapsulated in a proposed rule, the particular facts are still considered and play crucial roles in the argument, in particular to test the proposed rule. In informal everyday argument also the technique is common: "the last time we did this that happened". Given the prevalence of such arguments, it is worthwhile to address the requirements for such dialogues and how they differ from the traditional persuasion dialogues described in [5]. Quite apart from the widespread use of arguments from experience of this sort there are compelling pragmatic reasons for investigating such arguments. The formation of effective belief bases requires a good deal of, typically expensive and skilled, effort. The so called knowledge engineering bottleneck has bedevilled the practical implementation of knowledge based systems throughout their history. If we see the dialogue system as a way of adding value to existing systems, we will find that there are very few suitable belief bases available. In contrast, there are many large datasets available, with each record in the dataset representing a particular case, a particular experience. This provides an extensive amount of experience to draw on, if we can find a way of deploying it through argumentation.

In the context of these arguments from experience, typically all of the facts regarding the case under consideration are available at the outset. Thus this source of incompleteness, resolved through belief based persuasion dialogues, is not present. Nor can the rules be incomplete or give rise to inconsistency: there are no rules. In such arguments persuasion occurs not through one participant telling the other something previously unknown, but rather because the experience has been incorrectly or unsafely generalised to the current case, or because - importantly - experience differs from participant to participant, and one participant may have encountered an untypical or overly narrow set of examples. For example, generalising on experiences confined to the Northern hemisphere, one might conclude that a bird was white on being told that it was a swan, but should be open to persuasion by another participant with experience of Australia also.

Having seen a need to model arguments from experience, we now need to consider what speech acts will be typical of such dialogues, and see how they differ from those typical of the belief based persuasion dialogues identified by Prakken. One field in which arguing on the basis of precedent examples is Law. Important work has been carried out by, amongst others Rissland, Ashley and Aleven [8, 9,10]. We will draw inspiration from this work, although we must note two important differences. Case based approaches tend to use particular, often landmark, decisions, the persuasiveness of the precedent coming from the authority conferred by its status. In contrast, in PADUA the persuasiveness of the reasons offered comes from the weight of numbers in the past experience. Second: approaches such as [9, 10] use factors rather than bare facts, which involves some prior analysis and some conceptualization of the domain. In PADUA we use simple facts, as transcribed from claim form to database, and rely on the system itself to find factors or similar intermediate predicates [11]. What has emerged from the work on Case Based Reasoning is there are three key types of move:

- Citing a case
- Distinguishing a case

- Providing a Counter Example

We will discuss each of these in turn, anticipating the next section by indicating in brackets the name of the corresponding speech acts in the PADUA protocol. Citing a case involves identifying a previous case with a particular outcome which has features in common with the case under consideration. Given these things in common, the suggestion is that the outcome should be the same. Applied to argument from experience regarding the classification of an example, the argument is something like: *in my experience, typically things with these features are Cs: this has those features, so it is a C* (PADUA's *propose rule*). The features in common are thus presented as reasons for classifying the example as *C*, justified by the experience of the large proportion of previous examples with these features which were classified as Cs.

Distinguishing is one way of objecting to this, 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 (type 1). For example the feature may exhibit an exception: *although typically things with these features are Cs, this is not so when this additional feature is present* (PADUA'S *distinguish*). As an example, swans are typically white, but this is not so for Australian swans. Another form of distinction (type 2) is to find a missing feature that suggests that the case is not typical: *while things with these features are typically Cs, Cs with these features normally have some additional feature, but this is not present in the current example* (PADUA's *unwanted consequences*). Suppose we were considering a duck billed platypus: while we might classify it as mammal on the basis of several of its features, we would need to consider the objection that mammals are typically viviparous, whereas the platypus lays eggs. A third kind of distinction would be to supply a more typical case: *while many things with these features are Cs, experience would support the classification more strongly if some additional feature were also present* (type 3) (PADUA's *increase confidence*). For example, we would be more confident in classifying a water bird with black feathers as a black swan if this bird has the additional feature of coming from Australia[2].

Thus we can have three types of distinction, with differing forces. The first argues that the current example is an exception to the rule proposed; the second that there are reasons to think the case untypical, and so that it may be an exception to the rule proposed; the third argues no more than that confidence in the classification would be increased if some additional features were present. 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 point about confidence is important: arguments from experience are typically associated with some degree of confidence: our experience will suggest that things with certain features are often/ usually/ almost always/ without exception *Cs*. This is also why dialogues to enable experience to be shared 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 to a certain class of exceptions: a person classifying swans with experience only of the Northern hemisphere, needs this to be supplemented with experience of Australian swans. In less extreme cases, it may only be the confidence in the classification that varies.

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. The objection here is something like: *while these features do typically suggest that the thing is a C, these other features typically suggest that it is not* (PADUA's *counter rule*). Here the response is either to argue about the relative confidence in the competing reasons, or to attempt to distinguish the counter example. Thus a dialogue supporting argument from experience will need to accommodate these moves: in the next section we will describe how they are realized in the PADUA protocol.

Another lesson from work on reasoning with legal precedent is the importance of intermediate concepts e.g. [11]. The point is analogous to the difficulty in classifying examples of *XOR* using a single layer perceptron [12]. No simple classification rule for *XOR* over two variables can be produced using only the truth functions of the inputs. Rather we must produce the intermediate classifications "and" and "or" and then classify in terms of these ("or" and not "and"). So too, with

[2] Black swans (Cygnus atratus) are native to the Southern hemisphere.

law[3]: 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. Dialogues representing arguments from experience should therefore be able to accommodate a degree of nesting, where first the applicability of intermediate concepts is resolved, and then used in the main classification debate.

3. PADUA protocol

In this section we describe PADUA (*Protocol for Argumentation Dialogue Using Association Rules*) an argumentation protocol designed to enable participants to debate on the basis of their experience. We first describe what agents require to participate in PADUA. PADUA has as participants agents with distinct datasets of records relating to a classification problem. These agents produce reasons for and against classifications by mining association rules from their datasets using standard data mining techniques [13, 14, 15]. By "association rule" we mean no more than that the antecedent is a set of reasons for believing the consequent.

Before moving on to describe the details of the PADUA protocol we provide an example dialogue illustrating how the protocol works. Let us assume that we have two parties arguing about the classification of some water bird they are watching, the proponent is in favour of classifying the bird as a black swan, while the opponent is rejecting the proponent claim. The proponent starts the dialogue by "proposing a new rule" as follows: *"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"*.

The opponent may then distinguish the previous argument by saying: "*Although water birds which have red bills and mate for life are highly likely to be swans but this particular bird has black feathers, in my experience water birds with black feathers are not likely to be swans"*.

The proponent may respond to this attack by saying: "*Although water birds which have black feathers are unlikely to be swans in the Northern hemisphere, but this particular bird comes from Australia. In my experience water birds with black feathers that have all the other features we've been talking about and live in the southeast and southwest regions of Australia are* **more** *likely to be black swans"*

This three steps dialogues is provides a short example how PADUA is applied, and the type of dialogues produced by the proposed protocol. More in-depth examples are discussed later in this paper. In what follows $P \rightarrow Q$ should be read as "*P* are reasons to believe *Q*". Association rules are associated with a degree of *confidence* (the percentage of examples satisfying P which are Qs) and a degree of *support* (the percentage of Qs which satisfy P). The six dialogue moves in PADUA relate to the argument moves identified above. One represents citing a generalization of experience, three pose the different types of distinction mentioned above, one enables counter examples to be proposed, and one enables a rule to be refined to meet objections.

