An Architecture to Legal Distributed Case Repositories

Marco Costa, Orlando Sousa and José Neves

Universidade do Minho, Departamento de Informática
{marco,orlando}@di-ia.uminho.pt, jneves@di.uminho.pt

Abstract

The study of past cases is an important part of the every-day work carried out by people involved with the application of the law. Currently, the consultation and analysis of legal precedents are, in the majority of the cases, made manually, without the use of any supporting tool.

This paper presents an architecture that establishes legal case repositories. These repositories can serve as a base to decision support systems that are able to conduct intelligent analysis of legal precedents. The cases are stored in replicated repositories using the Standard Generalized Markup Language (SGML).

The architecture is constituted by two levels, each one implementing its own functionalities. The first level includes the repositories, which can be accessed to consult past cases, being the selection of the cases of interest made based in a model of indexation of the information contained in the cases themselves. On top of the first level reside the Artificial Intelligence modules, implement personal assistants, which assist the user in the intelligent analysis of cases.

1 Introduction

People involved in the application of the lawLaw often need to have access to information related to cases similar to their own. In this way, one can take advantage of past experiences, re-using the same arguments, trying to reach similar conclusions. Even a difference between cases can be profitable, since it may be used in a clarification process, and if nothing else, to decide which forms of reasoning do not apply to a particular problem (Branting 1989).

Under a classical setting, if someone needs information about a particular case similar to his own, he needs to search for the information manually on whatever textbooks or jurisprudence archives are available. Those elements may not be at hand, may be incomplete, or may not contain all the necessary information. A manual search is a time-consuming process, and relevant cases may be overlooked.

In this work we propose the creation of legal case repositories that can be accessed directly by the users, or can serve as a source of raw material to be used in intelligent (and possibly semi-automatic) cases' analysis.

In section 2, the first level of the architecture is presented, with a special focus on the structure of the information kept in the repositories, and on the replication of the repositories. The second level of the architecture is presented in section 3, which mentions some of the Artificial Intelligence (AI) techniques that can be used to conduct cases' analysis, and introduces personal assistants based on agent technology as the way to facilitate the access by the users to the AI modules.

2 Legal Case Repositories

2.1 Gathering the information

Figure 1 presents the structure of the first level of the architecture. In this level the repositories are kept that hold the information concerning concluded cases.

It is necessary to identify the relevant sources of information; any archive that contains information about concluded cases must be transferred to the case repositories, and the information on new cases must be collected automatically. To make this possible, a process of forwarding the information to the repositories must be established.

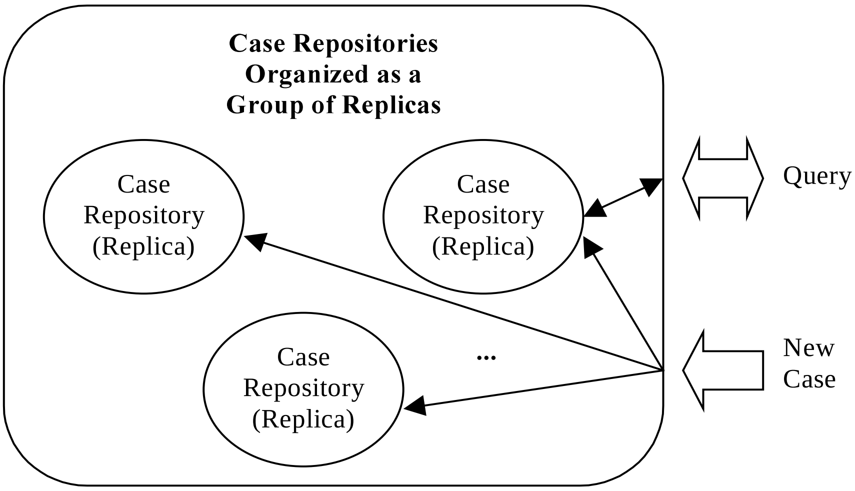

Figure 1. The First Level of the Case Repositories' Architecture

2.2 Replicating the Repositories

The case repositories are replicated using the *process groups* model (Birman 1991) (Navaratnam, Chanson, and Neufeld 1988). This model of computation is used to create a group having the case repositories as members. The objective is to maintain a group of replicas, each one with a copy of the case repository's information.

