Carneiro D., Novais P., Andrade F., Zeleznikow J., Neves J., Using Case-based Reasoning to Support Alternative Dispute Resolution, in Distributed Computing and Artificial Intelligence, Andre Carvalho, Sara Rod-González, Juan Paz and Juan M. Corchado (Eds) (7th International Symposium on Distributed Computing and Artificial Intelligence (DCAI 2010), Valência, Spain), Springer - Series Advances in Intelligent and Soft Computing, vol. 79, ISBN: 978-3-642-14882-8, pp. 123--130, 2010.

Using Case Based Reasoning to Support Alternative Dispute Resolution

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

[3] School of Management and Information Systems
Victoria University, Melbourne, Australia
John.Zeleznikow@vu.edu.au

Abstract. Recent trends in communication technologies led to a shift in the already traditional Alternative Dispute Resolution paradigm, giving birth to the Online Dispute Resolution one. In this new paradigm, technologies are used as a way to deliver better, faster and cheaper alternatives to litigation in court. However, the role that technology plays can be even further enhanced through the use of artefacts from the Artificial Intelligence field. In this paper we present UMCourt, an Online Dispute Resolution tool that borrows concepts from the fields of Law and Artificial Intelligence. The system keeps the parties informed about the possible consequences of their litigation if their problems are to be settled in court. Moreover, it makes use of a Case-based Reasoning algorithm that searches for solutions for the litigation considering past known similar cases, as a way to enhance the negotiation process. When parties have access to all this information and are aware of the consequences of their choices, they can take better decisions that encompass all the important aspects of a litigation process.

Keywords: Case-based Reasoning, Multi-agent System, Online Dispute Resolution

1 Introduction

The shift of already traditional Alternative Dispute Resolution (ADR) methods from a physical to virtual place [1] led to a new paradigm called Online Dispute Resolution (ODR). Using this new technology-based approach, disputant parties have an easier, simpler and faster course than litigation in court, saving both temporal and monetary costs [2]. In that sense, several methods of ADR and ODR may be considered, from negotiation and mediation to modified arbitration or modified jury proceedings [3].

Carneiro D., Novais P., Andrade F., Zeleznikow J., Neves J., Using Case-based Reasoning to Support Alternative Dispute Resolution, in Distributed Computing and Artificial Intelligence, Andre Carvalho, Sara Rod-González, Juan Paz and Juan M. Corchado (Eds) (7th International Symposium on Distributed Computing and Artificial Intelligence (DCAI 2010), Valência, Spain), Springer - Series Advances in Intelligent and Soft Computing, vol. 79, ISBN: 978-3-642-14882-8, pp. 123--130, 2010.

The process of developing ODR systems frequently consists on the development of tools that provide legal advice to the disputing parties. Here, it must be considered the role of the BATNA or Best Alternative to a Negotiated Agreement [4]. In fact, when parties enter into a negotiation process, they expect to achieve better results than it would otherwise occur. It is of utter importance that, during this negotiation process, the 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 better of rejecting or rejecting one that they would better of enter 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 negotiation by looking at the whole picture. ODR platforms that embody such concepts as BATNAs and WATNAs can help parties to take better decisions [5]. When technology is given a more autonomous and major role, , the ODR system is categorized as a second generation one [6]. It not only puts the parties into contact. It is used for idea generation, planning, strategy definition and decision making processes. The development of Second Generation ODR, in which an ODR system might act as an autonomous agent [6] is an appealing way for solving disputes. The architecture of such systems needs to be expansible, modular and compatible. Thus using Case-based Reasoning (CBR) [7], Multi-agent Systems and Rule-based Systems is appropriate.

In this paper we present a hybrid system that merges the versatility of an agent-based architecture with the completeness of CBR and the simplicity and efficiency of rules. This allows the system to look at past cases, select the most similar ones and adapt the solutions to the current problem. Furthermore, rules are used to define the simpler tasks and secure the whole process. This work is being developed in UMCourt, an ODR platform in the context of the Portuguese Labour Law. This platform is part of the TIARAC project - Telematics and Artificial Intelligence in Alternative Conflict Resolution, a project funded by FCT – the Portuguese Science and Technology Foundation.