3.1. PADUA Framework

This section introduces the formal framework for the PADUA protocol. This framework borrows various elements from the different formal systems suggested by Amgoud et al. [16] and Prakken [5] and [6] and employs these elements in the context of PADUA dialogue mode. This framework is modelled as the following tuple:

$$PADUA\ Dialogue\ Framework = <Lt, Lc, DP, \varphi, A, E, P, O, S>$$

Where:

1. ***L*$_t$:** The *topic language* of the PADUA dialogue game:
   - $I = \{i_1, i_2, .., i_n\}$ be the set of items. Each item $i_k \in I$ has a set of possible values $V_i = \{v_{i1}, v_{i2}, v_{im}\}$, these values can be anything (Numbers, Strings, etc).
   - D = the set of database records, each record $T \in D$ is a subset of the items in I. A transaction T satisfies a set of items $X \subseteq I$ if and only if $X \subseteq T$.

[3] In some cases legal intermediate predicates are used for convenience, but in others there does seem to be no truth functional relationship between the factors considered in applying the concept: the example in [39] is "living together as husband and wife". The distinction between these different kinds of intermediate concept is also discussed in [24].

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

2. ***L_c:*** The communication language comprising:
   - **Speech Acts:** denoted SA = {propose rule, distinguish, unwanted consequences, counter rule, increase confidence, withdraw unwanted consequences} where:
     1. *Propose Rule*: this speech act proposes a new rule with a confidence higher than the threshold (*Conf*), (in the case of two players games the confidence of this rule should also be higher than any other move played by the other side).
     2. *Distinguish*: this act adds some new premise(s) to a previously proposed rule, such that the confidence of the new rule is lower than the confidence threshold (Conf)
     3. *Unwanted Consequences*: This speech act suggests that certain consequences (conclusions) of some rule previously played in the dialogue do not match the studied case.
     4. *Counter Rule*: this speech act places a new rule that contradicts the previous rule. The confidence of the proposed counter rule should be higher than the confidence of the previous rule (and higher than the threshold *Conf*).
     5. *Increase Confidence*: this speech act adds some new premises to a previous rule so that the overall confidence rises to some acceptable level.
     6. *Withdraw Unwanted Consequences*: this act excludes the unwanted consequences of the rule it previously proposed, while maintaining a certain level of confidence (at least higher than the confidence threshold *Conf*).
   - **Moves:** a move $m \in M$ (the set of all moves) is defined as a tuple $<sa, content>$ such that:
     - $sa \in SA$ is the move speech act (type)
     - content is the content of this move:
       - If ($sa \neq$ *Unwanted Consequences*): *content* has the form of $ar(p \rightarrow Q, conf)$.
       - If ($sa =$ *Unwanted Consequences*): $content = U \subset I$ (the set of unwanted consequences).
   - **Dialogue Moves:** a dialogue move dm∈DM (the set of all dialogue moves) is defined as a tuple $<S, H, m, t>$ such that:
     - $S \in Ag$ is the agent that utters the move, given by *Speaker(dm)* denotes *S.*.
     - $H \subseteq Ag$ denotes the set of agents to which the move is addressed, given by a function *Hearer(dm)* denotes *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).
   - **Dialogues:** The set of dialogues, denoted by $DM^{\leq\infty}$, is the set of all sequences from L_c, and the set of finite dialogues, denoted by $DM^{<\infty}$, is the set of all finite sequences from L_c. For any dialogue $d = dm_1 \ldots dm_n$, the subsequence $dm_1 \ldots dmi$ is denoted with d_i. For any dialogue $d = \{dm_1 \ldots dm_n\}$, the speech act of the first move ($dm1$) is a propose rule. The current dialogue $d_{current}$ is the actual dialogue taking place between the set of participants taking part in every instantiation of the framework.
3. ***DP:*** *The dialogue purpose*, the dialogue purpose of PADUA dialogues is the resolution of conflicting opinions about the classification of an instance $\varphi \subseteq I$, for example in the case of two players (the proponent and the opponent), the proponent may claim that the case falls under some class (c_1), while the opponent opposes the proponent claim, and tries to prove that case actually falls under some other class ($c_2 = \neg c_1$). This purpose is met when the dialogue is terminated, and is identified with the classification proposed by the winner of the dialogue game.
4. ***φ:*** The instance argued about i.e. the dialogue subject. This instance is identified 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 $v\varphi_j \in V$. each instance of the framework has one instance case, the participants in any PADUA game

will argue about this instance case, each trying to convince the others that this case falls under one possible class. The result of any PADUA game will be a classification of this instance case.

5. ***A:*** the set of participants (players): A= $\{a_1,.. a_n\}$. Each player in PADUA game is defined as:

$$\forall a \in A \text{ then } a = <name_a,\ I_a, G_a, \Sigma_a, C_a>.$$

Where:

- $name_a$: the player (agent) name
- I_a: the set of items this player can understand (included in the player's database).
- G_a: the set of attributes (classes) this player tries to prove true. $G_a \subseteq I_a$.
- C_a: the commitment store of agent (player) a.
- Σ_a: is a representation of the player's background database enables this player to mine for the suitable association rules as needed, for example Σ_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 this player, R_a is the set of association rules previously mined by this player (i.e. $R_a = \{ar: ar(P \rightarrow Q, conf)\}$), and Drp is a function that maps between legal moves and their suitable rules ($Drp : Tp \times M \rightarrow R$, where R is the set of all possible association rules).

6. ***E:*** effect rules for L_c, specifying for each move dm $<a, H, m, t> \in DM: a \in A$ its effects on the commitments of the participants. These rules are specified in Table 1.

Table 1 – Effect Rules

Rule	Played move	Effects
E1	propose rule - counter rule ar(P→Q, conf)	$C_a = C_a \cup Q$ $\forall h \in H\ C_h = C_h$
E2	unwanted consequences (U)	$Cp = Cp$ $\forall h \in H\ C_h = C_h - U$
E3	distinguish, increase confidence	$C_a = C_a$ $\forall h \in H\ C_h = C_h$
E4	withdraw unwanted consequences (P→Q', conf)	$C_a = C_a \cup Q'$ $\forall h \in H\ C_h = C_h$

7. ***P:*** A protocol for L_c specifying the legal moves at each stage of a dialogue. P is defined formally as a function: $P: M \rightarrow 2^M$, where M is the set of dialogue acts (moves). Thus, P links each speech act with a set of possible next moves that are legal in the context of PADUA games. Table 2 lists the *possible next moves* after each move in PADUA protocol.
   - *Termination Rules*: in PADUA, the dialogue ends when a player fails to play a legal move in its turn, in this case, this particular player loses the game while the other player wins it.
   - *Turn taking Rules*: PADUA applies a simple turn taking policy, in which each player is allowed to play exactly one move in its turn, and the turn shifts to the other agent in the dialoguer.

Table 2 – The protocol legal next moves. New Rule column indicates if the speech act (move) produces a new rule, or refer to a previous move without producing any new rule (this is the case in *distinguish* and *unwanted consequences*).