The two main categories of services associated to process groups are the membership services, and the communication services. The membership services allow the creation and the dynamic reconfiguration of groups. A group is created when an element enters a non-existing group. The group is destroyed when the last member leaves the group. During the lifetime of a group, different processes can join or abandon the membergroup, being each member of the group is informed of the changes in the group membership. At a given time, each process can be a member of zero or more groups.

The communication services must supply an efficient and versatile way for the group members to exchange data; the services must accept as a destination address a specific process (point-to-point communication), a list of processes (multicast) or all the members of the group (group broadcast).

The members of a group can assume different forms of behavior. They may compete, cooperate, or they can simply operate as replicas. When the group members are replicas of one another, if communication problems occur, the group can be divided into partitions, and later, when the communications are restored, the partitions can be reunited after a time period for re-synchronization.

In this case, the group is used to send the information to all replicas (members of the group) at the same time, and to keep them synchronized. The introduction of new cases into the repository is made simultaneously to all replicas, using group broadcast.

The replicas are intended to be distributed geographically, allowing a faster access from any point on the network. The replicas also allow for fault tolerance behavior, keeping the system working when something goes wrong (e.g. a server goes down).

2.3 Structured information

To make information processable, it must have an explicit structure which the computer can recognize and act upon. Every information set lies at some point on a continuum form, from orderly to chaotic. At the orderly end of the continuum are phone books and banking information systems, which have perfectly regular and unambiguous structures. At the chaotic end of the continuum are novels and poetry that have no structure at all. Having structure here means that the cases' repositories consist of identifiable components that can be processed by rule-based sys tems. Most chaotic information sets have a degree of literary structure that can only be processed by human sensibility.

The more orderly an information system is, the more processable it is. Relational database schemes have been used with great success to manage information sets at the orderly end of the continuum. At the chaotic end, no processing procedure is possible (or much required). In the middle of the continuum, where a degree of order exists, a degree of processability has been achieved with the use of markup languages, including, but not restricted to, those based on the SGML standard.

2.3.1 Introduction to SGML

SGML (Standard Generalized Markup Language) (Goldfarb 1981, 1990, Herwijnen 1994, Smith 1988) was developed in the early and mid 1980's as a generalized means of defining generic markup languages. It is a language for describing languages that represent documents. This language can readily be read and understood by human beings, but it is also a formally specified language, which can be processed by computer programs.

An SGML language is defined in a Document Type Definition (DTD). The DTD defines the types of elements that can be used in the document and the possible relationships among those elements (e.g. HTML is defined by a particular SGML DTD).

SGML deals with only the "syntactic" aspects of a document: the definition of a "chapter" can indicate that it is called a "chapter", describe what markup codes are recognized inside a chapter and can define the structural interrelationships of the components of a chapter (title, text, sub-sections, etc.). SGML does not deal with the "semantic" aspects of a document: the SGML definition contains no indication of what a "chapter" is or how it is to be dealt with. Figure 2 shows an example of a simple SGML document and a graphical view of its DTD.

```
<address>
<street>250 Some St.</street>
<city>Some City</city>
<zip>99000</zip>
</address>
```

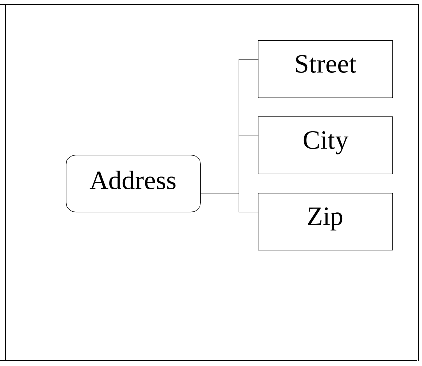

Figure 2. A Simple SGML Document and its DTD

2.3.2 *Intelligent Documents*

An intelligent document is one that knows more about itself than can be seen or deduced by simply looking at the printed page or its computer screen equivalent.

