Using BATNAs and WATNAs in Online Dispute Resolution

Francisco Andrade[1], Paulo Novais[2], Davide Carneiro[2], John Zeleznikow[3] and José Neves[2]

[1] Escola de Direito, Universidade do Minho, Braga, Portugal
fandrade@direito.uminho.pt
[2] DI-CCTC, Universidade do Minho, Braga, Portugal
{pjon, dcarneiro, jneves}@di.uminho.pt
[3] School of Management and Information Systems, Victoria University, Melbourne, Australia
john.zeleznikow@vu.edu.au

Abstract. When contracting through software agents, disputes will inevitably arise. Thus there is an urgent need to find alternatives to litigation for resolving conflicts. Methods of Online Dispute Resolution (ODR) need to be considered to resolve such disputes. Having agents understanding what the dispute is about, managing all interaction between the parties and even formulating proposed solutions is an important innovation. Hence it is of the utmost relevance that the agents may be able to recognise and evaluate the facts, the position of the parties and understand all the relevant data. In many circumstances, risk management and avoidance will be a crucial point to be considered. In this sense we analyze the usefulness of a parallel concept to BATNA – Best Alternative to Negotiated Agreement, that of a WATNA – Worst Alternative to Negotiated Agreement, allowing the software agents to consider the space between BATNA and WATNA as a useful element to be taken into account when making or accepting a proposal. These software agents embodied with intelligent techniques are integrated in an architecture designed to provide support to the ODR in a system we have developed for the resolution of labour disputes - UMCourt. In this context software agents are used to compute and provide the parties with the best and worst alternative to a negotiated agreement.

Keywords: On-Line Dispute Resolution, Negotiation, BATNA, WATNA

1 Introduction

When moving to a global information society, new needs have appeared in the field of dispute resolution, since disputes can now take place between virtually any two entities in the world. With the integration of new communication technologies into our daily lives, traditional Alternative Dispute Resolution (ADR) mechanisms including mediation, conciliation, negotiation or modified arbitration and jury proceedings ([10] and [30]) have slowly started to adapt, giving birth to what is now known as Online Dispute Resolution (ODR).

ODR allows for the moving of already traditional alternative dispute resolution methods "from a physical to virtual place" [3]. This provides the parties with an easier

course than litigation, for dealing simply and efficiently with disputes, saving both "temporal and monetary costs" [12]. This new model for dispute resolution aims at being an online alternative to litigation and traditional ADR. It can expand the possibilities of common ADR systems as, with the introduction of entities with enhanced abilities, increases the generation of solutions and the possible ways of achieving them.

Techniques for developing ODR systems include legal knowledge based systems that provide legal advice to the disputing parties and also "systems that (help) settle disputes in an online environment" [6]. In this sense we can enumerate projects that make use of rule-based systems such as [25], negotiation support systems as in [26], [27] and [28], and others that look at game theory and heuristics [29]. In this paper, we consider the use of a Case-based Reasoning (CBR) [1] approach for the purpose of retrieving similar cases in order to advise the parties about the probable and possible outcomes and solution paths given former similar cases.

The so-called second generation of ODR systems is essentially defined by a more active role of technology [16]. It goes beyond putting the parties into contact and is used for idea generation, planning, strategy definition and decision making processes. The technologies used in this new generation of ODR systems will comprise not only the communication technologies used nowadays but also subfields of areas such as Artificial Intelligence, mathematics or philosophy: neural networks, intelligent agents, case-based reasoning, logical deduction, argumentation, methods for uncertain reasoning and learning methods. Thus being, the development of Second Generation ODR, in which an ODR system might act "as an autonomous agent" [16] is an appealing way for solving disputes.

In considering this possibility, we take in consideration the Katsh/Rifkin vision of the four parties in an ODR process: the two opposing parties, the third party neutral and the technology that works with the mediator or arbitrator [11]. But we must assume a gradual tendency to foster the intervention of software agents, acting either as decision support systems [3] or as real electronic mediators [16]. This latest role for software agents implies the use of artificial intelligence techniques such as case based reasoning and information and knowledge representation. "Models of the description of the fact situations, of the factors relevant for their legal effects allow the agents to be supplied with both the static knowledge of the facts and the dynamic sequence of events" [16].

Merely representing facts and events, whilst useful, is not sufficient for dispute resolution; the software agent, in order to perform actions of utility for the resolution of the dispute, also needs to know not only the terms of the dispute but also the rights or wrongs of the parties [16], and to foresee the legal consequences of the said facts and events. Thus we have to consider the issue of software agents really understanding law and to consider legal reasoning by software agents and its eventual legal responsibility: As [4] states, "are law abiding agents realistic?".