Move	**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	Yes
5	Increase Conf	3, 2, 4	Yes
6	Withdraw Unwanted Cons	3, 2, 4	Yes

8. ***O:*** The Outcome rules of PADUA dialogues define for each dialogue d and instance φ the winners and losers of d with respect to instanceφ. In two-player PADUA, the winner of a PADUA 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 the output of the dialogue. The outcome of the dialogue is defined as the class attribute of the association rule of the last move played in this dialogue. More precisely *O* consists of two functions w_φ and l_φ. the first returns the winner of the game and the second returns the loser of the game
   - w_φ(a∈A) = true if outcome(D) ∈ G_a .
   - l_φ(a∈A) = true if outcome(D) ∉ G_a .
   - Outcome (D∈$DM^{<\infty}$) =o : o ∈ $G_{a1} \cup G_{a2}$ *and* **o** ∈consequences of the content of the last move played in D.
   - The two functions w_φ and l_φ satisfy the following conditions:
     - $w_\varphi(d) \cap l_\varphi(d) = \varnothing$
     - $w_\varphi(d) = \varnothing$ *iff* $l_\varphi(d) = \varnothing$
     - if $|A| = 2$, then $w_\varphi(d)$ and $l_\varphi(d)$ are at most singletons
9. ***S:*** A Strategy function (the definition of this function as used in PADUA is listed in details in section 4)

3.2. Nested Dialogues

PADUA allows for dialogues to be nested so that a number of secondary dialogues may take place to solve the sub disputes over some intermediate classifications, before the main dialogue over the main classification starts. This is because, in some domains, cases under discussion may have *intermediate attributes* (i.e. attributes that should be considered together as one unit as they describe one concept in domain). Such attributes increase the complexity of the dialogues, leading in some cases to misclassifications or in the least to irrelevant steps in the dialogue (when players regard these attributes separately rather than as a whole unit). By resolving intermediate classifications over intermediate attributes first, PADUA decreases the complexity of the main dialogue by replacing any intermediate attributes with the result of the intermediate concept these attributes relate to, hence focusing the main dialogue on the main issues only.

To realize this view of nested dialogues, a Control Layer has been implemented into the PADUA Framework. 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: The formalization of the PADUA control layer is defined in the terms of the following components:

- *Players*: is the set of players engaged in all the PADUA dialogues controlled by this layer.
- *Gs*: set of PADUA secondary dialogue games. Each $gs \in Gs$ is defined as an instance of PADUA framework.
- *gm*: PADUA main dialogue game, defined as an instance of PADUA framework.
- *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.

It is important to note, that nested dialogues always take place prior to the start of the main dialogue, the players engaging in these nested dialogues will argue for and against some intermediate classification of a sub case of the case under discussion; the output of each nested dialogue represents a resolution between the players as to what intermediate class would replace the sub case of each nested dialogue, in the main dialogue regarding the main case. Consider, for example, the case where two players are arguing if some vehicle is a car or not; the two players may first engage in a nested dialogue regarding if some of the attributes of this vehicle represent a car door or not, let us assume, that the side believing these attributes present a car door won this dialogue. This result could then be applied in the main dialogue, replacing the set of attributes included in the nested dialogue with the final output of this dialogue (e.g. the vehicle has a car door). The main dialogue then could proceed, the case argued about will include any attributes that were not included in the nested dialogue and one new attribute : (the vehicle has a car door = yes).

4. Strategies and Tactics for PADUA

4.1. Dialogue Strategies

We first discuss some previous argumentation systems that have considered argument selection strategies:
Moore, in his work with the DC dialectical system [17], concluded from his studies that an agent's argumentation strategy is best analyzed at three levels:

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

The first two levels refer to the agent's strategy, i.e. the high level aims of the argumentation, while the third level refers to the tactics, i.e. the means to achieve the aims fixed at the strategic levels. In [18] a computational system was suggested that captures 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. When using the build strategy (b-strategy), an agent tries to assert arguments the strength of which satisfies its prudence level. If the b-strategy fails, it switches to the destroy strategy (d-strategy), where it tries to use any possible way to attack the opponent's arguments. The basic drawback of this approach is that computational limits may affect the agent's choice.

In [19] a three layer system was proposed to model argumentation strategies: the first layer consists of the "default" rules, which have the form (utterance- condition); the higher two layers provide preference orderings over the rules. The system is shown to be deterministic, i.e. a particular utterance is selected in a given situation every time, but this system still requires hand crafting of the rules. In [20], a decision heuristic was proposed to allow the agents to decide which argument to advance. The idea behind this work is that an agent should, while attempting to win a dispute, reveal as little of what it knows as possible, as revealing too much information in a current dialogue might damage an agent's chances of winning a future argument. A new argumentation framework was developed to represent the suggested heuristics and arguments. The main shortcoming of this approach is the exponential complexity of the algorithms used.

4.2. Strategies and Tactics for PADUA

In PADUA, players must select the kind of move to be played, and also the particular content of this move depending on: the classification this player is arguing for (or against), the case under discussion, the association rules that are supported by the player's dataset, the amount of information this agent is willing to expose in its move, and the player's current state in the dialogue. All these factors must be considered in the strategy the player adopts and the tactics applied to implement this strategy. It worth noting that we use the term "strategy" here to refer to a "high level" specification of how players may plan their moves in the context of PADUA dialogue to achieve their goals (for example, convincing the other side that the case under discussion classifies according to the players' proposed class). In our model each "strategy" comprises of three type of tactics designed to fulfill the strategy. In this sense, each player may adopt a strategy from a range of possible strategies, and then apply this strategy by selecting the tactics that match this strategy[4].
Table 2 listed the possible next moves after each of the legal moves in PADUA protocol. A player must select a single move to play in its turn; moreover every possible next move can be linked to a set of possible association rules: this set contains the association rules that match the selection criteria of the move, i.e. their confidence; premises and conclusion match this move, for example, in case of distinguishing, the confidence of the new rule should be lower than the acceptance level (Conf) and that the premises and the conclusion of the new rule should include the premises and conclusion of the distinguished rule (respectively). Except for unwanted consequences, the moves introduce a new rule. Proposing a counter rule leads to a switch in the rule being considered. The notion of move (act) and content selection is argued to be best captured at different levels, as

[4] Our use, which is standard English, should not be confused with the very particular meaning given to the term "strategy" in Game Theory.

suggested by Moore [17]. In [18, 21] the first level of Moore's layered strategy was replaced with different profiles for the agents involved in the interaction. We also adopt this approach. Here we also add another level to Moore's structure (level 0) which distinguishes PADUA games into two basic classes. In one (which we call *win mode*) players attempt to win using as few steps as possible, i.e. the move's type and content are chosen so that the played move gives the opponent's the least freedom to plan its next move. Alternatively, in what we call *dialogue mode*, games are played so as to explore the characteristics of the underlying argumentation system, and dialogue game, the argument is, by giving the participants a chance to engage in longer dialogues we may uncover interesting observations regarding the various aspects of PADUA that would have remained uncovered otherwise. So in this case the move's type and content are chosen so that the played move will restrict the opponent's freedom to plan its next move to the least extent possible. The first level of Moore's layered strategy was also reconsidered, here we use agents' profiles to maintain the focus of the dispute. The layered strategy system we adopt consists of the following levels (layers):

- Level 0: Define the game mode: i.e. *Win* mode or *Dialogue* mode.
- Level 1: Define the players (agents) profiles.
- Level 3: Choose some appropriate argumentative content: depending on the tactics and heuristics suggested.

Agent Profile

In [21], which used arguments based on standard *if then* rules; five classes of agent profiles were defined as follows:

1. Agreeable Agent: Accept whenever possible.
2. Disagreeable Agent: Only accept when no reason not to.
3. Open-minded Agent: Only challenge when necessary.
4. Argumentative Agent: Challenge whenever possible.
5. Elephant Child Agent: Question whenever possible.

In PADUA we have used only the first two profiles (i.e. agreeable and disagreeable agents), as these attitudes are the most appropriate for the particular style of argument we are using.

PADUA Strategies

For each player taking part in the PADUA dialogues, a∈A, The function $Play_a$ is defined as follows:

$$Play_a : M_{poss} \times R_{poss} \times D_{current} \times S_a \times \text{Tactics}_a \rightarrow M_{poss}$$

Where: $D_{current}$ is the current dialogue this player is taking part in (as identified in the previous section), thus $D_{current}$ represent 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 2). R_{poss}: is the set of legal rules that this agent can put forward in the dialogue; this set contains the rules that match the 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 details in the following sub-section.

