SIREDOJ: a Legal Assistance Application about Contracts in the Building Industry *

Diego Bercovich

Seminario de Informática y Derecho
Facultad de Derecho
Universidad de Zaragoza
50009 - Zaragoza - Spain
Tel 34 + 76 + 55 59 80
Fax 34 + 76 + 56 58 52

Abstract

One of the current projects being carried out at the Seminario de Informática y Derecho (University of Zaragoza) is the design of a computer application called SIREDOJ. This application is aimed at assisting the user in the legal qualification of the case presented and at providing the documentation needed to justify the conclusions at which SIREDOJ arrives, in the matter of the contracts that can be signed in the building industry. This paper describes the characteristics of the legal area chosen and the representation and use of this knowledge in a hybrid object-oriented / inference rules environment.

* This work was supported in part by the Dirección General de Investigación Científica y Técnica (DGICYT). Project PB87-0632.

1. Introduction

For several years now the University of Zaragoza's Seminario de Informática y Derecho has been creating computer applications for the law community. The efforts concentrated in database construction, containing legislation, court decisions and doctrinal bibliography, and in the building of expert systems in some areas of civil law, developed in languages such as LISP or PROLOG, or shells such as CRYSTAL.

In this paper we present SIREDOJ, an application which has characteristics of both the expert systems and the database management systems. SIREDOJ simulates the activities of a legal expert who advises the user on the theme of building industry contracts, assisting him in the qualification of his problem, arriving at one or several possible legal conclusions and furnishing him with the documentation needed to justify the conclusions, which is extracted from a database that is external to the application.

The session that the user establishes with SIREDOJ begins with the identification of the parties (e.g. landowner, architect) and the characterisation of the contract signed (e.g. a contract for the designing of a project, for the

supervision of the building process). Depending on the type of the contract, the application verifies the existence of the corresponding clauses and the accomplishment of the obligations by the parties. Each alteration gives rise to a conclusion, such as "the architect may bring a claim for payment for the project". Each conclusion is supported by a set of legal norms, judicial antecedents and bibliographic doctrine which may be retrieved form the database, with the aim of this being pleaded in a judicial process [Galindo 90].

2. The user

As with every application development, we began by identifying the possible user. The first approximation was that anyone who is related with the building industry is a potential user of SIREDOJ, for, as we stated earlier, SIREDOJ represents the advice given by an expert lawyer to his client. Unfortunately, since SIREDOJ cannot acquire the necessary knowledge in order to understand the user problem in everyday concepts, it must share the qualification tasks with the user. This restricts the user field to those capable of qualifying a case from a legal point of view, though not being able to establish a legal conclusion in the building industry field.

The users of SIREDOJ are then: those lawyers or professionals working in the field of law who are not experts in this theme, advanced students of law or training schools for the judiciary (because of the possibility of explanation of the argument process), teachers, etc.

3. The legal area

The building process involves several parties who establish many kinds of contracts. In this area we find, with respect to the possibility of bringing legal actions, a group of legal circumstances and events such as breach of contract -both in the planning and in the construction stages-, or the appearance of defects in the work carried out.

The area that has been chosen is a very rich research field for a spanish jurist, because of the existence in parallel of regulations and jurisprudence which sometimes contradict each other.

In this subject matter we find legal norms and regulations (for example, the general law of contracts, urban planning legislation, rules provided by established jurisprudence), professional norms and regulations (for example the rules which govern how a project must be developed), court precedents with respect to the given case, and the applicable doctrine.

It is interesting to note that the chosen subject matter is in a state of constant change. Within it, we can see a phenomenon which has been studied by, for example [Hart 61] or [Viehweg 84]: changes which take place, over a period of time, to the meaning of the terms included in the rules. With respect to the subject matter which we are working on, we could quote the concept of *ruin* as an example, which nowadays covers more situations that it did when it was first introduced into Spanish Law.

In earlier times, ruin could only be established if the building had suffered from some collapse or caving-in. Today, according to established jurisprudence, it is possible to classify as a ruin a building which is inadequate for the use it was originally intended for (by virtue, as in the example above, of the presence of dampness in a fur wharehouse).

Thus we have a body of legislation which states that there is ruin only if there is collapse or cave-in, this being opposed by a body of judicial precedent which further states that ruin exists if there is an inadequate aspect, such as the one described above, or if there are defects which exceed normal imperfections, etc.

We can see that it is possible, within the same problem, to obtain different legal conclusions according to whether the criteria of judicial precedent is used or not. In our design, we have opted for the inclusion of these new criteria in the knowledge base. This inclusion means that the base does not possess the quality of overall consistency, it being limited to the existence of groups of local consistency [Bridgeland 90].