We need to consider whether agents can evaluate the position of the parties and present them with useful proposals, "taking into a consideration of which of the two parties would have a higher probability of being penalised or supported by a judicial decision of the dispute and, therefore, who would be more or less willing to make concessions in their claims" [16]. The ability to understand the position of the parties is vital for the successful involvement of software agents in the process. To do so, it is

mandatory for the software agent to have the characteristics of consistency, transparency, efficiency and enhanced support for dispute resolution, in order to allow it to replicate "the manner in which decisions are made" and thus make the parties "aware of the likely outcome of litigation" [3]. That is to say, software agent intervention in an ODR procedure should take into account the alternatives, for the parties, to an ODR negotiated agreement. This kind of ODR environment involves much more than just transposing ADR ideas into ODR environments. It should actually proceed by being "guided by judicial reasoning", and getting disputants "to arrive at outcomes in line with those a judge would reach" [14]. Despite there being difficulties to overcome, the generalised use of software agents as decision support systems in a negotiation, is nevertheless a useful approach.

2. The Role of BATNA (Best Alternative to a Negotiated Agreement)

Principled negotiation is based on four fundamental principles: separate the people from the problem; focus on interests, not positions; invent options for mutual gain; insist on objective criteria [8, 21]. In interest-based negotiation, the disputants attempt to reconcile their underlying interests. Most negotiations are interest-based[1]. In this situation, disputing parties need to know their BATNA (or, the possible best outcome "along a particular path if I try to get my interests satisfied in a way that does not require negotiation with the other party"[15].

Whilst principled negotiation as an important concept, it must be supplemented with other approaches to negotiation. Justice or rights based negotiation (pointing out to the determination of who is – or who could be considered to be – right in accordance to norms or rules of behaviour) – should also be considered [8].

When taking a principled negotiation approach, we must understand the notion of a BATNA and what role it should play in ODR. "A precise notion of what constitutes a BATNA is not available" [6]. But "knowing one's BATNA may contribute to the acknowledgement that an agreement may be disadvantageous" [12].

Entering into negotiation or mediation is justified if the parties expect to get better results than those that could be obtained without the process. In order to evaluate this, one needs to know, at least what the best alternative to the negotiated agreement would be. Of course, parties will tend to enter into an agreement if they know that a possible settlement in ODR is undoubtedly better than her own BATNA [12]. This is an obvious case of interest in knowing one's BATNA. But the position of the parties may become much more unclear if they cannot foresee the possible results in case the negotiation / mediation fails. "If you are unaware of what results you could obtain if the negotiations are unsuccessful, you run the risk of entering into an agreement that you would be better off rejecting or rejecting an agreement that you would be better off entering into" [9]. That is to say, the parties, by determining their BATNA, would

---

[1] One exception is Australian Family Law where the paramount interests of the children trump the interests of the divorcing parents. This is however not the case in US family law.

on one side become "better protected against agreements that should be rejected" and, on the other side, they would be in a better condition to "reach an agreement that better satisfies their interests" [6].

A BATNA may also provide additional interesting features for the parties in the dispute procedure. For instance, it might also be used as a "way to put pressure on the other party", especially in dispute resolution procedures allowing the choice of going to court [6]. The important thing is that the choice of going to court, instead of continuing ADR or ODR, should be a "well-informed choice". And in ODR environments, either by the use of data mining techniques, semantic web technology or other adequate techniques possibly used to determinate the BATNA, the parties can foresee the possible outcome of the judicial dispute in the case of not reaching an agreement through ODR [3]. For that purpose, some technical possibilities have already been pointed out in literature. For instance, the use of a BATNA agent, an agent that has the knowledge necessary to compute the value of the BATNA, using Toulmin argument structures providing a "mechanism for decomposing a task into sub-tasks" has been pointed out [2]. Similarly, the possibility of the BATNA agent being modified in order to "include current case data and incorporate changes to law" is an important development [23]. The role of technology is becoming more appealing especially for the task of determining or establishing objective BATNAs [6].

3. How understanding WATNAs can improve the ODR process

No matter the alternative dispute resolution method chosen, parties will tend "to develop an overly optimistic view on their chances in disputes" [6]. This rather optimistic view may lead to differing attitudes taken by the parties, especially in their calculation of chances of success obtaining their goals in the dispute, and influencing the way disputants calculate their BATNA. In the course of the dispute, the parties may tend either to reject generous offers from the other parties, or to stand stubbornly fixed in some positions or even support "positions or options that are incorrect" [6]. This "optimistic overconfidence" [6] may lead the parties to miscalculate the possibilities of success in an eventual judicial decision.

