Using Mediation to Solve Disputes with Avoiding Parties

Davide Carneiro[1], Paulo Novais[1], Francisco Andrade[2], José Neves[1]

[1] Department of Informatics, University of Minho, Braga, Portugal
{dcarneiro, pjon, jneves}@di.uminho.pt
[2] Law School, University of Minho, Braga, Portugal
fandrade@direito.uminho.pt

Abstract. In virtually every environment there is the chance that, sooner or later, a dispute will arise. Disputes can take place in the most different scenarios and concern the most different subjects. With the advent of the telecommunication technologies, disputes also started to take place in virtual environments. In order to settle these new disputes, Online Dispute Resolution tools started to emerge. In this paper we present one of such tools, aimed at supporting mediation between two or more parties. Specifically, this tool looks at past known mediation processes and tries to guide the process into a successful outcome. It targets scenarios in which one or more party exhibits avoiding or uncooperative conflict styles, i.e., the party cannot or is not willing to generate valid proposals for dispute resolution.

Keywords: Case-based Reasoning, Multi-agent Systems, Online Dispute Resolution, Mediation.

1 Introduction

In recent years the changes caused by technological evolution have been overwhelming. This has effects in every aspect of our lives. Namely, we can now easily engage into electronic transactions or contracts with virtually anyone in the world. One thing that did not change is the fact that, despite the nature of the contract, disputes are likely to arise. In traditional contracts the process followed to solve the dispute usually consists in going into court. There are however some known drawbacks in this method. Generally, litigation in court is seen as costly and slow. Moreover, it is a win-lose or even a lose-lose process. This means that one party wins what the other loses or both parties will be worst at the end of the process than they were at the beginning. Litigation thus results in a highly competitive approach to solve disputes. Moreover, it is also a public process in which the parties have a high level of exposure to the media, which is usually undesirable.

In order to address some of these drawbacks alternative approaches emerged that aimed at solving disputes in a more cooperative fashion. The so-called Alternative Dispute Resolution (ADR) methods include mediation [1], negotiation [2], arbitration [3], among others. Essentially, these methods aim at creating a private environment in which the parties cooperate to solve the dispute in a cheaper and faster way.

Nevertheless, we easily conclude that these methods are outdated when considering the new nature of disputes.

In fact, nowadays disputes are not only increasing in number but have also new characteristics. The most evident is that the two disputant parties are no longer in the geographical proximity of each other but can be anywhere in the world. Traditional litigation is not suited anymore to deal with these disputes as, in order to settle it, the parties would have to meet in a physical place that is, probably, distant from their respective home locations. Being a slow process, it is also not suited to deal with the increasing amount of disputes. Likewise, ADR methodologies are outdated, namely because of the question of the location.

A new trend thus emerged that aims at bringing the advantages of ADR methodologies into virtual spaces, the so-called Online Dispute Resolution (ODR) methodologies [4]. These are similar to the ones of ADR but can be used by parties scattered around the world through the use of the new telematic means. Moreover, these methodologies can be supported by intelligent software agents that can automate processes and support the parties involved [5, 6]. In fact, increasing the efficiency of the processes is paramount in order to deal with the rising amount of disputes.

The work presented in this paper builds on a pre-existing ODR platform: UMCourt. Specifically, we are looking at ways to improve the efficiency of mediation processes in which one or more parties exhibit an avoiding behavior. Such parties are not willing or cannot generate potential solutions, thus constituting an impediment for the effective resolution of the dispute. In that sense, our approach consists in letting the system assume the responsibility of proposing solutions for the resolution of the dispute rather than asking the parties for them. Thus, parties have only to express how they feel about each proposal that the system provides. This is then used by the system as an input to decide about the next solution to propose with the aim to achieve a solution that can satisfy all the parties. Moreover, in order to decide about the following outcome, the system also considers a list of previous successful similar mediation processes. Instead of having to explicitly plan a strategy and propose a solution in each step, parties have only to answer to proposals of the system. In that sense, we expect to increase the acceptation and use rate of these platforms among parties that show an avoiding behavior.