From the point of view of reasoning, this phenomenon of local consistency causes the existence of various lines of reasoning possible with respect to the same case leading, therefore, to conclusions which might be different or even contradictory [Vey 89].

Another peculiarity of this subject matter is the possibility of obtaining legal conclusions which do not contradict each other, even in simple cases. For example, given the case of a buyer of an apartment in a building in which construction faults appear, it is possible to invoke jointly: i) the specific building legislation, ii) the general legislation governing breach of contract, or iii) the consumer protection legislation.

4. The knowledge analysis

The work of gathering together the knowledge, a task carried out jointly by jurists and computer specialists, was divided into stages in order to simplify the work and to

distinguish between the different types of knowledge, according to that established by [Susskind 89], with his differentiation between academic legal knowledge (norms, judicial precedents, doctrine) and experiential legal knowledge (rules derived from the expert's own experience, common sense, etc.).

The first stage consisted of the extraction, from both the texts and the experts' own knowledge of the theme, of the so called **objects of knowledge**, made up of small portions of expertise, both academic and experiential. When we had gathered together a significant number of objects, we proceeded to study the possible existence of common patterns of expression, that is, the *manner* or *structure* in which the knowledge was expressed.

Some objects adjusted perfectly to the **inference rule** paradigm, where there is a group of antecedents and one or more consequents. For example, "*If* a building contract exists, the work consists of the construction of a building, the work has been completed, delivered, received, but not paid for, *then* it is possible to bring an action claiming payment for the work".

Another of the patterns expressed the knowledge which establishes relationships between **classes** and **sub-classes** in objects such as: "The following are classes of building contracts: a contract to plan the construction of a building, a contract to build a building,...", or "An agreement can be in writing, verbal or by way of document".

Some of the objects we studied expressed the **parts** or the **components** of something. For example: "The following are elements of a project: the pre-amble, the abbreviations, the plans, the budget, the calculations and the documentation".

Other objects expressed the **phases of a process**, "The phases of construction of a building are: the study of the plot, excavation, cementation, reinforcing, bricklaying, plumbing,...".

Finally, with respect to the objects, we encountered patterns corresponding to the **definition** of concepts, such as the earlier mentioned definition of ruin.

The second stage consisted in the reinforcing of **clusters of knowledge** [Meyer 89], which consist of small universes where the *contents* of the earlier mentioned objects have a high interrelation rate. The content of the clusters helps us to better handle the reciprocal interactions between the objects and to distinguish between the different normative types (in the broadest sense) which exist.

For example, let us study the work of a planner whose object is to carry out a project. In this cluster we can distinguish three objects: the planner, the process of developing the project and the project itself.

i) The planner has characteristics such as his personal data or his professional qualifications.

ii) The development of the project consists of different phases: the study of the problem (e.g. the needs of the client); the study of the regulations corresponding to the object to be constructed (urban planning regulations, for example); the study of the rules and the procedures to be followed in the process of the construction of the object; the study of the antecedents of the object to be constructed (buildings with similar functions or characteristics); the study of the antecedents of construction processes (for example, similar construction techniques); and carrying out the project.

iii) The project, made-up of the elements which we listed above: the pre-amble, abbreviations, plans, etc.

Governing certain of the characteristics of these objects, we find the corresponding norms: i) the professional qualification of the planner is regulated: he must be an architect, an engineer, etc.; ii) the process of developing the project is governed by regulations; iii) the contents of a project and its organisation are also regulated.

As we had said earlier, the development of the clusters of knowledge and the process of distinguishing between the diverse normative sources, have been a great help in our work. When the user presents the case, if we understand this as a machine whose equilibrium has been broken, then the task of searching for the fault is simplified when the user is asked, "that which refers to the development of the project, is it correct?". If, in this cluster, there have been no problems, then it is put to one side without intervention.

5. The software tool

As we stated above, the Seminario has developed several expert systems, whose knowledge representation schema has been rule-based. Due to the limits that this paradigm imposed on the expression of the knowledge, we decided to build this application using a tool with more expressive power, without losing the advantages of rule-based systems. We opted for KAPPA, which is a tool that combines the rule and the object-oriented paradigms, in a unique environment.

6. The design of the application

We will begin to describe here the internal structure of the application. SIREDOJ is composed of three areas: i) the **agents** area, where the legal knowledge resides, ii) the **possible objects** area, where we can find the representations of all the things that may be known to the expert (e.g. the different types of contracts or subjects who can sign them), and iii) the **scenario**, where depending on the case being presented by the user the actual objects are instanciated from the possible objects area.

With this structure we establish a clear difference between what is the reasoning task of the expert and what are the elements which conform the possible situations. The idea is that, after the presentation of the case, on the scenario we have the elements which represent it. Other expert system's sections may take these elements and retrieve documentation, draw a writing, etc.