2 Related Work

Given the fact that legal practice is largely based on the concept of precedence and the notion of case, it is important to investigate CBR [18]. The many successful cases of application of CBR techniques to the legal domain attest the ability of this technique to deal with this knowledge-based domain and support our efforts to enhance the dispute resolution process. In MEDIATOR [8], CBR is used to look at past cases in order to find solutions for problems in the context of international disputes. JUDGE [9] in the other hand, focuses on criminal sentencing. We can also mention HYPO [10] that addresses patent law, and CABARET [11], the result of improving HYPO with Rule-based reasoning.

BEST [16] is a project that specifically looks at the use of the BATNA in a semantic web context and INSPIRE [17] focuses on the study of negotiation processes. Whilst BEST and INSPIRE support ADR, most decision support tools have one thing in common: they attempt to help lawyers to win cases in a trial.

Carneiro D., Novais P., Andrade F., Zeleznikow J., Neves J., Using Case-based Reasoning to Support Alternative Dispute Resolution, in Distributed Computing and Artificial Intelligence, Andre Carvalho, Sara Rod-González, Juan Paz and Juan M. Corchado (Eds) (7th International Symposium on Distributed Computing and Artificial Intelligence (DCAI 2010), Valência, Spain), Springer - Series Advances in Intelligent and Soft Computing, vol. 79, ISBN: 978-3-642-14882-8, pp. 123--130, 2010.

This means that parties will engage in potentially time-consuming and expensive processes, in which both parties will lose something (e.g. time, privacy, reputation, money) even if one of them eventually wins the litigation.

In this paper we propose an alternative approach to handle this problem, based on two key ideas. On the one hand, we look at past cases and suggest to the parties a possible outcome. In order to propose this outcome, we use the CBR algorithm to determine the MLATNA - Most Likely Outcome for a Negotiated Agreement [19]. This concept denotes the most likely outcome scenario if the negotiation process fails. If the parties do not agree on the suggestion, they can start to gradually work out a more satisfactory one, using as a starting point the MLATNA. On the other hand, we warn the parties about 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. Essentially, with this approach we plan to diminish the number of cases that actually have to be solved through litigation.

3 System Architecture

As being said in the introductory section, the architecture is based on the multi-agent paradigm [12]. Specifically, we are using Jade (Java Agent Development Framework), in compliance with FIPA (Foundation for Intelligent Physical Agents) specifications. This allows us to develop the application layer in a highly modular fashion which makes it possible to build an architecture that is highly expansible and extensible. The development of the architecture is depicted in [13] and is out of the scope of this paper. Instead, we will briefly describe the agents that make up the architecture and their roles as this is important for the services on top of which the system is built are understood.

The agents 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 which have no autonomy, having as its foremost objective to support the actions of the main 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.

The services that these agents provide can be individually used by external agents or can be used in specific sequences in order to implement more complex tasks. Thus, this architecture can extend external systems and it can be extended either by making use of external services or by means of new agents.

Table 1. The Main Agents that implement the CBR process.

Agent Name	Role
Coordinator	Receives task requests from other agents and takes the necessary steps in order to execute them. This agent maintains a list of active tasks and has access to a list of automata that define the next action for each task.
Retriever	Retrieves the more similar cases. It has the autonomy to change, in real-

Carneiro D., Novais P., Andrade F., Zeleznikow J., Neves J., Using Case-based Reasoning to Support Alternative Dispute Resolution, in Distributed Computing and Artificial Intelligence, Andre Carvalho, Sara Rod-González, Juan Paz and Juan M. Corchado (Eds) (7th International Symposium on Distributed Computing and Artificial Intelligence (DCAI 2010), Valência, Spain), Springer - Series Advances in Intelligent and Soft Computing, vol. 79, ISBN: 978-3-642-14882-8, pp. 123--130, 2010.

	time, search settings, similarity parameters and retrieve algorithms.
Reuse	Performs the necessary actions to adapt a known case to a new context, so that it can be used.
Reviser	Looks at a group of cases in order to select an outcome/solution. Proposes the outcome to the coordinator as well as the corresponding justification and waits for the outcome. If the outcome does not comply with the one suggested, it compiles a list of possible reasons for the failure.
Learning	Has the autonomy to make changes to the knowledge base and to the rules that reflect the result of the actions of the system.

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

Name	Role
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 CBR Process

The basic unit of information in the CBR paradigm is the case. It represents a past experience that took place under a context that is also considered in the case. It can therefore be described as a contextualized piece of knowledge. This allows estimating the outcome of an experience that we are now living by looking at a past similar one and its respective outcome. Cases in UMCourt contain the description of the problem, the solution adopted and the verified outcome [20]. Part of this information is indexed in the database, whereas the remaining is stored in XML files.