2 Conflict Resolution

Conflict resolution is the action by means of which two entities use a method for alleviating or eliminating sources of conflict. These methods generally include negotiation, mediation or arbitration, most of the times with the assistance or supervision of a neutral third party. In any case, conflict resolution aims at solving the dispute outside of courts, i.e. it aims at avoiding traditional litigation in court. Conflicts can arise in the most different scenarios and they are present in our day-to-day life. In 1974, Kenneth Thomas and Ralph Kilmann formalized the way we respond to conflict situations into five different modes in terms of individual's assertiveness and cooperativeness [7]. In this context, assertiveness denotes the extent to which the person attempts to satisfy his/her own interests while cooperativeness

denotes the extent to which the person attempts to satisfy the other person's interests. The conflict styles are:

- Competing - This is an uncooperative style by means of which an individual aims at maximizing his/her own gain at the other's expenses. This is a power-oriented style in which an individual will use whatever power seems appropriate to win his/her position (e.g. ability to argue, rank, economic sanctions);
- Accommodating – This style is the opposite of competing, i.e., it is cooperative. When an individual shows an accommodating behavior, he/she neglects his/her own gain to maximize the gain of the other. Under this behavior one founds an element of self-sacrifice. Accommodating includes well-known behaviors such as selfless generosity or charity, obeying another individual's order when we may prefer not to do so or accepting another's point of view;
- Avoiding - The individual that shows an avoiding style of conflict tries to satisfy neither his/her own interests nor those of the other individual. It can be said that he/she is not dealing with the conflict. This style may be evidenced by behaviors such as diplomatically sidestepping an issue, postponing an issue until a better opportunity arises, or simply withdrawing from a threatening situation;
- Collaborating – This cooperative style is the complete opposite of avoiding. When an individual collaborates, he/she attempts to work with the other party to find some solution that fully satisfies the interests of both parties. In this process, the individual explores an issue to discover the underlying desires and fears of the two individuals. An individual that is collaborating might try to explore a disagreement to learn from other's insights or to try to find a creative solution to an interpersonal problem; and
- Compromising – When an individual has a compromising style of dealing with a conflict, he/she tries to find some expedient, mutually acceptable solution that can partially satisfy both parties. This style is somewhat an intermediate one between competing and accommodating. In fact, compromising gives up more than competing but less than accommodating. Similarly, it addresses an issue more directly than avoiding, but does not explore it in such a detail as collaborating. Generally, compromising can mean splitting the differences between the two positions, exchanging concessions, or seeking a quick middle-ground solution.

Whether it is because of the practice or because of our temperament, each of us is capable of using all of these conflict-handling styles. Moreover, none of us can be characterized as having one single style of dealing with a conflict. Nevertheless, certain individuals rely on some modes more than others and, therefore tend to use them more often. It is therefore important, in the first place, to determine the style of the parties towards the conflict resolution process in an attempt to define how to guide it. One way of determining this style is by taking the TKI test (Thomas-Kilmann Conflict-Mode Instrument), a test designed specifically by Thomas and Kilmann.

Once the styles are identified there are several strategies that can be used for conflict resolution. One of them is the well-known Interest-Based Relational (IBR).

This approach is characterized by respecting peoples' individuality while at the same time helping each party to avoid to become too entrenched in a fixed position.

3 UMCourt

UMCourt is an agent-based platform for the development of ODR services. This platform implements a wide range of functionalities that can be used to build complex services. The key component of UMCourt is a Case-based Reasoning (CBR) module that allows to develop services such as retrieving similar cases, retrieving most likely solutions, retrieve similar solutions, retrieve similar mediation processes, among others. This work essentially builds on this module. As the architecture of UMCourt has already been presented previously, we will here only briefly describe the agents that build it and their high level roles. For a more detailed description see [8].