6.1 The agents area

The agents (also called actors [Agha 90]) are software objects, where each of them is designed to reason about a part of the problem and to solve it [Ishida 90]. The set of all agents represent the reasoning of the expert when solving a given case. In other words, we have split the expert's knowledge in small pieces.

Typically, the reasoning of each agent is composed of a small software program (called method in the object-oriented terminology). This method proceeds by asking to the user some questions, by firing inference rules, by passing the control to another agent (what is called message sending) or by interacting with the other areas (e.g. by creating instances of objects over the scenario or by filling their slots).

The set of the questions that may be asked to the user (and their possible answers) is contained in the so called slots. These slots are structurally part of each agent, as the methods are. Each slot has three parts: the question to be posed to the user, the set of possible answers, and the actual answer given by the user.

The inference rules have as antecedents possible states of slots (that is, possible answers given by the user). As consequents, the rules typically send a message to an object, telling him to start its reasoning, or to pose some question to the user.

For instance, in our application we have an agent charged of verifying the presence of the essential clauses in the contracts for designing a project. Its reasoning method proceeds by asking the user: "Does exist in the contract a clause that specifies the obligation of making the project?", or "Does exist in the contract a clause that specifies the obligation of paying when the project is delivered?", etc.

As we stated above, the inference engine is fired after each answer. In this example, there exists a rule that says: "If there is no clause specifying the obligation of paying, then send an activation message to the agent charged of concluding with the following message: the contract is not well formed, and the architect has grounds for a claim of type X". That rule will be fired so depending on the answer given by the user.

One of the main features of the object-oriented methodology is the possibility of arranging the objects in a taxonomy of classes and subclasses, where the characteristics of the upper objects are inherited by the lower objects. These characteristics, as we have seen, are expressed by methods and slots.

In our case, we have arranged the agents according to their legal field, but all under a general agent whose purpose is to define some methods which are common: one that serves to ask some question to the user, another that serves to fire the inference engine, another to erase the answers in the agent's slots (each time the session begins), etc. We have so two types of methods in each agent: that which represents its legal reasoning, and those which are inherent to the functioning of the agents.

There are some agents which are not strictly part of the legal reasoning, whose tasks are necessary to the application. We have an agent charged of concluding, as we have seen above; another related to the screen management (to show a window, to hide it, to display the presentation, etc.); another related to the information retrieval (to seek a given search key, to retrieve the documents, etc.).

From an artificial intelligence point of view, the reasoning is made using a forward chaining process, beginning and proceeding with the facts, arriving at the conclusions (following [Wahlgren 89] and [Fiedler 85], amongst others). The algorithm used is breadhtfirst, in order to allow more than one legal conclusion from a given set of facts. For instance, is very common in our application that, given a situation, is possible to conclude that one party has grounds both for claiming for the resolution of the contract and for claiming for its fulfillment. Each conclusion is supported by different norms and antecedents.

6.2 The possible objects area

This area contains the objects with which the agent area (e.g. the expert) may operate. From an external point of view, here we have the pieces of real world that may be instanciated on the scenario.

For example, at this level we have the class of contracts or the class of possible subjects (landowners, architects, builders, etc.). The class of contracts has subclasses like the one of the contracts for the direction of a building, or the contracts for its construction. The upper class has slots that represent the parties who sign the contract; the lower inherits these slots, and adds new ones, for instance, those that indicate the clauses of the contract or the obligations of the parties. Each subclass of contracts has control methods which establish certain properties of the contract, once it has been instanciated over the scenario and the slots been completed by the agents. These methods fill other slots, like the one that may indicate the party that has not complied with her obligation.

6.3 The scenario

At this level we meet the objects which are instanciated depending on the case presented by the user. The instanciation is produced by the agents over the possible objects. In other words, there exists an agent that, given a case where a contract of class X is signed, produces an instance of the class X contracts over the scenario. Other agents will be charged, the session proceeding, of filling the corresponding slots of the contract.

As we stated above, once the case has been introduced by the user, the scenario reflects the characteristics of the real world problem that the user has. We stress the difference with the possible objects area, that represents the external things which may be known by the expert and so may be instanciated in a given case.

7. The state of the application

SIREDOJ is developing following two axes: the first broadens its legal competence, the second adds more capacities to the application. Up to date, in the first axis, we have introduced the contracts that may be signed between the landowner or the promoter, on the one side, and the builder, the architect or the technical architect on the other. For each contract, SIREDOJ checks that it contains the essential clauses and verifies the accomplishment of the obligations. We have yet i) to introduce what happens once finished the contract, if a problem appears in the building (as the ruin cited above), that is, the problem of responsibility, and ii) to enrich the application with the concept of temporal prescription. In the second axis, we have the entirety of the legal qualification - conclusion section implemented, and we are developing: i) the information retrieval section, ii) the *What?* facility, allowing the application to describe the meaning of the questions being presented to the user, and iii) the *What if?* facility, this being implemented form the dialog obtained in the session established with the user.