A “dumb” document is one that consists of text and formatting information: it is distinguished by having a single, fixed appearance and a single, linear order in which the text of the document is presented.

Complex indexes introduce a requirement for a higher level of intelligence in a document: a document has to know enough about itself to know which words or phrases are “significant”, in some sense known to the author or indexer. Using a hierarchical structure makes it possible for a computer to parse the pieces of a document and process them like one large data structure, with each tag identifying each piece and what composes it.

2.3.3 *Using SGML*

Queries on unstructured documents are limited in complexity to keyword search or string matching. Using SGML, more powerful queries can be made; e.g., instead of making the query “Braga” to search for all the cases where the plaintiff is from Braga, using SGML documents, one can make the query “Braga in City in Address in Plaintiff in Case” (Figure 3). This last query will return only the cases that one really wants, instead of returning all the cases where the word “Braga” occurs.

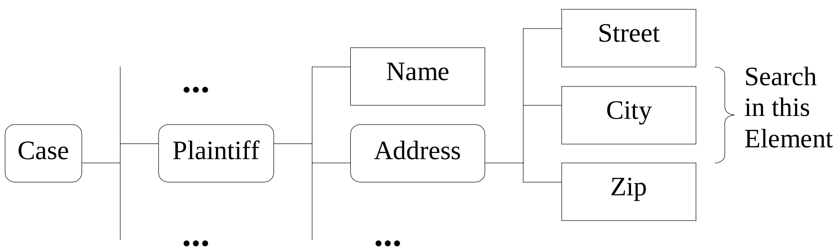

Figure 3. Querying a Structured Document

Since the processing of information structured with markup languages has proved less sophisticated and less economically advantageous than processing information structured with databases, architectures have been created that use both the relational database model and hierarchical document markup to describe its structure.

2.4 Direct Access to the Repositories

To make the information directly accessible to the users, each replica will run a web server, allowing the users to access the repositories through the World Wide Web (WWW) using a simple web browser (e.g. Netscape). In this way a mode of access is applied that is similar to the one used to search the Internet: the cases are selected and retrieved by a pattern matching like process.

3 Artificial Intelligence Modules

The AI modules constitute the second level of the architecture. These go beyond the simple consultation of past cases, establishing a higher level of functionality and abstraction that is adapted to the different types of users (e.g. solicitors, judges, lawyers).

To allow an easy access to the facilities offered at this level, intelligent agents can be used to establish the connection between the users and the AI modules (Figure 4), giving rise to so-called personal assistants.

An intelligent agent is a rational and autonomous entity, that controls its decisions, acting in accordance to its perception of the environment, trying to maximize the gain/costs quotient (Jennings and Wooldridge 1995, 1996).

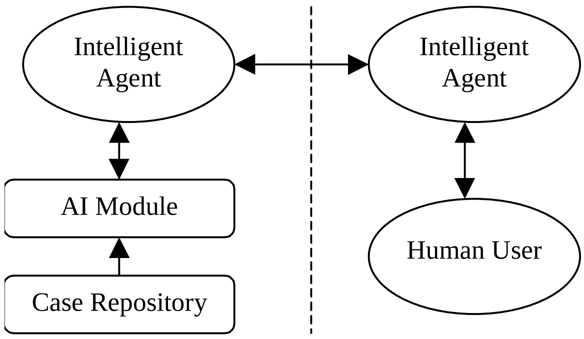

Figure 4. The Second Level of the Case Repositories' Architecture

Below will be referred to some of the contributions that Neural Networks and Case-Based Reasoning can make to the intelligent analysis of cases.