In UMCourt agents run inside an instance of a Jade agent platform [9] and are organized in two groups. The *Main Agents* group is populated by agents that have a major and autonomous role in the CBR process. These are detailed in Table 1. In Table 2 the agents of the *Secondary Agents* group are listed. These have no autonomy, having as its foremost objective to support the actions of the main agents. Figure 1 depicts a high level view of the organization of the agents. This departure between main and secondary agents has been performed in order to simplify the first ones. Following this line of attack, we not only simplify the main agents but also increase code (thus functionalities) reuse.

Table 1. The Main Agents and their functionalities.

Name	Functionalities
Coordinator	Receive task requests from other agents (e.g. external agents, interface agents) and take the necessary steps (requesting tasks to other agents) in order to perform them. This agent maintains a list of active tasks and has access to a list of finite state automata that define the next action for each task, provided by the FSA agent.
Retriever	Retrieve the cases more similar to a given one. This agent has the autonomy to change the search settings, the similarity parameters and the retrieve algorithms in order to perform a better selection of cases.
Reuse	When requested by the Coordinator, performs the necessary actions to adapt a given case so that it can be used.
Reviser	Looks at a group of cases in order to select an outcome/solution for a given case. Proposes the outcome to the coordinator as well as a justification and waits for the outcome. If the outcome does not comply with the one suggested provides a list of more probable reasons for the failure.
Learning	Has the autonomy to make changes to the knowledge base and to the rules according to the each proposed outcome and real outcome. This agent embodies the ability to acquire new experiences and learn with failed ones.

Table 2. The Secondary Agents that support the lifecycle of the Main ones.

Name	Functionalities
FSA	Contains a list of Jade FSM behaviours that describe the guidelines or steps necessary for an agent to implement specific actions.
Selector	Multiple instances of this agent exist that implement different pre-selected algorithms (e.g. Template Retrieval, Clustering).
Similarity	This agent is able to compute the values of similarity between two cases, according to the desired rules.
Settings	Defines several search and similarity settings according to which retrieve parameters can be changed.
Database	Implements an application layer that surrounds the database of cases, that caters for all the actions to be applied to the cases stored.
Rules	Embodies rules of type if *condition* then *action* that provide the basic reactive actions for guiding some of the remaining agents.
ATNA	Computes the BATNA and WATNA in a given context using a set of logical rules defined after the Portuguese labour law.
Loader	Loads the information of cases from XML files and provides it as a Java object maintaining and updating loaded cases.
Indexer	Indexes each new case in the database according to the rules defined.
Parser	Checks the validity and parses XML files according to the defined schemas.
Process Validity	Verifies the validity of a case in terms of the dates and the corresponding statutory periods.
Roles	Contains information about the roles of registered external agents. This is used to decide which actions each external agent can perform.

4 The Mediation Process Model

4.1 Compilation of Initial Information

As mentioned, the mediation process depicted in this work builds on the previously existing UMCourt architecture. Namely, UMCourt is used to compile some initial information that is useful for both the mediation tool and the disputant parties. This information includes the BATNA or Best Alternative to a Negotiated Agreement [10]. In fact, when parties enter into a dispute resolution process, they expect to achieve better results than would otherwise occur. It is of utter importance that, during this process, parties are aware of the possible results if the negotiation is unsuccessful. In fact, failing to do so may drive the parties into accepting an agreement that they would do better or rejecting one that they would do better of fall into. Likewise, the WATNA, or the Worst Alternative to a Negotiated Agreement is equally important. Looking at these two elements, parties can definitively improve their outcome by looking at the whole picture. ODR platforms that embody such concepts can help parties take better decisions [11].