It is important to reflect on the usefulness of the concept of a BATNA. On one side, a BATNA may be misevaluated through the above optimistic overconfidence of the parties. On the other side, there is no probabilistic measure for the correctness of BATNA. That is to say, the best alternative may not be the most probable one. And parties will certainly tend to underestimate the probabilities of an undesired result in judicial decision-making.

In many situations, the calculation of the possible outcomes of a judicial decision may become quite complex. One of the major reasons that disputants try to avoid litigation is the risks they might incur – in terms of legal costs and outcome – if they are unsuccessful [30]. In this situation it could certainly be useful, besides the BATNA, to consider a WATNA (Worst alternative to Negotiated Agreement) [8, 15, 20]. A WATNA intends to estimate the worst possible outcome along a litigation path [15]. It can be quite relevant in complementing principled negotiation with a justice or rights based approach and thus leading to a calculation of the real risks that

parties will face in judicially determined litigation, imagining the worst possible outcome for the party. This calculation could prove interesting both for ADR and for ODR. In the case of second generation ODR, it would be useful to develop a software agent to consider the whole space between the BATNA and WATNA. The larger the space, the greater the benefit in making, or accepting, a proposal.

Consider, for instance, from an organizational perspective, the relevance of labour rules in the functioning of professional virtual communities: as Willy Picard states, "the organizational structure of the population that may potentially execute activities may evolve, as some employees are promoted or are fired" [17]. In the case of a employee being fired, litigation will most likely occur. Under legal systems such as that of Portugal, a huge deal of legal parameters need to be considered:

(a) the antiquity of the worker in the company,

(b) supplementary work,

(c) night work,

(d) justified or unjustified absence to work,

(e) the possibility of a "just cause for dismissal" being declared by Court,

(f) the existence (or not) of a valid and legal procedure of dismissal,

(g) the possibility of dismissal being accepted without indemnities or

(h) of it being accepted but accompanied by indemnities that could range from a very low to a very high amount of money [7].

To dismiss a worker, the company needs to calculate the potential ensuing financial penalties. For the worker, the amounts involved are not irrelevant: being fired without good indemnities may be seen as a double sacrifice: not only would he lose his job but he could get no or little payment for his loss. But he might, on the other hand, receive adequate financial compensation. For the parties in a labour conflict, it can be said that the calculation of the possible results of litigation (or of the various possible outcomes for litigation) are vital.

In order to clearly understand the advantages of a proposed agreement, parties need to know not only their BATNA but also their WATNA (the worst alternative they may obtain in case they do not reach an agreement), and they certainly should consider the spectrum between their BATNA and their WATNA. Of course, the less space there is between BATNA and WATNA, the less dangerous it becomes for the party not to accept the agreement (unless, of course, their BATNA is really disadvantageous). A wider space between BATNA and WATNA would usually mean that it can become rather dangerous for the party not to accept the ODR agreement (except in situations when the WATNA is not undesirable for the party).

Of course, this consideration of the values appearing between the BATNA and the WATNA is related to the Zone of Possible Agreement proposed by Raiffa (1982) [19]. It is the zone where an agreement can be met that is acceptable to both parties. The consideration of the space between BATNA and WATNA has, in our vision, a clear risk oriented approach – the intention is to estimate the risks and, thus, to avoid them. And this vision may well push the possible agreement to a space not exactly coincident with the traditional ZOPA. And certainly it can even be considered here the existence of a MLATNA – most likely alternative to a negotiated agreement [20].

In terms of the system we are developing, it does not much matter what is the most likely outcome, which might be hard to estimate, but rather it is vital to foresee the real risks that the parties are facing. And the extreme value presented by WATNA

may well force the parties to change the ideas they have about their BATNA and ZOPA.

We accept that this analysis is still in an early stage and that other relevant parameters should also be considered: for instance, the existence of metrics in order to measure the probabilities of each possible outcome. Nonetheless, judicial decisions, although having to be based on legal rules and reasoned from them, arise from a process in which it must be determined that some issues are true or false, or are considered as proved, partially proved, or not proved [18]. This characteristic of judicial decisions certainly makes it advisable for parties to consider not just a single value, in the case of judicial litigation, but rather a spectrum of values, situated between a BATNA and a WATNA.

4. UMCourt Architecture

On-line dispute resolution methods can provide easy, efficient, fast ways for resolving disputes. Labour disputes need to be quickly resolved. The judicial path (which, in countries such as Portugal, often leads to a judicial conciliation led by a Judge) is expensive and time consuming. First and second generation ODR [16], with agents performing relevant parts of the agreement procedure can be of inestimable use for the parties in a Portugese labour dispute.