PADUA Tactics

In this subsection a set of tactics are suggested to fulfill the strategic considerations discussed above; these concern the best move to play and, where applicable, the content of the chosen move, i.e. the best rule to be put forward in the dialogue. We begin by ordering the legal moves. The idea is to characterize a strategy with the order in which legal (possible) moves are considered when selecting the next move. 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 agent is following a build or a destroy strategy. In a destroy strategy the agent will wish 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 an agent will prefer to propose its own rule, and will only attempt to discredit the rule of its opponent if it has no better

rule of its own to put forward. The preferred order for the two strategies is shown in Table3. Whether players are agreeable or disagreeable will have an influence on whether the agent wishes to dispute the rule put forward by its opponent, and, the nature of the challenge if one is made.

Table 3 – Possible Moves Preferences

Last Move	Label of the Last Move	Preferable next move in Build Mode	Preferable next move in Destroy Mode
1	Propose Rule	4,3,2	3,2,4
2	Distinguish	1,3,6	3,6,1
3	Unwanted Cons	1,6	6,1
4	Counter Rule	1,3,2	3,2,1
5	Increase Conf	1,3,2	3,2,1
6	Withdraw Unwanted Cons	1,3,2	3,2,1

Profile Tactics:
The Agent Profile tactic articulates how each profile affects the players' criteria for attacking an adversary. Agreeable players tend to be less aggressive: if an agreement with the others 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 the proposed rule would be acceptable according to their own data. Agreement will not be conceded as long as there is a and room to maneuver. In the following the two profiles (Agreeable and disagreeable) used in PADUA are described.

Agreeable Players: An agreeable player $ap \in P$ accepts a played rule without challenging it if:

- An exact match of this rule can be mined from the player's background dataset (Σ_{ap}) with a higher or similar confidence.
- A partial match of this rule can be mined from the player's background dataset (Σ_{ap}). a rule $r_{pm} \in \Sigma_{ap}$ is considered to be a partial match of another rule $r \in \Sigma_{ap}$ if it has the same conclusion (consequences) of r, its set of premises is an superset of rule r premises, and all these premises match the case; and finally it has a higher or similar confidence.

Otherwise the agreeable agent challenges the played move, depending whether it wishes to build or destroy using the legal moves preferences shown in Table 3 selecting a rule using the following content tactics:

- Confidence: Confidence of moves played by agreeable agent should be considerably lower/higher than the attacked rule, otherwise the agent agrees with its opponent.
- Consequences: Consequences always contain a class attribute. Minimum changes to previous move consequences. As few attributes as possible.
- Premises: Premises are always true of the case. Minimum changes to previous move premises. 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 Table 3 and the choice of rule is in accordance with the following content tactics:

- Confidence: Confidence of moves played can be:
  1. Considerably different from last move
  2. Slightly different from last move.
  3. The choice of confidence depend on the general mode of game whether it is in a win-mood or a dialogue-mood.
- Consequences: Consequences always contain a class attribute. As few attributes as possible.
- Premises: Premises are always true of the case. As few attributes as possible.

Best Move
Table 4 brings these considerations together and shows the best move relative to the agent type and the game mode, for each of the move types. For example in win mode an agent will want to propose a rule with high confidence, as one which the opponent is likely to be forced to accept, whereas in game mode, where a more thorough exploration of the search space is sought, any acceptable rule can be used to stimulate discussion.

Table 4 - Best move content tactics. *Here moderate confidence means that the confidence in the rule chosen by the player is within a certain range, the limits of this range differ according to the player, and should be determined before the start of the game.*

Best Moves				
	Agreeable Disagreeable		Agreeable Disagreeable	
	Win mode	Game mode	Win mode	Game mode
Propose	High confidence	Average confidence	High confidence	Moderate confidence
	Fewest attributes	Moderate attributes	Fewest 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 confidence
	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.			

4.3. Strategies summary

PADUA provides a way of determining the classification of cases on the basis of distributed collections of examples related to the domain without the need to share information stored in each the datasets of the players, and without the need for analysis and representation of the examples. The argumentation leads to a classification which, while uncertain, is mutually acceptable and consistent with the different collections of examples. It is important to note that although it is possible to mine association rules that are not classification rules, the underlying association rule mining algorithms used in PADUA are modified such that they only consider mining classification rules, as only these rules are related to the context of PADUA games. Different strategies for move selection give rise to dialogues with different characteristics. Using disagreeable agents gives rise to a persuasion dialogue, since the opponent will do anything possible to avoid accepting the proposal. Win mode will lead to the swiftest resolution: game mode between disagreeable agents will lead to a lengthier exchange, and concession may be forced without the best argument being produced. A dialogue between two agreeable agents has the characteristics of a deliberation dialogue in that here the opponent is happy to concede once an acceptable proposal has been made. Win mode between agreeable agents may be a very short exchange, since this simply verifies that the proponent's best rule is also acceptable with respect to the second agent's data set. When game mode is used, the game has the flavour of brainstorming in that more ideas, even some which are less promising, will be explored.

5. Examples

5.1. Examples datasets

To illustrate experimentally the kinds of dialogues produced by PADUA, and to demonstrate its usefulness in the legal domain, we have applied PADUA to a fictional welfare 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). Although fictional the conditions have been designed to be typical of those

found in such benefits. According to the putative legislation, the benefit 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 the United Kingdom, or absent only by virtue of "service to the nation", and should have an established connection with the UK labour force. These conditions need to be interpreted [22] if they are to be applied to the facts of individual cases. Interpretations may be communicated to the adjudicators through regulations, centrally issued guidelines, or they may grow up through custom and practice as the adjudicators exercise their discretion. We suppose the following to be the interpretations desired by the policy makers:

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 for another benefit.
4. *Residence condition*: "Resident in this country" is interpreted as having a UK address. Residence exception: "Service to the Nation" is interpreted as a member of the armed forces.
5. *Contribution condition*: "Established connection with the UK labour force" is interpreted as having paid National Insurance contributions in 3 of the last 5 years.

These conditions fall under a number of typical types: conditions (2 and 3) represent necessary conditions over continuous values while conditions (4 and 5) represent a restriction and an exception to the applicant's residency. 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. These conditions have been used in several previous experiments in AI and Law e.g. [23], [35]. We have conducted a number of experiments using datasets based on this benefit to explore the effectiveness of PADUA, some designed to investigate individual dialogues for insight into various aspects of its working and to examine the different strategies, and others to evaluate its effective in moderating large batches of cases. In 5.2 we present an example to explore nested dialogues relating to intermediate predicates, and in section 6 we present the evaluation of effectiveness of moderation.

5.2. Nested dialogue Example

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 legalization 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 of 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 association rules were mined from these sets 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.