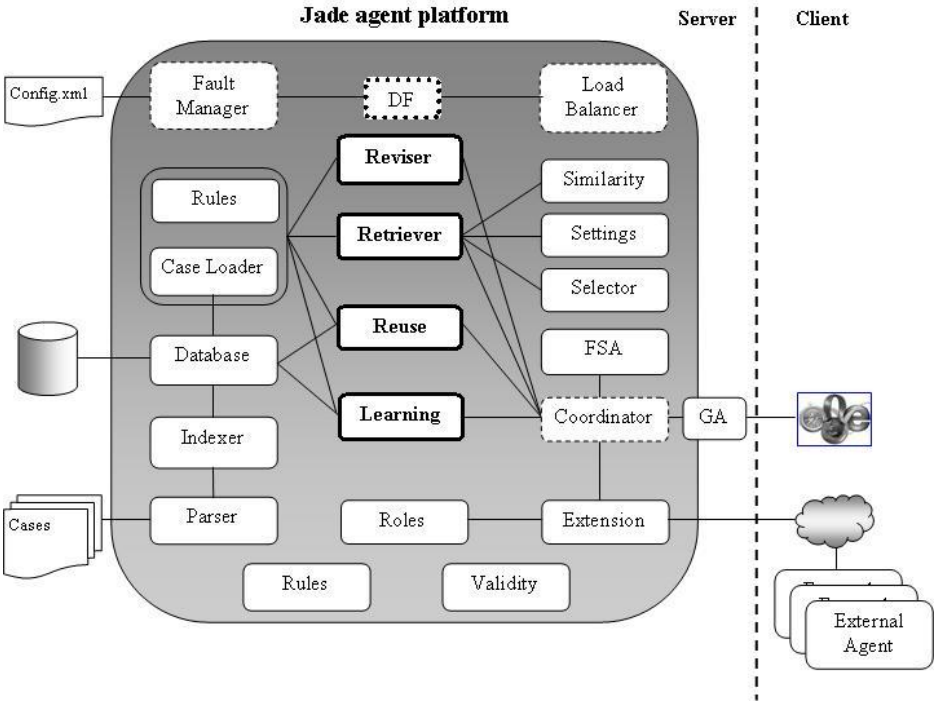

Fig. 1. The organization of the agents in UMCourt. Rounded rectangles represent agents and lines represent the main communication paths.

Although these two values are important, they may not be enough for the parties to take the best decision possible. In fact, parties that consider only their BATNA run the risk of holding to a position that is unrealistic. Indeed, the best alternative to a negotiated agreement is usually not the most likely. In that sense, UMCourt also provides the MLATNA - Most Likely Outcome for a Negotiated Agreement [12] for a given case. This perception takes us to the most likely scenario if the dispute resolution process fails. It represents therefore a good starting point for the process to start. When using these details, parties are aware of the possible and potential consequences of solving the dispute in court. In this sense, parties are able to take their decisions while encompassing the whole picture.

The last information that the mediation tool uses is a list of past successful mediation processes that happened in the past and concerned a similar dispute. This list is compiled by the Case-based Reasoning module of UMCourt and is based on the similarity of the current dispute with past known disputes. In that sense, UMCourt will select the most similar cases and the corresponding mediation processes that led to the solution, as detailed in [13]. This information will then be used by the system to try to guide the dispute into a successful end as will be seen below.

4.2 New Agents

A software agent can be a complex entity and can thus be defined from several points of view. However, independently of the type of agent, there are some abstract notions that are common and can be used to define it. Namely, we define an agent according to their roles, services and world description.

Definition 4.1. An agent is defined as a 4-tuple:

$$A:=(Id, R, S, WD)$$

where Id is the unique identifier of the agent, R a set of roles, S a set of services that are provided by the agent and WD a set of additional descriptors of the world in which the agent exists.

Definition 4.2. The world is defined as a 3-tuple:

$$WD:=(WId, OD, ACLD)$$

where WId uniquely identifies the world, i.e., an instance of the dispute resolution platform, OD is an ontology description and ACLD is an agent communication language description.