UMCourt is a project being developed at University of Minho in the context of the TIARAC project (Telematics and Artificial Intelligence in Alternative Conflict Resolution) that intends to help parties involved in legal disputes. The current application relates to the domain of Portuguese labor law. UMCourt represents one of the first steps in Portugal to implement the ideas depicted in the previous sections [5].

It is based on the agent paradigm, which means that the resulting architecture is highly modular and expansible. This choice has not been a random one. Although we are currently addressing the specific domain of the Portuguese labour law, we are aware that by defining a few domain-dependent agents and reusing many of the core agents, it is possible to extend the platform to address other domains. To put this idea into practice, an extension to this architecture that addresses consumer protection law is now being developed, that uses much of the already defined architecture. The building blocks of this modular architecture are agents or groups of agents with well defined roles that, through their interactions, configure an intelligent system. Following the methodology proposed by [22], our work in this system began with a high level definition of the members of the architecture in terms of their roles. In this phase we have looked at existing agent-based architectures in the legal domain, namely at [2], and made the necessary improvements in order to adapt it to our needs. We therefore arrived at a configuration of four high-level agents with their roles shown in Table 1. The implementation of this architecture is based on a range of technologies chosen with the objective of making it a distributed, expansible and independent one [5].

The core of the architecture is the Jade (Java Agent Development Framework) platform[2]. JADE is a software framework that significantly facilitates the development of agent-based applications in compliance with the FIPA specifications. FIPA (Foundation for Intelligent Physical Agents) promotes standards that aim at the interoperability and compatibility of agents, specifically targeting the fields of agent communication, agent transport, agent management, agent architecture and applications. Of these FIPA categories, agent communication is the core category at the FIPA multi-agent system model and is the one that is more closely followed in our system. Our interest in this category is focused on specification 61, which defines the structure of the Agent Communication Language (ACL), .i.e., the structure that the messages exchanged between agents respect. An example of use of this standard is shown below.

Example of an ACL message from agent Coordinator to agent Retriever requesting the cases similar to 1263491000923, assuming the default settings.

```
Sender : ( agent-identifier
  :name Coordinator@davide-desktop:1099/JADE
  :addresses (sequence http://davide-desktop:7778/acc ))
Conversation-ID : 1263492569251
Reply-To : Coordinator@davide-desktop:1099/JADE
Ontology : CBR_LABOUR
Content : RETRIEVE_SIMILAR DEFAULT 1263491000923
```

Jade has also the advantage of dealing with all the issues of message transport and agent registry thanks to a wide number of services provided by the *ams* and *df* agents. This significantly simplifies the creation of new agents thus decreasing the development time and costs.

Table 1. The four high-level agents and their main roles.

High-level Agent	Description	Main Roles
Security	This agent is responsible for dealing with all the security issues of the system	Establish secure sessions with users
		Access levels and control
		Control the interactions with the knowledge base
		Control the lifecycle of the remaining agents
Knowledge Base	This agent provides methods for interacting with the knowledge stored in the system	Read information from the KB
		Store new information in the KB
		Support the management of files within the system
Reasoning	This agent embodies the intelligent mechanisms of the system	Compute the BATNA and WATNA values
		Compute the most significant outcomes and their respective likeliness
		Proactively compile and provide useful information based on the phase of the dispute resolution process

[2] See http://jade.tilab.com/ last accessed January 2 2010.

Interface	This agent is responsible for establishing the interface between the system and the user in a intuitive fashion	Define a intuitive representation of the information of each process
		Provide an intuitive interface for the interaction of the user with the system
		Provide simple and easy access to important information (e.g. laws) according to the process domain and phase

The interaction with the system can be performed in two ways: by means of a JSP based Guided User Interface (GUI) or with remote agents that interact directly with the agent platform. In the first case, the interface was designed so that the users could remotely interact with the system using any common web browser. Through the browser, the client sends requests to the server which interacts with the Jade platform, collects the answers and returns the HTML code to be shown in the browser of the user. By doing this, we not only make sure that the user can understand and interact with the system through an intuitive interface but also grant the security of the whole system. In the second case, Jade agents external to the platform can interact with the agents present by means of FIPA-ACL messages. In fact, using the Jade platform, sending and receiving remote messages becomes as easy as performing the task locally. This increases the expansibility and compatibility of the architecture, making sure that it can interact with other architectures with similar or complementary functionalities or even that automated agents representing the parties can interact.
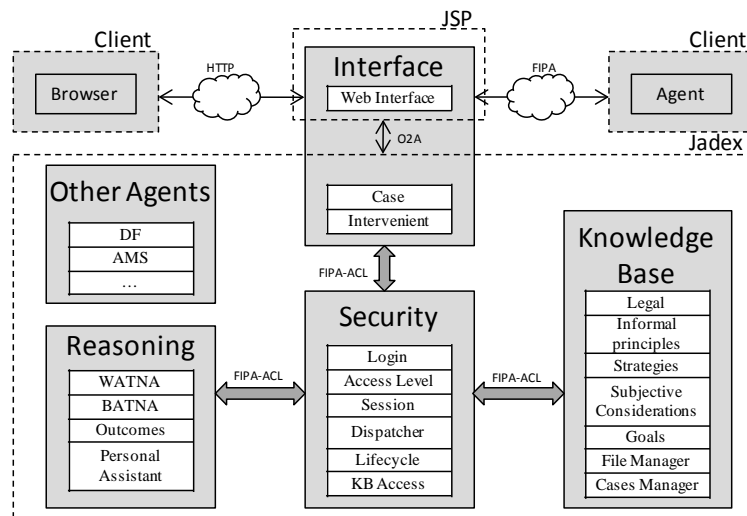

