A knowledge engineering framework for intelligent retrieval of legal case studies

Adel Saadoun, Jean-Louis Ermine, Claude Belair, Jean-Marc Pouyot

A knowledge engineering framework for intelligent retrieval of legal case studies

Adel Saadoun, Jean-Louis Ermine, Claude Belair, Jean-Marc Pouyot

---

Artificial Intelligence and Law, 1-27, 1997, Kluwer Academic Publishers

(Version française non publiée)

A knowledge engineering framework for intelligent retrieval of legal case studies

Adel Saadoun[*/**], **Jean-Louis Ermine**[**], **Claude Belair**[***], **Jean-Marc Pouyot**[*]

[*]*Scalaire,*
Rue Lafaurie Montbadon, 33000 Bordeaux

[**]*Commissariat à l'Énergie Atomique*
DIST/SMTI
Groupe Gestion des Connaissances
Centre d'Études de Saclay
91191 Gif sur Yvette Cedex
jlermine@tabarly.saclay.cea.fr

[***]*Éditions du JURIS-CLASSEUR,*
141 Rue de Javel, 75747 Paris Cedex

ABSTRACT: Juris-Data is one of the largest case-study base in France. The case studies are indexed by legal classification elaborated by the Juris-Data Group. Knowledge engineering was used to design an intelligent interface for information retrieval based on this classification. The aim of the system is to help users find the case-study which is the most relevant to their own.

The approach is potentially very useful, but for standardising it for other legal document bases, it is necessary to extract a legal classification of the primary documents. Thus, a methodology for the construction of these classifications was designed together with a framework for index construction. The project led to the implementation of a **Legal Case Studies Engineering Framework** based on the accumulated experimentation and the methodologies designed. It consists of a set of computerised tools which support the life-cycle of the legal document from their processing by legal experts to their consultation by clients.

KEY WORDS: Legal Databases; Information Retrieval, Artificial Intelligence, Knowledge Engineering, Document base.

1. Introduction

Most of large document bases have been so far developed with the only aim of being able to manage a considerable amount of material. Therefore we have seen the automation of existing documentation systems or the setting up of new systems based on existing methods of paper documentation.

When studying legal documents, the problem is seen differently, due to the specific characteristics of legal documents. Because of these differences, existing software and systems cannot be transposed. Therefore it is necessary to design another kind of system for case-study bases, taking into account content, organisation, analysis of the contents and a way of developing a system based on a purely cognitive methodology, which is close to the user's mode of thinking. As a result, this approach required a theoretical approach that implied the involvement of jurists, computer experts and specialists in Artificial Intelligence [Bourcier 1991].

The aim of the project was to develop a "Legal Case Studies Engineering Framework" (LCSEF), covering a number of computer tools federated by a method that is structured and designed to help, standardise and make a coherent approach in producing and using case studies.

In this article, we are interested in one of the components of the "Framework", in knowing how to construct "intelligent" user interfaces in order to help users look for information. Such user interfaces based on a relevant knowledge extracted from legal documents should be able to guide users in their search and help them to decide their approach to the information retrieval.

Our project contains several steps divided into two parts:

1) A study of the knowledge management issues surrounding the JURIS-DATA project. This phase resulted in a user interface designed to query a structured case-study base called JURIS-DATA (JD). By "structured" we mean any document base which has at least a classification structure which is relevant to (and representative of) such documents.

2) A generalisation of the user query interface applied to other legal document databases which are less structured than JD, or which do not have a classification structure. This interface was primarily based on a document classification; a methodology for establishing a classification structure has been elaborated and tested. This methodology required the help of a legal expert throughout its different stages. When no expert is available, various automatic techniques of extraction and data skeletons based on the documents are used. These techniques were borrowed from several fields of study: Linguistics, Artificial Intelligence, Information Science and so on.

2. Overview of JURIS-DATA

JURIS-DATA is one of the biggest and oldest French case studies databases which is available on the French Minitel network. It has been based on the abstract and analysis method, covering 450,000 documents taken from various legal sources. Here, we can find cases of jurisprudence[1], legal doctrine[2] and government opinions[3]. Due to the increasing numbers of case studies produced every year in France, it is almost impossible to develop an efficient documentary system without previously selecting the information to be entered.