In order to implement the mediation tool, UMCourt has been extended with two new agents that implement the concept of blackboard. In this paradigm, there is a shared space, the blackboard, to which the participants can publish their messages as well as read messages from the others participants or external entities taking part in the process. In this context, these messages essentially contain outcome proposals from the mediator or the parties. Specifically, this environment is built on top of instances of two different agents: *blackboard* and *party*. The former is responsible for controlling the whole process, while the second facilitates the interaction of the parties with the blackboard. In a few words, the process model implemented by these two types of agents consists in iteratively send proposals for outcomes until an agreement is reached or the process ends by another reason. The roles and services of these two agents are presented in table 3 and table 4.

Table 3. The information about the roles and services of agent *blackboard*.

Roles	Services
Request Initial Information	Get Information
Initiate Mediation	Get Status
Accept Register	Register
Refuse Register	Unregister
Build new Outcome	
End Mediation	

Table 4. The information about the roles and services of agent *party*.

Roles	Services
Register	Get Message
Exit	Edit Message
Propose new Outcome	Send Message
Accept	
ReplyTo	
Ignore	
Refuse	

4.3 The Mediation Process Model

The mediation process starts with the system providing some useful information for the parties. This information comprises the BATNA, WATNA and the MLATNA and will allow the parties to take better and more weighted decisions throughout the process. Basically, the mediation develops as follows: the platform goes on suggesting solutions until one party leaves the process, all agree on the proposed solution or the platform runs out of solutions.

In order to implement the mediation algorithm, a list of previous negotiation sessions is needed. This list is provided by UMCourt. It contains mediation sessions that happened in the past and have a given degree of similarity with the current case. Moreover, they have already been applied successfully in the past so they may constitute a valid solution for the current dispute. Each of these mediation sessions contains a list of solutions. These solutions describe the several steps that the mediation session took, i.e., how the mediation evolved until the final outcome was achieved. In order for UMCourt to build this list of past mediation sessions, it looks at similarity of the current case with past known cases. It will select the mediation sessions that were used to solve similar cases. To implement this, UMCourt makes use of the CBR module, as depicted in [13]. Moreover, the mediation sessions are sorted according to the decreasing degree of similarity of the corresponding cases so it is a sorted list.

The mediation process model is implemented as a cycle. In each step of this cycle the platform suggests a possible solution and waits for the answers of the parties. When the process starts, the platform proposes the first solution of the most similar mediation process known. This is also the MLATNA computed by UMCourt, i.e., the system starts by proposing the most likely outcome to the parties. A solution will then be worked out from this point on. After publishing this proposal on the blackboard, the system waits for the parties to read it and answer. As this approach is focused on parties that exhibit avoiding behaviors, the system does not expect them to answer with an alternative suggestion for a solution. Parties can rather reply to the proposal in one of three ways: they can either agree, disagree or simply ignore it. Once all the parties published their answers on the blackboard, the platform will analyze them. If at least one party leaves the mediation, the process ends unsuccessfully. If all the parties agree on the solution, the process ends with success as one consensual outcome has been achieved. Otherwise, there are two possible paths.

If the majority of the parties agree with the solution proposed by the system, it means that the mediation is following a promising path. In that sense, the platform

will propose the following solution of the mediation session being considered. However, if there are no more solutions on the current session, the system will move on to the first solution of the following mediation session, if any.

On the other hand, if the majority of the parties does not agree with the proposed solution, the platform will drop the mediation session being used and advance into the following one, if any. The platform does so because it interprets this configuration of replies as indicating that the mediation path being followed is not promising. It will thus start following the next known mediation session in the following round.

The high-level algorithm that implements the mediation process model presented in this paper

```
Algorithm Mediation is
  input:  List of previous mediation processes, L
          List of parties, P
  output: A solution for the dispute

  round := 0       (identifies the round)
  msgSet := []     (a list of received messages)
  agree := 0       (the number of agents that agree)
  exit := 0        (the number of agents that exit)
  proposal := []   (the current proposal)
  i := 0      (the mediation process being currently used)
  j := 0           (the current step being proposed)
  while (agree < length(L) and exit < 1)
       mediation := L_i
       solution := mediation_j
       publish proposal
       for each party in P
           msg := receive from party or timeout
           msgSet := msgSet ∪ msg
       agree = count "agree" in msgSet
       exit  = count "exit" in msgSet
  if (exit > 0)
      return null
  if (agree = Length(P))
      return solution
  else if (agree > Length(P)/2)
            j++
       else i++
            j := 0
  if (j = Length(mediation))
       j := 0
       i++
  if (i = Length(L))
       return null
  round++
```

To better follow the evolution of this process, several user interfaces have been developed. In Figure 2 the interface for the *blackboard* agent is shown.