However, purely storing information is not enough. In this section we define the processes that acquire and use this information, defined after the work of [7] and [14] (Figure 1).

Carneiro D., Novais P., Andrade F., Zeleznikow J., Neves J., Using Case-based Reasoning to Support Alternative Dispute Resolution, in Distributed Computing and Artificial Intelligence, Andre Carvalho, Sara Rod-González, Juan Paz and Juan M. Corchado (Eds) (7th International Symposium on Distributed Computing and Artificial Intelligence (DCAI 2010), Valência, Spain), Springer - Series Advances in Intelligent and Soft Computing, vol. 79, ISBN: 978-3-642-14882-8, pp. 123--130, 2010.

Retrieve The retrieve process is used to select a group of the most similar cases that are of relevance for solving a given problem by means of a similarity measure. Unlike database searches that target a specific value in a record, retrieval of cases from the case-base must be equipped with heuristics that perform partial matches, since in general there is no existing case that exactly matches the new case [15].

In UMCourt, we combine a template algorithm with a nearest neighbour one, resulting in a hybrid approach. The key idea is for the template retrieval algorithm to narrow the search space so that the nearest neighbour algorithm performs quicker. In this context, template retrieval behaves similarly to SQL queries. In the next step, the nearest neighbour algorithm is applied only to this set of cases instead of applying it to all the cases in the case memory, thus increasing the efficiency.

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

Equation 1 shows the closest neighbour algorithm in use. In this equation, n represents the number of elements to be considered for the similarity measure; W_i denotes the weight of element i with respect to the overall similarity; $Fsim$ speaks for the similarity function for element i; Arg refers to the arguments for the similarity function representing the values of the element i for the new case and the retrieved case, respectively N and R.

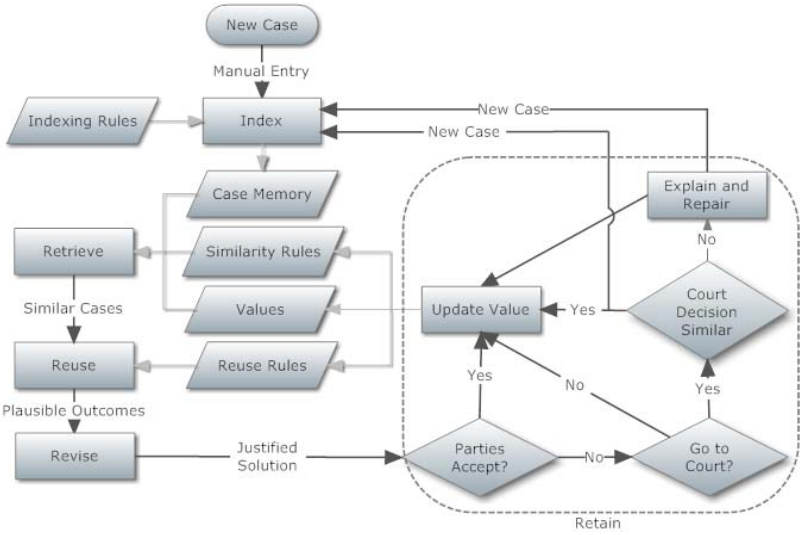

Fig. 1. A Flowchart depicting the major steps in the CBR process. Gray arrows represent access to information structures while black arrows represent the flow of control. Rectangles represent processes, parallelograms represent information structures and rhombuses represent decisions.

It is also important to 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 our application, we consider three types of information: (1) the

Carneiro D., Novais P., Andrade F., Zeleznikow J., Neves J., Using Case-based Reasoning to Support Alternative Dispute Resolution, in Distributed Computing and Artificial Intelligence, Andre Carvalho, Sara Rod-González, Juan Paz and Juan M. Corchado (Eds) (7th International Symposium on Distributed Computing and Artificial Intelligence (DCAI 2010), Valência, Spain), Springer - Series Advances in Intelligent and Soft Computing, vol. 79, ISBN: 978-3-642-14882-8, pp. 123--130, 2010.

objectives stated by each party at the beginning of the dispute; (2) the norms addressed by parties and witnesses; (3) 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). Concerning the date, the similarity function verifies if the two dates are within a given time frame.