2.1. Selecting the documents

Selection of documents for JD, notably for jurisprudence, must respect two criteria:

- The qualitative aspect: all legal decisions which are fairly limited in their use for the legal topics to be consulted must be eliminated;

- The quantitative aspect: all relevant but repetitive information which would cause an unnecessary accumulation in the database, thus inhibiting the emergence of information, must be eliminated.

2.2. Document Analysis

Once the selection has been made, and in order to guarantee a certain coherence of the abstracts, primary case studies are analysed according to pre-established structures called Analysis Structures (AS).

2.2.1. Analysis Structures

An AS is a division of the law into different steps starting from the law itself and finishing by the facts. It is a division of the legal science resulting from a legal classification established by the legislator for the different codes (the Civil Code, the Employment Code, the Penal Code and so on) and the doctrine which governs those areas which do not have a codified legal existence (business law, environmental law and so on). This classification is organised, like a natural classification based on five levels. [Vogel 1988].

The first level groups the different types of legal problems, also called legal topics (JD has about thirty legal topics): Insurance, Commercial leases, Construction and so on. The other sublevels are progressive refinements, composed of specialisation refined from previous hierarchical level. An example of AS is given below (Figure. 2.1) : the generic level identifies special parts of the legal topics, and is used as the general title of an abstract, the specific level is a decomposition of the generic one, an abstract can be referred by several items of specific levels. The classification continues, down to a "varietal" and "sub-varietal" levels (specialisation of the upper levels).

[1] A corpus of legal decisions handed down by different jurisdictions

[2] A corpus of legal opinions on legal problems

[3] A corpus of ministerial replies of the government to questions from public representatives

Legal topic: INSURANCE

Generic Level: GENERAL INSURANCE, OBLIGATION OF THE INSURER

Specific Level:
- ARTICLE L.113 OF THE INSURANCE CODE, EXCLUSION OF GUARANTEE
  - * FORMAL AND LIMITED EXCLUSION (YES/NO),....
  - * GENERAL CHARACTERISTIC OF THE EXCLUSION (YES/NO),....
  - * >>DEFINITION/DISTINCTION<< OF THE EXCLUSION,....
- ARTICLE L.113-1 ALINEA 2 OF INSURANCE CODE, EXCLUSION OF GUARANTEE
  - * >>INTENTIONAL/FRAUDULENT<< FAULT OF THE INSURED (YES/NO),...
- ARTICLE L.113-5 OF THE INSURANCE CODE
  - * EXECUTION OF THE CONTRACT WITHIN THE SPECIFIED PERIOD (YES/NO),...
- BEGINNING OF CANCELLATION OF >>CONTRACT/POLICY<<,....
- REFUSAL TO RENEW THE >>CONTRACT/POLICY<< BY INSURER,....
- >>RESPECT/NON-RESPECT<< OF ITS OBLIGATION (YES/NO),....

Figure 2.1: Example of Analysis structures in JD

The whole set of analysis structures, currently available in document form, is divided into about thirty different legal topics. Each one comprises about twenty ASs which deal with the joint legal problems. These ASs, used to create abstracts, were established by JD experts using the primary case studies. Using the legal decision taken as a basis, the analysts begin by enumerating the factual problems. A legal qualification of these problems helps them to determine which texts are applicable and to place them back in their legal topics in order to reach the top of a new AS. After, the analysts go on with the enrichment phase of the AS. Here, they add the following information: the relevant text used, a list of all the possible solutions that can be handed down by the court, the justifications of these solutions and the instructions aimed to the analysts when creating an abstract.

2.2.2. Primary Documents Analysis and Creation of the Abstracts

Analysts begin by locating the legal problem described in the primary case study which allows them to first determine the corresponding legal topic and then to identify the pertinent one. When using these Ass, analysts can create the abstracts. Thus one AS constitutes the backbone of the abstract to create. After having entered the information which is easily at hand in the keyboarding slip (the date the decision was published, the jurisdiction and so on), the analyst proceeds to determine the legal content, respecting the following constraints:

- One abstract per legal decision.
- One paragraph per legal problem.
- One sentence per legal concept.