In the example discussed below the applicant is a male aged around 70 years, a UK resident who satisfies all the entitlement conditions except that he had paid contributions to the UK labour force in three out of the last *six* years (namely last year, the year before that and 6 years before), rather than three out of the last five as required. This is the case φ `to be` argued about between the game players (A = {*proponent*, *opponent*}) ; the datasets we used are those described in the last paragraph. Both players use disagreeable profiles, and PADUA game is played in *Game* mode so that we can explore the different aspects of PADUA dialogues. The proponent agent plays in build mode while the opponent agent plays in destroy mode. The dialogue purpose *DP* is to decide whether this applicant is entitled to housing benefit or not, where the proponent says he does not

($C_{proponent}$ = {(*entitles*, *no*)}) while the opponent thinks he does ($C_{Opponent}$ = {(*entitles*, *yes*)}). The dialogue starts with the proponent proposing the rule (`R1: contr y5= not paid -> entitles= no`) with a confidence= 73.14%. The opponent then tries to *distinguish* this rule in the light of to its own experience. For the opponent the rule (`R2: contr y4= not paid, contr y5= not paid -> entitles= no, capital > 3000`) holds with confidence = 2.34% only. This rule is available to the opponent because the opponent uses an incorrect interpretation based on its own data, in which the sixth contributions year is considered, also the opponent prefers this move to other possible moves because of its destroy mode strategy. This move can be defeated by the proponent using *unwanted consequences* since (`capital>3000` does not hold). The opponent then proposes a *counter rule* (`R4: age>=65, residence= UK, gross disposable income <20%, 2500 < capital <3000 -> entitles= yes`) with 77.11% confidence, but the proponent can successfully *distinguish* this rule by repeating the fact that the candidate has not paid the contribution fees in the fifth year. Here too the proponent chose to play this move to lengthen the dialogue according to the "game" mode the dialogue is using. The dialogue then progresses in a similar way with the proponent focusing on unpaid contributions in various years and the opponent trying to get away from this topic in accordance with its own interpretation. For example the proponent proposes rules: such as (`R13: contr y3= not paid, contr y5= not paid -> entitles= no`) (88.77% confidence), (`R21: gender= male, contr y3=not paid, contr y5= not paid -> entitles= no`) (89.39% confidence). These cannot, however, be accepted by the opponent.

Finally the proponent puts forward the rule (`R23: contr y3= not paid, contr y4= not paid, contr y5= not paid ->entitles= no`), with a confidence of 89.39%, and this rule successfully exposes the mistake in the case under discussion, as by playing this rule the proponent manages to indicate the three years in which the contributions were not paid. The opponent tries then to distinguish this rule by manipulating its premises so it plays the rule (`R24: gender= male, contr y3= not paid, contr y4= not paid, contr y5= not paid -> entitles= no, contr y2= not paid`) in which the confidence falls to 37.89%, but the opponent's move is defeated by the *unwanted consequences* attack (the second year contribution is actually paid). The dialogue ends here as the opponent fails to defeat the rule R23 and the proponent wins the game, and the candidate is classified as not entitled to the housing benefits. This game takes 24 moves, but eventually reaches the correct decision.

Unfortunately when *n* out of *m* attributes are needed to decide whether a condition is satisfied or not it is not always the case that the classification process will run correctly, like the contribution years in our example, each applicant should have paid contribution in "any" three out of the last five years to qualify for benefits, and in every dialogue the players consider each of these five years as a stand-alone attribute, which may confuse them and affect the final result of the dialogue. More reliable results can be achieved if we allow for 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 if, we take the case φ_2 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 contribution years are considered as independent factors, as can be shown by some of the rules played in the dialogue:

```
R1-proponent-Propose Rule: contr y5= not paid -> entitles=no: confidence=
73.14.
R23-proponent-Propose Rule: gender=male, contr y2= not paid, contr y5=
not paid -> entitles=no: confidence=87.69.
R29-proponent-Propose Rule: residence=UK, contr y1= not paid, contr y2=
not paid, contr y5= not paid -> age>=65, entitles=no: confidence=95.31
```