3.1 Neural Networks

3.1.1 Introduction to Neural Networks in the Legal Domain

Artificial neural networks (Bochereau, Bourcier, and Bourgine 1991, Hunter 1994, Warner 1990) were inspired by some of the features presented in the human mind. Neural networks are used to produce a set of output signals from a set of input ones. After a training period, when the network is presented with sets of input signals and the respective sets of correct output ones, it is expected that the network is able to produce good predictions for given sets of input signals.

Neural networks use statistical models to arrive at their conclusions. The methodology used to represent the cases must be consistent with the statistical nature of the neural networks. It is necessary to verify whether the statistical methods are coherent with one's conceptual model for the legal cases' processing.

The use of statistical methods can be appropriate in the domain of legal reasoning, depending on the purpose for which they are used, and on the model of legal reasoning adopted. One of the interesting aspects relative to the use of statistical techniques is that they do not supply any normative base to the taking of decisions, nor capture any element of the abstract reasoning conducted by judges.

For the neural networks to be able to predict the result of cases, one must first identify the features of the cases that are important for the relevant forms of legal reasoning. It is not necessary to evaluate the importance of these features, since the neural networks do that by themselves. It is thus verified that, in as far as it concerns the neural networks, legal doctrine is less important than legal reality.

3.1.2 Using Neural Networks to study legal cases

The underlying statistical model associated with neural networks makes them behave reasonably well in the identification of the correlations between a new pattern and the ones used to train the networks. In the domain of law application, one expects that the neural networks exhibit the same behavior. The neural networks can identify patterns associated with new cases by "remembering" identical patterns associated with cases present in the case repository, or recognize associations between related cases. If enough cases are presented with a set of similar attributes and values, the neural network can classify the cases in types. As in other domains, it is important that there are a great number of cases, allowing a good training of the neural network.

Normally, neural networks behave as true black boxes. Given a set of input values, a coherent set of output values is generated. In contrast to the symbolic processing paradigm very common in expert systems, using neural networks one normally cannot obtain an explanation of the results. Thus, a neural network will have difficulty in explaining why a particular conclusion was obtained. Having this in mind, and for now, neural networks must only be used when the explanation of the results is not so important, or can be obtained by some other means.

In the legal domain, the neural networks can be used to get fast predictions of legal cases outcomes. These predictions can be important when there is a high number of cases to deal with, and some of the human elements do not possess the necessary experience that allows them to make a quick preliminary overview of the cases. Another application of neural networks is in the construction of a model that identifies different outcomes for a particular case, associating each outcome with certain characteristics that may be present in each case (e.g. if the outcomes for the same case differ when a particular attribute of the case is changed, then it can be concluded that this attribute is important to the outcome of the case).

Figure 5 illustrates the use of Neural Networks to study legal cases. Fast predictions are obtained by using input signals obtained from the case to be studied. By manipulating the input signals describing a case, one can verify their impact on the outcome of the case.

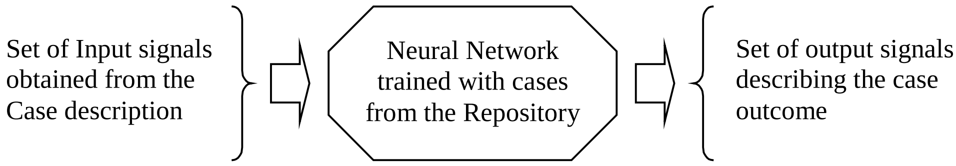

Figure 5. Using Neural Networks to Study Legal Cases

3.2 Case-Based Reasoning

3.2.1 Introduction to Case-Based Reasoning

The idea behind Case-Based Reasoning (CBR) (Aamodt and Plaza 1994) is simple: to solve a particular problem, analyze similar problems solved in the past, and adapt the old solution to the new one.