Fig. 1: A simplified view of the architecture.

The high level agents depicted in Table 1 have been submitted to a cut-down process in order to more precisely define their roles and make them more simple and refined. In this task we have defined several simpler agents, such as the coordinator, with the task of load balancing, the retriever which interacts with the KB agents in order to retrieve cases, among others. A simplified view of the architecture highlighting the main agents is presented in Figure 1. However, as this paper is centred on the concepts of BATNA and WATNA and the determination of the space

in between, we will from now on focus on the *Reasoning* agent. The *Reasoning* agent was defined with some objectives in mind, being one of them to assist the parties in determining the possible outcomes. When the parties have knowledge about what could possibly happen, they can take more informed and, hopefully, rational decisions. As exposed before, in this paper we support the idea that it is vital to know the BATNA and WATNA values, as well as the most significant values contained in the interval between them, so that an optimal global solution can be achieved. Indeed, the WATNA of one party is frequently close to the BATNA of the other so limiting the possible choices to these two values represents a serious drawback and the impossibility of reaching a global optimization. The optimal solution lies somewhere in the intersection of the possible solutions of each of the parties: the Zone of Possible Agreement (ZOPA) [13]. This zone represents all the solutions that can happen at the end of the process. The role of the *Reasoning* agent is, in the first place, to determine if an agreement is possible. If the ZOPA does not exist, then an agreement is not possible. However, if there is a ZOPA with a range of solutions, the objective of the agent is to determine which is the best option and suggest it to the parties, and then trying to work for a mutually favourable outcome from that point on. This agent has been subdivided into four simpler agents: *BATNA, WATNA, Outcomes* and *Personal Assistant*.

The *BATNA* and *WATNA* agents respectively compute the values of the BATNA and the WATNA. This calculus is based on mathematical formulae that are well defined in the Portuguese labour law and have been transported to the agents in the form of logical predicates. These are simple formulae that map a set of values of parameters such as antiquity or extra and night working hours to an economical value of indemnity. The first agent therefore analyzes the case and computes the value of the best legally possible outcome, i.e., it assumes that the employee is absolutely right in his allegations. In the other hand, the *WATNA* agent does the same, assuming that the employee cannot prove any of the arguments in his defence, determining the worst possible case according to the rules of the Portuguese labour law. The output of these two agents, according to what is defined in Portuguese labour law, is a pair of values: one quantifying the value of the indemnity and the other one stating if the employee looses the job or is reintegrated.

The *Outcomes* agent has as objective, as the name depicts, to compute the possible outcomes of a new case, which configures one important feature in online dispute resolution systems. For determining them, this agent uses a Case-based Reasoning model. Our conviction that CBR is an appropriate method for such a problem solving domain relies on the fact that law itself implements a very similar concept: the legal precedent [24]. This concept is defined by the Blacks Law dictionary as "an adjudged case or decision of a court, considered as furnishing an example or authority for an identical or similar case afterwards arising or a similar question of law."[31] A precedent, in the legal domain, can be sub-divided into two categories: the binding precedent which must be applied and the persuasive precedent, which is not mandatory but is relevant. This labeling has generally to do with the courts that decide on the case. If it is a higher court making the decision, it usually becomes a more persuasive precedent. By looking at all the significant past cases contained in the Knowledge Base, this agent is able to determine which outcomes are possible to occur, given the properties of the current case.

In order for this agent to correctly use the information contained in each case, a syntactic structure for the case has been defined. Each case contains the laws used by each party and all the remaining information that is requested by law, including a pair of values denoting the outcome: one quantifying the value of the indemnity and the other one stating if the employer looses the job or is reintegrated. All this data is stored in XML files but, in order to fasten the retrieval processes, the cases are indexed in a database by the laws that they address and the way that they are addressed, i.e., if these laws are used by the employee, employer or by a witness. This allows us to efficiently search for a case with given characteristics in the database and then retrieve it from its location in the file system to parse all the information.