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

```
R30-opponent-Counter Rule: age>=65, residence=UK, contr y3= paid, net
disposable income <20%, capital <2500 -> entitles=yes: 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 PADUA attacks. Here the inability of the proponent to force acceptance of any of its proposed rules means that a mistake is made. Table 5 shows how, by applying two dialogues

(nested and main) to the same case, so that the contributions issue can be settled separately, the proponent becomes able to win the game; and that by winning the nested dialogue over contribution years first, the result of that dialogue can then be applied to the main dialogue. As we can see in Table5 the two players engaged first in a nested dialogue regarding a sub-case of the main case including only the attributes representing the contribution years, this game ends in favour of the proponent who believed that the candidate has not paid enough contribution in the past five years; and the two players apply this result in the main dialogue over the main issue: if the candidate entitles to benefit or not. The attributes presenting the contribution years in the original case are replaced with one attribute (contribution paid = no) and a PADUA dialogue proceeds as usual, this dialogue also finishes in favour of the proponent. Of course, to apply this method, we must first identify the intermediate concepts which require this special treatment, and so must perform at least some of the sort of analysis found in [9,10]. This further strengthens the argument made in [11] and in other work such as [25] which stresses the crucial role of intermediate concepts.

Table 5 – Example Nested Dialogue.

Nested Dialogue	Main Dialogue
1 – proponent: Propose Rule {contribution y1 = not paid, contribution y5 = not paid}→{contribution=no}, Confidence = 74.71 2 – opponent: Distinguish {contribution y1 = not paid, contribution y3 = paid ,contribution y5 = not paid}→{contribution=no}, Confidence = 30.00 3 – proponent: Increase Confidence {contribution y1 = not paid, contribution y2 = not paid, contribution y3 = paid ,contribution y5 = not paid}→{contribution=no}, Confidence = 100.00 4 – opponent: Distinguish {contribution y1 = not paid, contribution y2 = not paid, contribution y3 = paid, contribution y5 = not paid, contribution y6 = paid}→{contribution=no}, Confidence = 30.00 5 – proponent: Increase Confidence {contribution y1 = not paid, contribution y2 = not paid, contribution y3 = paid, contribution y4 = paid, contribution y5 = not paid, contribution y6 = paid}→{contribution=no}, Confidence = 100.00 The opponent fails to counter the proponent attack, and the game ends in favor of the proponent.	1 – proponent: Propose Rule {contribution=no} → {age>65, entitles=no}, Confidence = 94.00 2 – opponent: Distinguish {gender=male, contribution=no} → {age>65, 2500<capital<3000, entitles=no}, Confidence = 18.85 3 – proponent: Unwanted Consequences {gender=male, contribution=no} → {age>65, 2500<capital<3000, entitles=no}, Confidence = 18.85 The opponent fails to counter the proponent attack, and the game ends in favor of the proponent.

6. PADUA as an aid to Moderating Decisions – Empirical Results

One application of PADUA other than generating argumentation dialogues, is as a classifier. PADUA could be used to classify new cases taking into account the previous cases stored in the datasets of each of the players taking part in PADUA games. In the following we discuss results obtained from applying PADUA as a classifier applied the RPHA benefit scenario. We generate new datasets, other than the one used in the previous example, then PADUA will be assessed by calculating the accuracy of the classifications it produces. We now suppose that the RPHA benefit, discussed in the previous section, is assessed in two different offices, covering different regions, and each producing errors through a different misinterpretation. We can assume that they will store the same information in the same format since, at least in the UK, the regions use centrally provided software and follow centrally determined procedures. We ran three experiments:

1. An experiment to test the extent to which classification could be improved by moderation using PADUA. This was done using a 10 fold cross validation test using the data stored in each office. A total of five well known classifiers were also applied to the data to provide a

comparison; these classifiers were applied using the data stored in both offices. Section 6.1 provides some background details regarding each of these classifiers.
1. A McNemar test to show the significance of the differences between classifiers.
2. We then performed a more detailed analysis of the performance of PADUA in order to discover some interesting properties of the moderation dialogues.

For tests we generated two sets of data. Each record comprises 12 fields (these fields comprise: gender, age, residency, capital, net income, gross income, one field for each of the considered contribution years) the information relevant to the above tests being surrounded by other features which should be irrelevant to the determination of the case. Both contained 500 cases where the benefit was "correctly" awarded and 500 cases where the benefit was "correctly" denied. Cases can fail on any one of five conditions, and the failing cases were evenly divided across them. One dataset 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 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 as well as members of the armed forces. Thus each dataset contains 1500 records (total of 3000 records in both datasets).

6.1. Cross Validation and Comparison with Other Classifiers

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 (recall that this level or error is not unusual). Five other classifiers were used, operating on the union of the two data sets (3000 records). These other classifiers were:

1. *TFPC*: TFPC, Total From Partial Classification ([26], [27]), is a Classification Association Rule Mining (CARM) algorithm founded on the TFP (Total From Partial) Association Rule Mining (ARM) algorithm ([28],[29]); which, in turn, is an extension of the Apriori-T [12] (Apriori Total) ARM algorithm.
   TFPC is designed to produce Classification Association Rules (CARs) whereas Apriori-T and TFP are designed to generate Association Rules (ARs). In its simplest form TFPC determines a classifier according to given support and confidence thresholds. The nature of the selected thresholds is therefore the most significant influencing factors on classification accuracy. A more sophisticated version of TFPC uses a hill climbing technique to find a best accuracy given start support and confidence thresholds.
2. *CBA*: CBA (Classification Based on Associations) is a Classification Association Rule Mining (CARM) algorithm developed by Liu, Hsu and Ma [30]. CBA operates using a two stage approach to generating a classifier:
   - Generating a complete set of CARs.
   - Prune the set of CARs to produce a classifier.
3. *CMAR*: CMAR (Classification based on Multiple Association Rules) is another CARM algorithm developed by Li, Han and Pei [31]. CMAR also operates using a two stage approach to generating a classifier:
   - Generating the complete set of CARs according to a user supplied:
   - Support threshold to determine frequent (large) item sets, and
   - Confidence threshold to confirm CRs.
   - Prune this set to produce a classifier.
4. *Decision Trees*: Classification using *decision trees* was one of the earliest forms of data mining. Ross Quinlan's C4.5 is arguably the most referenced decision tree algorithm [32]. One of the most significant issues in decision tree generation is deciding on which attribute to *split*. Various algorithms have been proposed in the literature. Two are used here:
   - Most frequently supported (or Random) Decision Trees (RDT):
   - Information Gain Decision Trees (IGDT).

   The first selects the first attribute in a list of attributes order according to its support frequency within the entire data set. Information gain [33] is one of the standard measures used in decision tree construction.

Table 6 – The results of the first experiment (the cross validation test). For each classifier the percentage of the correct classifications scored in each trial is displayed.

Trial	DS1%	DS2%	TFPC %	CBA%	CMAR%	RDT%	IGDT%
1	95.125	94.375	64.33	64.33	63.87	95.4	91.8
2	95.5	92.625	64.33	64.33	63.87	95.93	90.87

3	95.125	92.5	64.33	64.33	64.07	96.6	91.8
4	95.5	92.75	64.33	64.33	63.87	95.87	90.87
5	96	88.875	64.33	64.33	64.07	95.87	92
6	96.75	93.25	64.33	64.33	63.87	95.93	92
7	96.375	94.125	64.33	64.33	63.87	95.8	92
8	96.25	93.25	64.33	64.33	63.87	95.8	91
9	94.125	93.875	64.33	64.33	63.87	95.8	91.33
10	94.375	93.875	64.33	64.33	63.87	95.8	91.2
Summary	95.83	92.72	64.33	64.33	63.91	95.88	91.487

The cross validation was achieved by running the experiment 10 times, each time leaving out a randomly selected 10% of the available data, this 10% was used as testing set for each of the classifiers mentioned above (including PADUA). The accuracy of the classification was calculated as the number of correct classifications divided by the number of cases in the testing set. For PADUA, two runs were performed, one in which the agent with Dataset 1 (DS1, where the age condition was wrong) was the proponent (i.e. argued for award of benefit), and one in which the agent with Dataset 2 (DS2, where the residence exception is too broad) was the proponent. Again we used two disagreeable agents, playing in *Game* mode with the proponent agent using a build strategy while the opponent agent uses a destroy strategy. It also worth noting that we have applied a 70% confidence threshold and 1% support threshold for PADUA, TFPC, CBA and CMAR.
The results are presented in Table 6. From this we can see that the three association rule classifiers perform less well than the baseline, suggesting that they do not handle the inconsistency arising from combining the two datasets well. In contrast PADUA and the decision tree based classifiers perform significantly better than the other association rules classifiers (CARS) included in this experiment, attaining above 90% accuracy in all cases. While, however, the decision tree classifiers perform rather consistently throughout the ten trials, 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 will be considered in more detail in section 6.2 below.