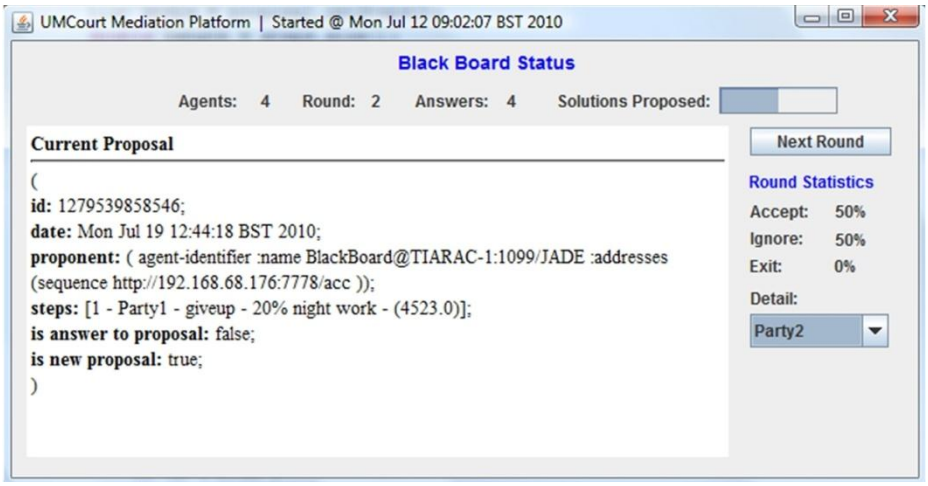

Fig. 2. The interface for the *blackboard* agent.

Essentially, this interface shows the state of the mediation. It provides information about the current round, the number of parties currently registered, the answers already received in this round, and the amount of solutions that have already been proposed. Moreover, it shows some statistics for the current round. It is also possible to select each party and see, in detail, the corresponding messages in another interface.

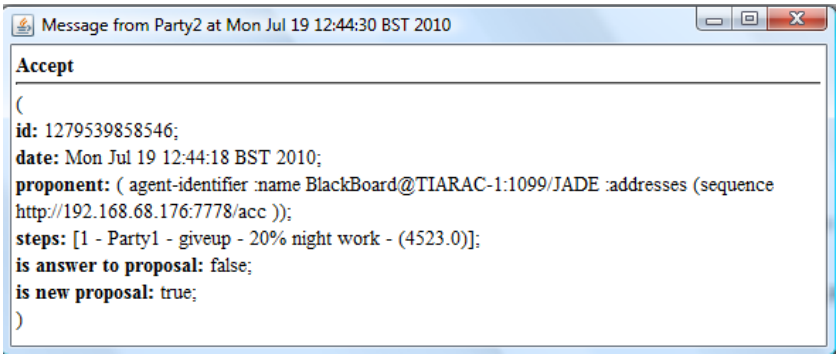

Fig. 3. An interface showing a message from a party denoting the acceptation of a given proposal.

In figure 3 it is shown a message from agent Party2 stating that it agrees with the proposal published by agent *BlackBoard@TIARAC-1:1099/JADE*, with id 1279539858546. This specific proposal is made up by only one step in which, according to the defined ontology, party 1 will gave up 20% of the night work, evaluated in a total amount of 4523.0.