Furthermore, the agent is able to dynamically change the metrics of the search and similarity algorithms. Specifically, the agent is able to choose what components to use. For instance, the agent can decide to compute the similarity of the norms addressed concerning only the article of each law (thus retrieving more laws but with less expected similarity). If the agent retrieves a significant amount of cases, it can be more precise and select the cases concerning also the number and even the item of each norm addressed, retrieving less cases, but with an assured higher degree of similarity.

$$fsim_{list} = \frac{|L_N \cap L_R|}{n}, n = \begin{Bmatrix} |L_N|, & |L_N| \geq |L_R| \\ |L_R|, & |L_N| < |L_R| \end{Bmatrix} \quad (2)$$

Reuse Having a list of cases with associated values of similarity, the system can present the users with the solutions that are most likely to occur, among other useful operations. This phase consists of adapting the solutions of the selected cases to match the context of the new case. Solutions are lists of steps that the parties take in order to achieve the outcome (typically trade-offs). This information is structured in a way that makes it possible to adapt it to other cases. The information considered is organized as a list of actions, each action containing a unique identifier and a structured textual description. These actions include demanding or offering an item, abdicating a given right in return for another item, among others. The items in dispute may represent components of the indemnity, rights or sums of money, just to name a few. Thus, in this phase, the solution of the retrieved cases is adapted by changing the necessary fields (e.g. names, dates, locations). The resulting solution can then be presented to the users in the form of plain text, in natural language.

Revise In the revise phase, the parties become aware of the outcome of the most similar case as a first solution for the dispute, which is the MLATNA. The parties can afterwards analyze the proposal, decide what each one should give and take and whether to accept each other decision or not. If the parties do not agree on the suggestion, they can search for another solution starting from the suggested one, by adding or removing actions to the original list. Furthermore, parties can state how much they value each item in dispute and let the system make suggestions that are aimed at the maximization of the mutual gain. At the same time, parties can analyze their BATNA and WATNA and know what they are risking if they decide not to accept an agreement. Moreover, the parties can see other similar cases as well as their respective solutions, which can be used to assist in the establishment of their own outcome. Each of these selected cases that the parties have access to are accompanied by a justification, stating why they are considered

Carneiro D., Novais P., Andrade F., Zeleznikow J., Neves J., Using Case-based Reasoning to Support Alternative Dispute Resolution, in Distributed Computing and Artificial Intelligence, Andre Carvalho, Sara Rod-González, Juan Paz and Juan M. Corchado (Eds) (7th International Symposium on Distributed Computing and Artificial Intelligence (DCAI 2010), Valência, Spain), Springer - Series Advances in Intelligent and Soft Computing, vol. 79, ISBN: 978-3-642-14882-8, pp. 123--130, 2010.

similar. This will help the parties determine how much attention they should devote to each case, according to their personal expectations.

Retain The last phase is the one in which the system embodies the changes that occurred during the whole CBR process. These changes include but are not limited to: changes to the value of cases, new cases learned, changes to thresholds or changes to rules. It is in this phase that the system actually adapts to new situations and learns new experiences, thus enriching its ability to deal with future problems. As an example, if the parties do not accept a proposed solution, the property of the corresponding case that denotes the acceptance rate will be decreased and vice-versa. Furthermore, if the parties decide to go into court, the system will also learn the new case that will be decided by a judge. Moreover, the system records the search and similarity settings for each type of case, and the ones that are more successful will be preferred in future iterations.

5 Results and Conclusions

Due to space restraints, in this paper we have focused in the description of the work done on the labour law domain. Nevertheless, the architecture is highly modular and can be applied in other domains, namely by changing secondary agents or making use of external services. In that sense, we are applying this architecture in two other prototypes in the fields of consumer law and property division. The common innovation that stems from these three prototypes is the use of three key concepts, namely BATNA, WATNA and MLATNA, as the way to define the negotiation process. Indeed, BATNA and WATNA define the Zone of Potential Agreement, i.e., the boundaries of the solution zone. Simultaneously, MLATNA constitutes the starting point for the negotiation process, representing the most likely outcome. Starting from this mutually favourable point, parties have a better possibility of reaching a satisfactory outcome. It does not aim to propose solutions to solve a dispute in court. It rather aims at preventing the parties from going into effective litigation, thus avoiding unnecessary costs. Knowledge-based domains such as the legal one are usually complex to model. Nevertheless, the role of the actors can be significantly improved with the use of Intelligent Systems based techniques. In UMCourt, parties can intuitively look at past cases and their solutions in an attempt to find a mutually satisfactory solution. At the same time, they can look at what may happen if they decide not to negotiate and solve the dispute in a court. Moreover, the system is not static. It is dynamic as it evolves with the results of its application to particular disputes, in an attempt to adapt to the desires of the parties and to the eventual legal changes.