Overall we find the level of performance encouraging. For comparison with other AI and Law systems, Bench-Capon [23] reported an accuracy of 98% for this dataset, but that was based on training set in which all the examples were correctly decided, and we would expect incorrect examples to lower the success rate. Ashley and Brüninghaus [34] reported a success rate of 91.4% for IBP, and Chorley and Bench-Capon [23] a success rate of between 91% and 93% for AGATHA, both applied to error free examples of US Trade Secret Law cases[5]. Robustness in the face of noise, which we explore here, is very important given that our data will contain many errors. One effort in AI and Law to explore how learning is affected by noise is [35]. They used a very similar dataset and introduced randomly, rather than systematically, miss decided cases. Their results gave a success rate of 92% with 20% noise falling to 85% with 40% noise for Argument based Explanation, and 89% for 20% noise falling to 83% for 40% noise for a conventional rule induction algorithm CN2. It seems therefore, from this previous work, that the level of accuracy attained by PADUA is towards the top end of what can be expected from successful classification systems in AI and Law.

6.2. McNemar Test

The McNemar test is a non-parametric test designed to explore the hypothesis that one classifier is significantly better than another. As might be expected from the results shown in Table 6, PADUA DS1 and DS2 were significantly better than the three association rule classifiers and RDT, but not significantly better or worse than IGDT. For this test PADUA operated on a set of newly generated cases (1500 cases identified as follows: 500 positive, 500 negative as before and 250 wrongly decided, as appropriate to each database[6]). This data was then used as a test set for the

[5] Since the complete dataset used in [33] is not publicly available, [24] used subset of this data collected from various published papers .
[6] We have generated 250 cases presenting female candidates whose age is between 60 and 64 years and should classify as entitled to the benefits (the agent using DS1 may misclassify these cases as not entitled).

other algorithms the original data used in 6.1 supplying the training set. The same support\confidence thresholds from the previous test were used here as well. As part of the test, we generated detailed information as to which cases are misclassified by one or both of the classifiers under consideration. These results for DS1 are shown in Table 7a and those for DS2 in Table 7b: n00 are cases misclassified by both, n01 are cases misclassified by PADUA only, n10 are cases correctly classified by PADUA and misclassified by the comparator, and n11 are cases correctly classified by both:

Table 7a - Comparison with DS1. The first value represents the actual number of cases, the second represents the percentage of these cases regarding the overall number of considered cases.

	DS2	TFPC	CMAR	CBA	RDT	IGDT
n00	5 (0.3%)	8(0.5%)	8(0.53%)	145(9.7%)	7 (0.5%)	10 (0.67%)
n01	146 (9.7%)	139 (9.3%)	139 (9.3%)	2 (0.13%)	140 (9.3%)	137 (9.1%)
n10	142 (9.45%)	318 (21.2%)	364 (23.6%)	461 (30.7%)	62 (4.13%)	129 (8.6%)
n11	1207 (80.5%)	1035 (69%)	989 (65.9%)	892 (59.5%)	1291(86.1%)	1224 (81.6%)

Table 7b - comparison with DS2. The first value represents the actual number of cases, the second represents the percentage of these cases regarding the overall number of considered cases.

	DS1	TFPC	CMAR	CBA	RDT	IGDT
n00	5 (03.3%)	48 (3.2%)	92 (6.13%)	55 (3.7%)	10 (0.7%)	22 (1.5%)
n01	142 (9.45%)	103 (6.9%)	59 (3.93%)	96 (6.4%)	141 (9.4%)	129 (8.6%)
n10	146 (9.7%)	214 (14.3%)	514 (34.3%)	66 (4.4%)	31 (2.1%)	119 (7.9%)
n11	1207 (80.5%)	1136 (75.7%)	835 (55.7%)	1283 (85.5%)	1318 (87.9%)	1230 (82%)

What is interesting here is that although both classifiers only succeed only on 86% of cases for RDT and DS1, 91% of cases for DS1 and IGDT and 82% of cases for DS2 and IDG; the mistakes are very different. Less than 0.5% of the cases are misclassified both by DS1 and RDT and only 1% by the worst combination, DS2 and IDGT. This suggests that we could profitably use PADUA and a decision tree method in combination. If cases where there was agreement were believed to be correct, and we used, for example, DS1 and RDT; and referred cases of disagreement to an expert for decision we could reduce error rates to below 0.05%, at the cost of checking some 13.5% of the cases. Since it is current practice to check 10% of decisions, chosen at random, we could thus, by focusing the cases for expert checking, substantially reduce the error rate with very little additional expert intervention. Moreover, DS1 and DS2 only both misclassify one case in three hundred, although they are both successful in only 80%. Using PADUA alone, but having each case argued for by both agents, therefore, could reduce the error rate to 0.3%, although it would require around 20% of cases in the overall collection of data (i.e. DS1 and DS2) to be checked. This, however, might be improved by first using an expert to resolve a proportion of the cases with disagreement (say a quarter, 5% of all the cases), entering the corrected values into the databases, and then rerunning the moderation. Since performance improves as noise is reduced, this should generate fewer disagreements, tending to reduce the overall checking requirement to an acceptable level.

6.3. Detailed Consideration of DS1 and DS2[7]

In this section we will look at the ten cross validation trials for PADUA in more detail. The detailed results are shown in Table 8a and Table 8b. Three points in particular can be noted from this data:

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

We also have generated 250 cases presenting candidates from the merchant navy or diplomatic services and should therefore classify as not entitled to benefits (the agent using DS2 may misclassify these as entitled).

[7] The difference in the accuracy scored when the proponent is using DS1 or DS2.

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 the proponent is. For example when DS1 is arguing for benefit for the misclassified age cases, (that is the cases it gets wrong) 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 does not obtain, 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.

It is this last point in particular that suggests that expert resolution of cases where there is disagreement when different agents act as the proponent is likely to be effective in selecting cases for expert checking.

Table 8a – Detailed tests results.

Test	Positive		Negative Age		Negative Income		Negative Capital		Negative Residency		Negative Contribution Years		All Female Exception		All UK Exception	
Pro	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 8b –Summary results. 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.

Trial	1	2	3	4	5	6	7	8	9	10
Pro1	95.1	95.5	95.1	95.5	96	96.8	96.4	96.3	94.1	94.4
Pro2	94.4	92.6	92.5	92.8	88.9	93.3	94.1	93.3	93.9	93.9

It worth noting that the above experiments were all conducted in game mode between disagreeable agents; the authors have also conducted other experiments including different profiles and different game and strategy modes (all where the proponent uses DS1). The result of these experiments revealed that a minor increase in accuracy could be achieved when the opponent applies a destroy strategy mode (96.01% compared to 95.83% achieved in the previous experiment). On the other hand, the worst possible strategy allocation yields a mere 85.63% accuracy level. In this allocation the proponent applies agreeable destroy strategy in game mode, while the opponent applies agreeable build strategy in win mode. More importantly, we have also investigated which strategies produced the longest dialogues and which produced the shortest. As expected the longest dialogues are obtained when both players employ disagreeable profiles (the average number of rounds when two parties are agreeable is 2.42 rounds) while the shortest dialogues are produced when two players are disagreeable (the average number of rounds when two parties are disagreeable is 8.57 rounds).

7. Conclusion

PADUA provides a means for agents to engage in discussion about a classification on the basis of raw data, unmediated by knowledge representation effort, in order to reach an agreed classification. PADUA necessarily has significant differences from the existing protocols designed to argue about knowledge represented as rules, and the resulting dialogues have a flavour akin to dialogues related to case based reasoning in law, but relying on sets of cases rather than particular decisions. The protocol is particularly applicable to domains in which there are large volumes of data available, but where it would prove unrealistic to 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 [24], which used dialogue to generate a rule based theory, can only be applied to comparatively small datasets. Moreover PADUA is ideal for applications with several distributed datasets generated from different samples, since it can exploit and reconcile any systematic differences in the underlying data available to the dialogue participants. Moreover, it has proved to be adequately robust when the datasets contain a substantial number of misclassified examples.

As it can be viewed as a dialogue game, there is also the question of what strategies and tactics the participants should adopt. Some preliminary work has been done on this [36], where it was shown that the participants can, for example, be represented as cooperative or adversarial. The experiments reported in this paper confirm that different strategies give rise to different flavours of dialogue. Some have the flavour of persuasion dialogues, others of deliberation dialogues, demonstrating how these distinct types of dialogue, identified by Walton and Krabbe [37], can be realised in the same protocol when different strategies are used. Further experiments will explore questions relating to how strategies impact on the quality of decisions and the quality of justifications. In [38] an argumentation framework for learning agents is proposed: this framework is similar to PADUA in taking the experience, in the form of past cases, of agents into consideration and focusing on the argument generation process. Yet, the suggested protocol applies learning algorithms techniques, while PADUA implements simpler association rule mining techniques to produce arguments. Also the protocol in [38] is designed for pairs of agents that collaborate to decide the joint solution of a given problem, while PADUA can be applied in variety of situations including persuasion, deliberation and classification.