5 Conclusions

Parties that cannot effectively generate solutions and strategies for dispute resolution can be a major obstacle. In such scenarios it is necessary the existence of an external entity, with the ability to generate valid potential solutions. In this paper we have presented such an entity. Particularly, we have developed a mediation platform that is able to propose valid solutions to parties that are not willing or cannot do so. Moreover, the solutions proposed are potential solutions as they are selected from previous successful mediation processes. In order to do this, the system uses a previously developed Case-based Reasoning algorithm that retrieves mediation processes based on the similarity of the cases addressed. The key idea in the whole process model is that the system has the responsibility to suggest potential outcomes while the parties have only to state whether they agree with it or not. This way, although parties cannot generate valid solutions, they can still solve their dispute with the help of an electronic mediator. Moreover, the feedback from the parties will be used by the system to infer if the mediation process is going in the correct direction or an alternative should be chosen.

6 Future Work

We intend to continue the development of the mediation platform by following two different paths. We will, on the one hand, develop web-based interfaces that will allow for the mediation tool to be used in any standard web browser. On the other hand, we will continue improving the core of the mediation platform with some new functionalities. Namely, we will develop a simulation platform that will allow us to study specific scenarios. Specifically, we aim at developing a configurable agent that will simulate a party with a given conflict resolution style in order to study how each different style affects the whole mediation process. Using this knowledge we will then make improvements to the mediation platform that will allow it to adapt to the different conflict resolution styles in order to maximize the efficiency of the process.

Acknowledgments. The work described in this paper is included in TIARAC - *Telematics and Artificial Intelligence in Alternative Conflict Resolution* Project (PTDC/JUR/71354/2006), which is a research project supported by FCT (Science & Technology Foundation), Portugal.

References

1. Brown, H., Marriott, A.: ADR Principles and Practice. Sweet and Maxwell (1999)
2. Raiffa, H.: The Art and Science of Negotiation. Harvard University Press (2002).
3. Bennett, S. C.: Arbitration: essential concepts. ALM Publishing (2002)
4. Katsch E. and Rifkin J.: Online dispute resolution – resolving conflicts in cyberspace. Jossey-Bass Wiley Company, San Francisco (2001)

5. Peruginelli, G. Artificial Intelligence in Alternative Dispute Resolution. In: Sartor, G. (Eds.) Proceedings of the workshop on the Law of Electronic Agents (20002)
6. Lodder, A., Thiessen, E.: The role of artificial intelligence in online dispute resolution. In: Workshop on Online Dispute Resolution at the International Conference on Artificial Intelligence and Law, Edinburgh, UK (2003)
7. Thomas, K., Kilmann, R.: Conflict and Conflict Management. Available at http://www.kilmann.com/conflict.html (1974) Accessed in: 07/2010
8. Andrade F., Novais P., Carneiro D., Zeleznikow J., Neves J., Using BATNAs and WATNAs in Online Dispute Resolution, in New Frontiers in Artificial Intelligence, Kumiyo Nakakoji, Yohei Murakami and Eric McCready (Eds), (JSAI-isAI 2009 Workshops, LENLS, JURISIN, KCSD, LLLL, Tokyo, Japan, 2009, Revised Selected Papers), Springer - LNAI 6284, ISBN 978-3-642-14887-3, pp 5-18, (2010).
9. Bellifemine, F., Poggi, A., Rimassa, G.: Developing Multi-agent Systems with JADE. In: Intelligent Agents VII Agent Theories Architectures and Languages, 2001, pp. 42-47 (2001)
10. Notini, J.: Effective Alternatives Analysis In Mediation: "BATNA/WATNA" Analysis Demystified. Available at http://www.mediate.com/articles/notini1.cfm. (2005) Accessed in 07/2010
11. De Vries, B.R., Leenes, R., Zeleznikow, J.: Fundamentals of providing negotiation support online: the need for developing BATNAs. Proceedings of the Second International ODR Workshop, Tilburg, Wolf Legal Publishers, pp- 59-67 (2005)
12. Guasco, M. P., & Robinson, P. R. Principles of negotiation. Entrepreneur Press (2007)
13. 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, pp 47--52 (2009)