In order to do this, the analyst begins by reproducing the top of the AS at the keyboarding slip. This part describes the legal topic in addition to the legal problem which arises. Then he determines the legal setting of the decision by referring to different kinds of forms proposed by the ASs. Each one is applied to a specific level (See Figure 2.1). These forms, which the analyst uses to create an abstract have two characteristics :

- Everything in capital letters must be included in the abstract,
- Everything in small letters makes up the instructions to the analysts and is not included in the abstract.

Conforming to these forms, the analyst will clarify the following information :

1) *The text used*.

2) *The solutions handed down by each court*. These solutions arise within the chain "SOLUTION OF THE COURT.(yes/no)". The part, yes/no, tells us if there has been a solution taken by the so called court.

3) *The justifications of court solutions*. These elements are presented under different forms: either by a simple choice in a list, i.e. >>choice$_1$ / choice$_2$ /.../ choice$_i$<<; either by making a choice in a list which completes a sentence, i.e. Sentence >>choice$_1$ / choice$_2$ /.../ choice$_i$<<; or a sentence preceded by a choice in a list, i.e. >>Choice$_1$ / Choice$_2$ /.../ Choice$_i$<< sentence (See Figure. 2.1).

4) *The Court Decision*.

5) *A small abstract in natural language* which follows certain criteria will complete the abstract.

The keyboarding slips are finally keyboarded and stored on JD. The corresponding primary case studies are either scanned or microfilmed. The digital images resulting from a scanning are also stored and made available to the user through the JURIMAGIS program. The link between primary and secondary documents is assured by a number, called "JD-Number", which is inscribed within a special fixed length data field of the abstract. An example of abstract is given in Figure 2.2. Thus an abstract is defined as the result of a case study analysis. It then provides a precise, concise and organised schema of the principal implicit or explicit legal concepts involved in the specific document to be analysed.

JURIS-DATA, Search No: 16 from 16/07/95 ========== (Analysts)
Your reference: Ermine
---------- Case Study 25, pg 1

Case study reference: 028591

APPEALS COURT, PARIS, CH. 19, SECTION B
from 04/11/1983

Construction, sub-contracting, claim for guarantee against the sub-contractor, admissibility (No); examination of the agreement signed between the architect and sub-contracting research consultancy, hand-written annotation which departs from the terms of the contract (Yes), annotation restricting the mission of the research consultancy to the co-ordination of the work, annotation discharging the research consultancy from errors in the co-ordination of the work (No), research consultancy fees, use of the term supervision of the work, notification; extension of other responsibilities of the project management of the architect in the sub-contract (No), Confirmation

(SIMART/CAGNI-RIGOTHIER)

Figure 2.2: Example of abstract in JD

After the indexing and integration phase into JD, the abstracts are finally ready to use. Querying the document base can be done in natural language from the IRS.* SYSDEX program developed by Scalaire. SYSDEX is a full text IRS, supporting natural language and boolean requests.

3. Knowledge Management Study

As we have seen, JD is not only a set of electronic documents, it is also a complex, evolving system with multiple aspects, the management of this system does not simply depend on one or two techniques but on an organised methodology. In order to better understand the problem, we began by studying a real "knowledge

* Information Retrieval System

management" problem : defining, perpetuating and transmitting the global knowledge accumulated over JD group working life. The models obtained allowed us to set up a systematic approach to the specific management problems of knowledge within the group, and thus be able to suggest solutions [Brunet 1994]. The task consists of several steps :

1) Activity design

Our first task was to put the activity into context in order to correctly understand the complexity of the problem of documentation as a whole. We then carried out a detailed functional analysis of JD data flows. This detailed view made by using the graphic language SADT. [IGL 89] reveals the existence of real knowledge or expertise, which was complex and structured, providing the know-how, specific to the JD unit. In addition, the functional analysis also revealed the stages where an intervention could improve the potential of the running system.