A case is the formal description of a problem and the respective solution. The starting point for CBR is the case repository, in which the representation of past cases is stored. The representation of a case includes some values that describe the problem and others that describe the solution.

CBR operates in the following manner:

- The characteristics of new cases are compared with the cases in the repository, and the most similar case is selected;
- The solution to the new case is obtained by adapting the solution of the past case to the particularities of the new one;
- The new case and the solution obtained are stored in the repository.

CBR cannot be applied to all kinds of problems. Some of the characteristics of the domains where CBR can be applied are:

- The existence of records of past problems solved in the past;
- The existence of previously solved cases is considered precious to the organizations;
- Specialists of the domain use examples in their discussions about the domain;
- The experience is considered as valuable as the knowledge contained in the manuals.

3.2.2 Using Case-Based Reasoning to study legal cases

Figure 6 illustrates the use of Case-Based Reasoning to study legal cases. The use of CBR in this context differs in some points from the way it is normally used. Here, CBR is used to study the cases, not to obtain their real outcome. The case outcomes are obtained in court, and CBR is only used as a decision support tool. Storing the new cases and their outcomes in the Case Repository is done at another level.

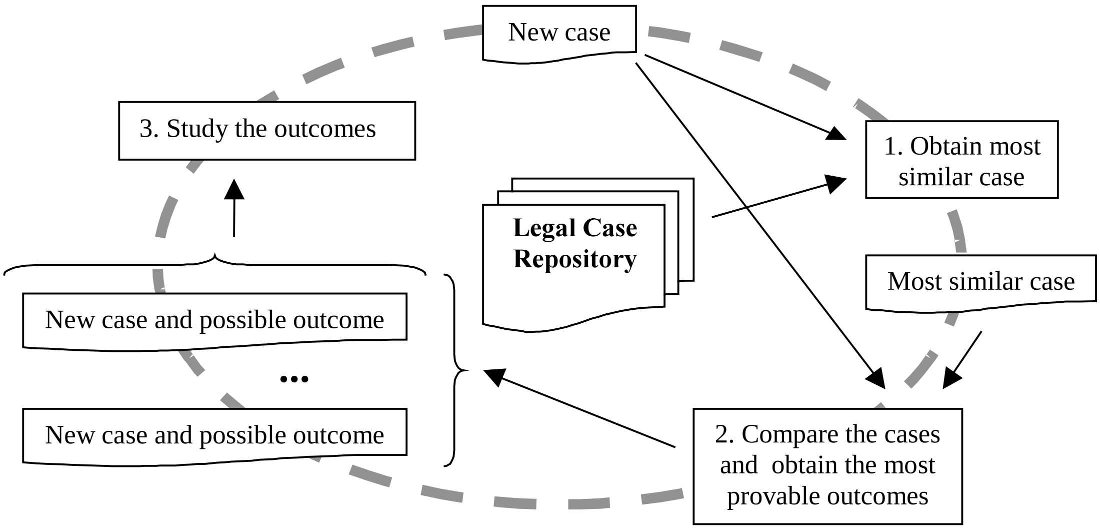

Figure 6. Using Case-Based Reasoning to Study Legal Cases

4 Users of the System

The different types of users will look at the system as an aiding tool to the tasks they normally carry out.

Students

Law students will be able to take advantage of the information contained in the system as material of support to their studies. The availability of a great amount of cases for consultation, as well as the possibility to conduct intelligent analysis of the cases, will help the students to best understand the spirit of the law (Ashley and Aleven 1991).

Judges

The duty of a judge is to pronounce sentences on the cases that are presented to him. Being able to count on a vast description of previously decided cases facilitates the resolution of the cases at hand, and leads to the pronunciation of fair sentences, supported by the analysis of similar cases.

Lawyers

The consultation of previous cases allows a better preparation for the defense of the cases in which the lawyers represent their clients. The analysis of similar cases will allow the lawyers to focus their attention on the aspects most relevant to their cases – those that will make a larger contribution to their outcome.