Acknowledgements. 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.

Carneiro D., Novais P., Andrade F., Zeleznikow J., Neves J., Using Case-based Reasoning to Support Alternative Dispute Resolution, in Distributed Computing and Artificial Intelligence, Andre Carvalho, Sara Rod-González, Juan Paz and Juan M. Corchado (Eds) (7th International Symposium on Distributed Computing and Artificial Intelligence (DCAI 2010), Valência, Spain), Springer - Series Advances in Intelligent and Soft Computing, vol. 79, ISBN: 978-3-642-14882-8, pp. 123--130, 2010.

References

1. Bellucci, E., Lodder, A., Zeleznikow, J.: Integrating artificial intelligence, argumentation and game theory to develop an online dispute resolution environment, 16th IEEE International Conference on Tools with AI, pp. 749-754. (2004)
2. Klaming, L., Van Veenen, J., Leenes, R.: I want the opposite of what you want. In: Expanding the horizons of ODR, Proceedings of the 5th International Workshop on Online Dispute Resolution (ODR Workshop'08), Firenze, Italy, pp. 84-94. (2008)
3. Goodman, J.W.: The pros and cons of online dispute resolution: an assessment of cyber-mediation websites. In: Duke Law and Technology Review Rev. 4 (2003)
4. Notini, J.: Effective Alternatives Analysis In Mediation: "BATNA/WATNA" Analysis Demystified. Available at http://www.mediate.com/articles/notini1.cfm. (2005)
5. 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)
6. Peruginelli, G., Chiti, G.: Artificial Intelligence in alternative dispute resolution. Proceedings of the Workshop on the law of electronic agents – LEA. (2002)
7. Aamodt, A., Plaza. E.: Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Communications 7(1), pp. 39–59. (1994)
8. Simpson, R. L.: A Computer Model of Case-Based Reasoning in Problem Solving: An Investigation in the Domain of Dispute Mediation. Technical Report GIT-ICS-85/18, Georgia Institute of Technology, School of Information and CS, Atlanta, US. (1985)
9. Bain, W.M.: Case-Based Reasoning: A Computer Model of Subjective Assessment. Ph.D. Thesis, Yale University, Yale, CT, US. (1986)
10. Ashley, K.D.: Arguing by Analogy in Law: A Case-Based Model. In: Analogical Reasoning: Perspectives of AI, Cognitive Science, and Philosophy. D. Reidel. (1988)
11. Rissland, E.L., Skala, D.B.: Combining case-based and rule-based reasoning: A heuristic approach. In: Eleventh International Joint Conference on Artificial Intelligence, IJCAI-89: pp. 524-30, Detroit, Michigan. (1989)
12. Wooldrige, M.: An Introduction to Multiagent Systems. John Wiley & Sons. (2002)
13. Andrade, F., Novais, P., Carneiro, D., Zeleznikow, J., Neves, J.: Using BATNAs and WATNAs in Online Dispute Resolution. In: JURISIN 2009 - Third International Workshop on Juris-informatics, Tokyo, Japan, pp 15-26. (2009)
14. Riesbeck, C., and Bain, W.: A Methodology for Implementing Case-Based Reasoning Systems. Lockheed. (1987)
15. Watson, I., Marir, F.: Case-based reasoning: A Review. In: Knowledge Engineering Review, vol. 9, pp. 327–354. (1994)
16. Wildeboer, G., Klein, M. and Uijttenbroek, E.: Explaining the Relevance of Court Decisions to Laymen, A.R. Lodder & L. Mommers (eds.) Proceedings of JURIX 2007, Amsterdam, Berlin, etc.: IOS Press: 129-138. (2007)
17. Kersten, G. E. and Noronha, S. J.: Negotiation via the World Wide Web: A Cross-cultural Study of Decision Making, Group Decision and Negotiation, 8:251-279. (1999)
18. Ashley, K.D.: Case-based reasoning and its implications for legal expert systems, Artificial Intelligence and Law, vol. 1, pp. 113-208. (1992)
19. Guasco, M.P. and Robinson, P.R.: Principles of negotiation, Entrepreneur Press, 2007.
20. Carneiro, D., Novais, P., Andrade, F., Balke, T., Neves, J.: Supporting Dispute Resolution with Case-based Reasoning, TIARAC Technical Report, 2010 (Available at http://tiaracserver.di.uminho.pt/tiarac/).