An important topic of discussion in recent work on reasoning with cases in law is the notion of intermediate predicates (see [11], [24] and [39]). In [39] the important distinction is made between intermediate predicates which are truth functionally determined by some base level predicates, and those for which there is no simple truth functional relationship. For these latter kinds of intermediate predicates, it may be necessary to first agree their application before deciding the main question. This is accommodated in PADUA through the possibility of nested dialogues, and the improvements gained were illustrated by an example in the section 5.2. While this does require some degree of domain analysis to identify and organize the intermediate predicates, so as to form what is termed in IBP [11] a “logical model” of the domain, this analysis is at a high level and does not require manual analysis of individual cases to apply the intermediate predicates. Once identified, this “logical model” can be used by the control layer of PADUA to set the agenda for the dialogue.

The features of PADUA described above mean that it is particularly applicable to legal domains where very large volumes of cases are processed by lay adjudicators, as in welfare benefits in the UK and elsewhere. These domains do have a large volume of examples and must be expected to contain much misclassification. Moreover since the decisions are made by different adjudicators and in different offices, we can readily identify a rationale for agents with distinct databases, which may be expected to exhibit different characteristics. The error rates experienced in processing claims for welfare benefits present a significant problem, for which a solution is highly desirable. We have proposed an approach to the problem by means of *moderation*: an argumentation dialogue between two agents, each using their own cases. We have reported experimental results which shows that this dialogue will result in reducing the misclassifications in the databases, from 33% in the original data to less that 10%, a performance superior to other association rule classifiers and comparable with decision tree classifiers, and previously reported AI and Law systems, even where they have used only correctly decided cases for training. Moreover we have shown 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 other as proponent, we find that we can reduce the number misclassified on both runs to 0.3%, although there is disagreement in 20% of cases. We suggest that this 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.

Current work is focused on extending PADUA from a two agent dialogue protocol to allow for multiple participants. The multi agent version, PISA, [40], will allow for the moderation dialogue to be conducted with a number of agents. Assuming there is variability in the mistakes made across agents, we believe that this will result in a highly accurate consensus. We will explore this hypothesis in future work.

References

[1] Groothius, M. and Svensson, J. (2000). Expert System Support and Juridical Quality. In Proceedings of Jurix 2000, 1–10. IOS Press: Amsterdam.

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

[3] Getting it right: Improving Decision-Making and Appeals in Social Security Benefits. Committee of Public Accounts. London: TSO, 2104 (House of Commons papers, session 2003/04; HC406).

[4] National Audit Office (2006). International benchmark of fraud and error in social security systems REPORT BY THE COMPTROLLER AND AUDITOR GENERAL | HC 1387 Session 2005-2006 | 20 July 2006

[5] H. Prakken. Formal systems for persuasion dialogue. In The Knowledge Engineering Review, Vol 21, (2006). pp: 163-188

[6] H. Prakken. On dialogue systems with speech acts, arguments, and counterarguments. In Proc. of the 7th European Workshop on Logic for Artificial Intelligence. Springer Lecture Notes in AI, Vol 1919. Berlin: Springer, (2000). pp. 224–238.

[7] J. Mackenzie: Question-begging in non-cumulative systems. In: Journal of Philosophical Logic, Vol 8, (1979). pp: 127-133.

[8] E. L. Rissland, D. B. Skalak and M. T. Friedman: BankXX. A Program to Generate Argument Through Case-Base Research. ICAIL (1993). pp: 117-124.

[9] V. Aleven: Teaching Case Based Argumentation Through an Example and Models. PhD thesis, University of Pittsburgh, Pittsburgh, PA, USA. 1997

[10] K. D. Ashley: Modeling Legal Argument. MIT Press, Cambridge, MA, USA, 1990.

[11] K. D. Ashley and S. Brüninghaus. A Predictive Role for Intermediate Legal Concepts. In Proc. of Jurix 2003, IOS Press: Amsterdam (2003). pp 153-162

[12] M Minsky and S. Papert: Perceptrons: An Introduction to Computational Geometry. Cambridge MA: MIT Press, (1969).

[13] R. Agrawal, T. Imielinski and A.N. Swami: Mining Association Rules between sets of items in large databases. In: Proc. of Int. Conf. on Management of Data, (ACM SIGMOD 93), Washington, (1993). pp: 217-216.

[14] F. P. Coenen, P. Leng and G. Goulbourne: Tree Structures for Mining Association Rules. In: Journal of Data Mining and Knowledge Discovery, Vol 8, No 1, (2004). pp: 25-51.

[15] G. Goulbourne, F. P. Coenen and P. Leng: Algorithms for Computing Association Rules Using a Partial- Support Tree. In: Proc. of ES99, Springer, London, UK, (1999). pp: 142-147.

[16] L. Amgoud, S. Belabbès, and H. Prade: A formal general setting for dialogue protocols. In: Proc. Of AIMSA'06 12th Int. Conf. on Artificial Intelligence: Methodology, Systems, Applications, Varna, Bulgaria, (2006). pp: 14 - 15.

[17] D. Moore: Dialogue game theory for intelligent tutoring systems. PhD thesis, Leeds Metropolitan University (1993).

[18] L. Amgoud, and N. Maudet. Strategical considerations for argumentative agents (preliminary report). In Proc. of 9th Int. Workshop on Non-Monotonic Reasoning (NMR'02). Toulouse, France, (2002). Special session on Argument, Dialogue, Decision. pp. 409-417.

[19] A.C. Kakas, N. Maudet and P. Moraitis. Flexible Agent Dialogue Strategies and Societal Communication Protocols. In 2nd Int. Workshop on Argument in Multi Agent Systems (ArgMAS'04), Springer LNCS 3366 (2005).

[20] N.Oren, T. J. Norman and A. Preece. Loose Lips Sink Ships: a Heuristic for Argumentation. In Proc. of 3rd Int. Workshop on Argumentation in Multi- Agent Systems(ArgMAS'06), Hakodate, Japan, (2006). pp 121-134.

[21] L. Amgoud and S. Parsons. Agent dialogues with conflicting preferences. In Proc. of 8th Int. Workshop on Agent Theories, Architectures and Languages. Seattle, Washignton, (2001). pp 1-15

[22] T.J.M. Bench-Capon: Knowledge Based Systems Applied To Law: A Framework for Discussion. In: T.J.M. Bench-Capon (ed), Knowledge Based Systems and Legal Applications, Academic Press, (1991). pp: 329-342.
[23] T. J. M. Bench-Capon. Neural Nets and Open Texture. In 4th Int. Conf. on AI and Law. ACM Press: Amsterdam, (1993). pp: 292–297.
[24] A. Chorley, T. J. M. Bench-Capon. AGATHA: Using heuristic search to automate the construction of case law theories. Artif. Intell. Law Vol 13(1), (2005). pp: 9-51
[25] K. Atkinson and T. J. M. Bench-Capon. Legal Case-based Reasoning as Practical Reasoning. Artif. Intell. Law Vol 13(1), (2005). pp: 93-131
[26] F. Coenen and P. Leng. Obtaining Best Parameter Values for Accurate Classification. In Proc. ICDM'05, IEEE, (2005). pp: 597-600.
[27] F. Coenen , P. Leng, and L. Zhang. Threshold Tuning for Improved Classification Association Rule Mining. In Proc. PAKD'05 , LNAI3158, Springer (2005). pp: 216-225.
[28] F. Coenen , P. Leng and S. Ahmed. Data Structures for association Rule Mining: T-trees and P-trees. In IEEE Transactions on Data and Knowledge Engineering, Vol 16(6), 2004. pp: 774-778.
[29] F. Coenen , P. Leng and G. Goulbourne. Tree Structures for Mining Association Rules. Journal of Data Mining and Knowledge Discovery, Vol 8(1), 2004. pp:25-51.
[30] B. Liu, W. Hsu and Y. Ma. Integrating Classification and Association Rule Mining. In Proc. of Int. Conf. on Knowledge Discovery in Databases (KDD'98), New York, AAAI (1998). pp: 80-86.
[31] W. Li, J. Han and J. Pei. CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules. In Proc. Int. Conf. on Data Mining (ICDM`01), (2001). pp: 369-376.
[32] J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, (1998).
[33] T.M. Mitchell. Machine Learning. McGraw-Hill. (1997).
[34] K. D. Ashley and S. Brüninghaus. A Predictive Role for Intermediate Legal Concepts. In Proc. of Jurix 2003, IOS Press: Amsterdam (2003). pp: 153-162.
[35] M. Mozina, J. Zabkar, T. Bench-Capon and I. Bratko. Argument based machine learning applied to law. In Journal Artificial Intelligence, Vol **13** (1), (2005). pp: 53–73.
[36] M. Wardeh, T. J. M. Bench-Capon and F. P. Coenen: PADUA Protocol: Strategies and Tactics. In Proc. ECSQARU , 10th European Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, LNAI 4724, (2007), 465-476.
[37] D. N. Walton and E. C. W. Krabbe: Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. SUNY Press, Albany, NY, USA, (1995).
[38] S. Ontañón and E. Plaza. Arguments and Counterexamples in Case-Based Joint Deliberation. In ArgMAS Hakodate, Japan (2006). pp 36-53.
[39] L. Lindahl and J. Odelstad: Normative positions within an algebraic approach to normative systems. In Journal of Applied Logic, Vol 17(2), (2005). pp: 63:91.
[40] M. Wardeh, T. J. M. Bench-Capon and F. P. Coenen. PISA - Pooling Information from Several Agents: Multiplayer Argumentation from Experience. In Proc. AI'2008, Springer, London, pp: 133-146.

Table Legends

Table 1 – Effect Rules

Table 3 – Possible Moves Preferences

Table 2 – The protocol legal next moves

Table 4 - Best move content tactics

Table 5 – Example Dialogue

Table 6 – Experiment 1 results.

Table 7a - Comparison with DS1

Table 7b - comparison with DS2

Table 8a – Detailed tests results.

Table 8b –Summary results.