Essentially, the process of estimating outcomes is as follows. The agent looks at the new case, specifically at the norms that are addressed by each party. Afterwards, it applies a template retrieval algorithm in order to narrow the search space. This can be performed, a priori in determining which type of cases have the possibility of being similar and which ones do not. In this sense, template retrieval works much like SQL queries: a set of cases, with given characteristics, is retrieved from the database. In the next step, a nearest neighbor algorithm is applied to this set of cases instead of applying it to all the cases in the case memory, a task that could be very time consuming as our nearest neighbor algorithm has linear complexity (Formula 1).

$$\frac{\sum_{i=1}^{n} W_i * fsim_i(Arg_i^N, Arg_i^R)}{\sum_{i=1}^{n} W_i} \tag{1}$$

In formula 1, our closest neighbor algorithm is shown. In this equation,
- n – number of elements to consider to compute the similarity;
- W_i – weight of element i in the overall similarity;
- Fsim – similarity function for element i;
- Arg – arguments for the similarity function representing the values of the element i for the new case and the retrieved case, respectively N and R.

We now discuss in greater detail the information of the case that is considered to be relevant for the computation of the similarity, i.e., the components. According to the scope of application, we consider three types of information: the objectives stated by each party in the beginning of the dispute, the norms addressed by each party and by the eventual witnesses and the date of the dispute. The norms addressed and the objectives are lists of elements, thus the similarity function consists in comparing two lists (equation 2). The similarity is higher when the two lists have a higher percentage of common members. As for the date, the similarity function verifies if the two dates are within a given time range, having a higher similarity when the two dates are closer.

$$fsim_{list} = \frac{|L_N \cap L_R|}{n}, n = \begin{cases} |L_N|, & |L_N| \geq |L_R| \\ |L_R|, & |L_N| < |L_R| \end{cases} \tag{2}$$

Once each case is associated with a value that denotes its similarity with another given case, we can perform more interesting and useful operations on the cases. One

of these operations is to determine to which extent we want the cases to be similar. We do so by selecting a threshold. If our current selection results in few cases, we can lower the threshold resulting in a wider number of cases but with an expected smaller degree of similarity. On the other hand, if we have many cases, we can increase the threshold in order to get a more restricted set with a higher degree of similarity.

The most interesting aspect is that the agent can autonomously apply these operations in real-time, when choosing the cases to present to the user: if there are many cases, the agent will increase the threshold to select less and more similar cases and vice versa. For the agent to determine if it should change the value of the threshold, it generates a box-and-whisker diagram using the values of the indemnities of each case selected, and looks at the dispersion of the data. The dispersion of the data is determined by the Euclidian distance between the indemnities of each of the two consecutive cases. The agent decides to decrease the value of the threshold if the data is much dispersed and the other way around if the data is not very dispersed.

Having done this, the agent sorts the cases for each of the parties, starting with the WATNA, passing through all the intermediary cases and ending with the BATNA. The cases are sorted according to the numeric value of the indemnity and how favorable it is to each party. At this point, the parties have an intuitive picture of what may happen, including not only the best and worst case but all the intermediary cases that have happened in the past and may happen again, accompanied by the respective likeliness to occur. All this information is shown in Figure 2. This figure contains two axes, one for each party, with a direction for increasing satisfaction. Cases are here represented in these axes by the smaller rectangles and the Euclidian distance between the values of their indemnities determines how they are distributed and ordered in the axis of each party and therefore highlights the dispersion.
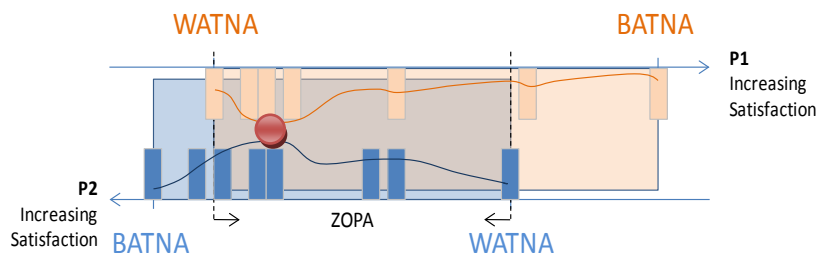

Fig. 2: The graphical representation of the possible outcomes for each party.

The likeliness of a given outcome is represented in Figure 2 by the colored curves which denote the area in which the cases are more likely to occur. We can see that the line is more distant from the axis when there are more cases that are concentrated. This denotes a higher likeliness for a case in this region. However, it is not only the amount of cases that is important. We also consider the type of case, i.e., if it is a case with a binding or a persuasive precedent or if it has been decided by a higher or a lower court. In some cases, we may even have groups of cases instead of single cases, as cases which are highly similar are grouped together into a single case with a weight that is proportional to the number of cases merged.