8. Future lines of research

With respect to the inclusion of extra-judicial principles, this is one of the matters to be dealt with in future research plans. Its effective use in the reasoning chains will take place when required by the user, as will be the prohibition -or obligation- of taking a specified deductive route (which would correspond to an argument which is politically or socially inadequate -respectively desired-).

These interventions in the knowledge base, inspired by [Lenat 90], will be carried out by way of specific commands, in order to introduce assertions into the base, to reject specific judgements which are components of the reasoning, etc.

This utility will be carefully designed and implemented (following the observations of [Gardner 87] page 60) in such a way that the user will have to be conscious of the "risks" taken by interpreting the law predicates which make up the reasoning chain and by modifying it.

The experience gained when working on the engineering of knowledge has shown us that it is fundamental to construct, progressively, a methodology of extraction and implemtentation of the knowledge which has, as a starting point, the objects, the clusters and the agents . Work with other areas of law will enrich our methodology and will permit us to discriminate between what is appropriate to the theme of building contracts and what is more generally applicable to other areas.

In collaboration with the University of Barcelona we are studying an interface for the application in natural language. This work, starting with the pertinent morpho-sintactic studies, is aimed at establishing a contact both with the agent and the possible objects areas.

9. Conclusion

With respect to the legal area of the application, I would like to comment that we began by selecting portions of legal categories related to the building industry, with the

aim of translating them into an expert system. We saw then that the departing point must be the actual problems, that is, the conflicts that can appear in that industry, and from then to study the possible legal qualifications and solutions. In other words, the application's internal structure and the dialogs do not represent any established legal category, the represent the ways in which a conflict may be resolved.

From a technical point of view, working with a hybrid object-oriented / inference rules environment has given us more flexibility in the expression of the knowledge, what implies a reduction in the semantic gap between the actual knowledge of the expert and its expression in the application.

References

[Agha 90] Gul Agha, "Concurrent Object-Oriented Programming", Comm. ACM, Vol.33, No.9, September 1990, pp. 125-141.

[Bridgeland 90] David Murray Bridgeland and Michael N. Huhns, "Distributed Truth Maintenance", AAAI '90, Boston, MA, USA, pp. 72-77.

[Fiedler 85] Herbert Fiedler, "Expert Systems as a Tool for Drafting Legal Decisions", II Int'l Conf. on Logica, Informatica e Diritto, Florence, 1985, pp. 265-273.

[Galindo 90] Fernando Galindo, "Acceso a Documentación Jurídica Mediante Ordenador. Principios Generales", Congreso Internacional de Informática y Derecho, Buenos Aires, Argentina, Oct. 1990, pp. 418-439.

[Gardner 87] Anne vdL Gardner, "An Artificial Intelligence Approach to Legal Reasoning", A Bradford Book, The MIT Press, 1987.

[Hart 61] H. L. A. Hart, "The Concept of Law", Oxford, University Press, 1961, VII-1.

[Ishida 90] Toru Ishida et. al., "An Organizational Approach to Adaptive Production Systems", AAAI '90, pp. 52-59.

[Lenat 90] Douglas B. Lenat et al., "Cyc, Toward Programs with Common Sense", Communications of the ACM, August 1990, Vol.33, No.8.

R. V. Guha and Douglas B. Lenat, "Cyc: A Mid-Term Report", AI Magazine, Fall 1990, Vol.11, No.3.

[Meyer 89] B. Meyer, "The New Culture of Software Development: Reflections on the Practice of Object-Oriented Design", Proc. of the First International Conference on the Technology of Object-Oriented Languages and Systems, 1989, pp. 13-23.

[Susskind 89] Richard Susskind, "The Latent Damage System: A Jurisprudential Analysis", ICAIL '89, Vancouver, Canada, pp. 23-32.

[Vey 89] C.N.J. de Vey Mestdagh and G. Bos, "Conflicting Legal Opinions, A Model for Legal Knowledge Systems", III Int'l Conf. on Logica, Informatica e Diritto, Florence, 1989, T. 1, pp. 217-242.

[Viehweg 84] Theodor Wiehweg, "Tópica y Jurisprudencia", Taurus, Madrid, 1984, 2. ed.

[Wahlgren 89] Peter Wahlgren, "Legal Reasoning - A Jurisprudential Description", ICAIL '89, Vancouver, Canada, pp. 147-156.