Others

There is everything to gain in making available information of a public nature, facilitating its consultation by everyone.

5 Conclusions and future work

This paper presented an architecture that allows the creation and use of legal cases repositories. Through the use of the WWW interface to the repositories, the data are available for consultation by anyone with access to the Word Wide Web. The second level of the architecture goes a step further, applying AI techniques that make intelligent use of the information available.

The next phase of the project will consist mainly in the development of the second level of the architecture, further studying of the use of several AI techniques (e.g. case-based reasoning, neural networks).

Acknowledgements

Marco Costa received financial support from the Portuguese Ministry of Science and Technology, under the contract Praxis XXI/BD/13650/97.

References

Aamodt and Plaza (1994)
Aamodt, A., and Plaza, E., Case-Based Reasoning: Foundational issues, Methodogical Variations, and System Approaches, *AI Communications*, 7(i), 1994, pp. 39-56.

Ashley and Aleven (1991)
Ashley, K., and Aleven, V., Toward an intelligent tutoring System for Teaching Law Students to Argue with Cases, in *Proceedings of the Third International Conference of Artificial Intelligence and Law*, Association for Computing Machinery, New York 1991.

Birman (1991)
Birman, Kenneth P., *The process group approach to reliable distributed computing*, Technical Report, Ithaca: Cornell University, 1991.

Bochereau, Bourcier, and Bourgine (1991)
Bochereau, L., Bourcier, D., and Bourgine, P., Extracting legal knowledge by means of a multilayer neural network application to municipal jurisprudence, in *Proceedings of the Third International Conference on Artificial Intelligence and Law*, Oxford: ACM Press 1991, pp. 288-296.

Branting (1989)
Branting, L.K, Representing and reusing explanations of legal precedents, in *Proceedings of the Second International Conference of Artificial Intelligence and Law*, Vancouver: ACM Press 1989, pp. 103-110.

Goldfarb (1981)
Goldfarb. C. F., A Generalized Approach to Document Markup, *ACM Sigplan Notices*, Vol. 16, No. 6, 1981, pp. 68-73.

Goldfarb (1990)
Goldfarb, C. F., *The SGML Handbook*, Clarendon Press 1990.

Herwijnen (1994)
Herwijnen, E. van, *Practical SGML*, Boston: Kluwer Academic Publishers 1994.

Hunter (1994)
Hunter, D., Looking for law in all the wrong places: Legal theory and legal neural networks, in Prakken, H., Muntjewerff, A.J., Soeteman, A. and Winkels, R. (eds), *Legal knowledge based systems: The relation with legal theory, Jurix '94*, Lelystad: Koninklijke Vermande 1994, pp. 55-64.

Jennings and Wooldridge (1995)
Jennings, N., and Wooldridge, M., Intelligent Agents: Theory and Practice, *The Knowledge Engineering Review* 10(2), 1995.

Jennings and Wooldridge (1996)
Jennings, N., and Wooldridge, M., Software Agents, *IEEE Review* 1996, pp. 17-20.

Navaratnam, Chanson, and Neufeld (1988)
Navaratnam, S., Chanson, S., and Neufeld, G., Reliable group communication in distributed systems, in *Proceedings of the 8th International Conference on Distributed Computing Systems*, San José: IEEE 1988, pp. 428-437.

Van Opdorp et al. (1991)
Van Opdorp, G.J., Walker, R.F., Schrickx, J.A., Groendijk, C., and Van den Berg, P.H., Networks at work: a connectionist approach to non-deductive legal reasoning, in *Proceedings of the Third International Conference on Artificial Intelligence and Law*, Oxford: ACM Press 1991, pp. 278-287.

Smith (1988)
Smith, J.M., *SGML and Related Standards*, Ellis Horwood 1988.

Warner (1990)
Warner, D.R., The role of neural networks in law machine development, *Rutgers Computer and Technology Law Journal* 16, 1990, pp.129-144.