Still looking at Figure 2, we can see the range of possible outcomes for each of the parties in the form of the two big colored rectangles and the result of its intersection, the ZOPA. It is also possible to see each case and its position in the

ordered axis of increasing satisfaction, in the shape of the smaller rectangles. As stated earlier, the cases are more likely to occur for each party when they are in the area where the colored lines are further away from the axis of that party. This is highlighted in the figure by the big dot. Therefore, the probable outcome of the dispute will probably be near the area where the two lines are closer. Looking at this information, the parties can have an approximate notion of the most likely outcome. Although a single solution is proposed by the system, parties can look at these cases in order to search for alternative solutions in search for a mutual agreement.

Finally, the *Personal Assistant* agent is responsible for knowing the role of the user and adapting the results of the remaining agents according to that role, i.e., the employee will see the information in a different way than the employer or even than the witnesses. This agent will also be extended with more features, namely the adaptation of the interfaces and the remaining information that is presented to the users, including help information, according to the roles.

5. Conclusions

When using software agents in electronic contracting the possibility of disputes is prevalent. Thus there is a demand for fast and efficient ways for resolving the eventual disputes using ODR. For a second generation ODR, software agents are a useful tool to help the parties reach an agreement. The acceptance of a certain proposal by the parties in ODR must take into consideration relevant parameters, such as the BATNA.

Parties in ODR tend to adopt an over-optimistic view on the possible outcomes in the case of litigation. In many situations the calculation of the possible outcomes of litigation may become rather complex, with a huge range of possibilities to be evaluated. This is particularly clear in Portuguese labour legal cases. In these situations it may become interesting to consider not just the value of BATNA but also the value of the WATNA.

It is vital, to consider the space that lies between the BATNA and the WATNA. It would be advisable for parties to consider not just a single value but rather a spectrum of values, situated between a BATNA and a WATNA. For this purpose, we have developed an architecture supported by a JADE platform, allowing the user to interact with the system. Software agents are used to compute and provide the parties with the best and worst alternative to a negotiated agreement as well as the spectrum of possible outcomes between these two values and their likeliness.

The system presented here, by being based on the multi-agent paradigm ensures that it can be easily extended with the addition of new agents. To address the challenges of incorporating new agents, we rely on open standards and technologies. Moreover, by adopting a case-based approach, we achieve a system that can learn and

adapt to new situations and changes in the law. In future work we intend to allow the system to estimate outcomes of dispute resolution processes based on CBR and other paradigms (e.g., neural networks, Adjusted Winner algorithm) in order to compare the performance and results of different approaches.

References

1. Aamodt, A., Plaza, E., Case-based reasoning: Foundational issues, methodological variations, and system approaches. In: AI Communications, 7(1), IOS Press (1994) 39-59.
2. Abrahams, B., Zeleznikow, J., A multi-agent architecture for online dispute resolution services. Expanding the horizons of ODR. Proceedings of the 5th International Workshop on Online Dispute Resolution (ODR Workshop'08), Firenze, Italy (2008) 51-61.
3. Bellucci, E., Lodder, A., Zeleznikow, J., Integrating artificial intelligence, argumentation and game theory to develop an online dispute resolution environment", ICTAI-2004 - 16th IEEE International Conference on Tools with Artificial Intelligence (2004) 749-754.
4. Brazier, F., Kubbe, O., Oskamp, A., Wijngaards, N., Are Law abiding agents realistic?. Proceedings of the workshop on the Law of Electronic Agents (LEA 2002), CIRSFID, University of Bologna (2002) 151-155.
5. Carneiro, D., Novais, P., Andrade, F., Zeleznikow, J., Neves, J., The Legal Precedent in Online Dispute Resolution, in Legal Knowledge and Information Systems, ed. Guido Governatori (proceedings of the Jurix 2009 - the 22nd International Conference on Legal Knowledge and Information Systems, Rotterdam, The Netherlands), IOS press, ISBN 978-1-60750-082-7, (2009) 47-52.
6. De Vries BR., Leenes, R., Zeleznikow, J., Fundamentals of providing negotiation support online: the need for developping BATNAs. Proceedings of the Second International ODR Workshop, Tilburg, Wolf Legal Publishers (2005) 59-67.
7. Fernandes, AM., Direito de Trabalho, Almedina (2005) (*in Portuguese*).
8. Fisher, R., Ury, W., Getting To Yes: Negotiating Agreement Without Giving In. Boston: Houghton Mifflin. ISBN 0-395-31757-6 (1981).
9. Goldberg, S.B., Sander, F.E., Rogers, N., Cole, S.R., Dispute Resolution: negotiation, mediation and other processes, Aspen Publishers, New York (2003).
10. Goodman, J.W., The pros and cons of online dispute resolution: an assessment of cyber-mediation websites, in Duke Law and Technology Review (2003).
11. Katsh, E., Rifkin, J., Online dispute resolution – resolving conflicts in cyberspace. Jossey-Bass Wiley Company, San Francisco (2001).
12. Klaming, L., Van Veenen, J., Leenes, R., I want the opposite of what you want: summary of a study on the reduction of fixed-pie perceptions in online negotiations. "Expanding the horizons of ODR", Proceedings of the 5th International Workshop on Online Dispute Resolution (ODR Workshop'08), Firenze, Italy (2008) 84-94.
13. Lewicki, R., Saunders, D., Minton, J., Zone of Potential Agreement, In Negotiation, 3rd Edition. Burr Ridge, IL: Irwin-McGraw Hill (1999).
14. Muecke, N., Stranieri, A., Miller, C., The integration of online dispute resolution and decision support systems. Expanding the horizons of ODR, Proceedings of the 5th International Workshop on Online Dispute Resolution (ODR Workshop'08), Firenze, Italy (2008) 62-72.
15. Notini, J., Effective Alternatives Analysis In Mediation: "BATNA/WATNA" Analysis Demystified, (http://www.mediate.com/articles/notini1.cfm), 2005.Accessed July 24 (2009).
16. Peruginelli, G., Chiti, G., Artificial Intelligence in alternative dispute resolution. Proceedings of the Workshop on the law of electronic agents – LEA (2002).
17. Picard, W., Support for Power in adaptation of social Protocols for Professional Virtual

Communities. Establishing the Foundation of Collaborative Networks, Camarinha-Matos L. Afsarmanesh H., Novais P., Analide C., (Eds), Springer-Verlag, Series: IFIP International Federation for Information Processing, ISBN: 978-0-387-73797-3 (2007) 363-370.

18. Pimenta, J.C., A Lógica da Sentença, Livraria Petrony, (2003) (*in Portuguese*).

19. Raiffa, H., The art and science of negotiation: how to resolve conflicts and get the best out of bargaining, Cambridge, The Belknap Press of Harvard University Press (1982).

20. Steenbergen, W., Rationalizing Dispute Resolution: From best alternative to the most likely one, in Proceedings 3rd ODR workshop, Brussels (2005).

21. Ury, W., Brett, J.M., Goldberg, S.B., Getting Disputes Resolved: Designing Systems to Cut the Costs of Conflict. San Francisco, CA: Jossey-Bass Publishers (1988).

22. Wooldridge, M., Jennings, N.R., Kinny. D., The Gaia Methodology for Agent-Oriented Analysis and Design. Autonomous Agents and Multi-Agent Systems, vol. 3 (2000).

23. Zeleznikow, J., Abrahams, B., Incorporating issues of fairness into development of a multi-agent negotiation support system. Proceedings of the 12th International Conference on Artificial Intelligence and Law, Barcelona, Spain: ACM (2009) 177-184.

24. Zweigert, K., Kötz, H., An Introduction to Comparative Law, Clarendon Press (1998).

25. Waterman, D.A., Peterson, M.. Rule-based models of legal expertise. In the Proceedings of the First National Conference on Artificial Intelligence, Stanford University (1980).

26. Cáceres, E., EXPERTIUS: A Mexican Judicial Decision-Support System in the Field of Family law. In Francesconi, E. B. E., Sartor, G., & Tiscornia, D. (Eds.), Legal Knowledge and Information Systems (pp. 78-87). IOS Press (2008).

27. Kersten, G., Noronha, S., Negotiation via the World Wide Web: A Cross-cultural Study of Decision Making, Group Decision and Negotiation, vol. 8, pp. 251-279 (1999).

28. Thiessen, E.M., ICANS: An Interactive Computer-Assisted Multi-party Negotiation Support System. PhD Dissertation, School of Civil & Environmental Engineering, Cornell University, Ithaca, NY (1993).

29. Zeleznikow, J., Bellucci, E., Family_Winner: integrating game theory and heuristics to provide negotiation support. In Proceedings of Sixteenth International Conference on Legal Knowledge Based System (pp. 21-30) (2003).

30. Lodder, A. and Zeleznikow, J. 2010. *Enhanced Dispute Resolution through the use of Information Technology*. Cambridge University Press

31. Black, H. C. 1990. *Black's Law Dictionary,* West Publishing Company, St. Paul, Minnesota