2) Important Factors Identification

Writing up abstracts is currently done manually. Although the analysts must follow the strict rules of abstract writing laid down by the AS, omissions, subjectivity, and jumping ahead to certain data can thus occur, falsifying a part of the analysis. In fact, knowing that the coherence, homogeneity and efficiency of JD depends on quality of abstracts, we therefore need to give the analysts adapted tools which minimise any risk of error.

If the abstract and analysis technique adopted by JD, compared to the FULL-TEXT technique, is better at standardising the vocabulary, it also makes it inconvenient for the non-specialist user who does not know which terms were chosen in structuring the abstract. So, in front of an empty screen, it is important that the user has already analysed the problem and can access the language needed for the search. To avoid this possible absence of preparation needed by particular users, especially novices to the system, it was necessary to develop an "intelligent" user interface. This interface should guide users in their search and help them in their analysis of the problem.

Taking these observations into account, the intervention stages determined were :

- Developing tools to help analysts in creating the abstracts
- Developing intelligent user interfaces better adapted to legal reasoning in order to help information retrieval.

We will be discussing this last point further in this article.

3) Solution Proposals

Because of the complexity of knowledge involved in JD, in analysis structures, in know-how of the JD group, in implicit contents of legal documents and so on, the knowledge engineering approach has been chosen. This methodological approach allows us to analyse a whole set of knowledge common to one or several specialists within a group and come up with a structured model which is coherent and operational. This kind of work leads to the design of softwares generically called Knowledge-Based Systems (KBS).

4) Cognitive modelling

For efficiency in artificial intelligence approaches, the necessity to structure the knowledge is getting greater. Thus a number of original knowledge engineering methods have been created. Although these methods were new, they have already proved to be efficient in the operational design of the KBSs. The best known among them are KADS (Knowledge Acquisition and Design System) [Hickman 1989) or KOD (Knowledge Oriented Design) [Vogel 1988], and these are currently the most widely used in Europe.

The KADS method helps to define the life-cycle model of a KBS. It identifies and describes a set of techniques and methods to construct the knowledge bases. KOD on the other hand does not try to define the complete life-cycle of a KBS, but is rather oriented towards the specification of the knowledge. It is a structured model which uses both techniques (interviews, collection of expertise and so on) and theoretical representations highly influenced by human sciences such as anthropology and linguistics.

For this project, we used the MOISE method (Organised method for expert systems engineering) designed by J.-L. Ermine [Ermine 93]. This method consists of software supporting a knowledge-based methodology based on the same concepts as KADS or KOD. The choice of MOISE was made because we simply have a better mastery of the method (developed by one of the authors of the present since 1989). The MOISE method is now included in a more general methodology for Knowledge Management, called MKSM (Methodology for Knowledge Systems Management).

In MOISE, the formal specification of the knowledge is made through an appropriate language. The part used in this project is divided into two components:

Static Knowledge : This is essentially a model of the domain of knowledge. This model is independent from its use. Therefore we have to specify the field of knowledge, that is, to define it in a rigorous, complete and coherent way. This means that we can create a common nucleus between several systems using this same static knowledge in a specific manner.

During this step, we have to specify the knowledge by objects and link them by strong semantic bonds using semantic networks. The main bonds used become the defining links, such as the ATO, for Attribute Of, and the classification links, such as AKO for A Kind Of.

Dynamic Knowledge : This is a representation of the specialist's strategy in solving the problem at hand. This is then the part which becomes the aim of the considered system. In general, it helps to frame the way the static knowledge will be used in order to solve one or several of the problems at hand. The formal language used for this modelling of knowledge is based on the "ergonomic description of cognitive tasks" [Scapin 1989].

The formal specification of this knowledge can be accomplished either by a formal mathematical language or by a graphic language [Alkhatib 1994, Charreton 1996].

4. An intelligent Interface to JURIS-DATA

4.1. Introduction

Working on the cognitive modelling and constructing the knowledge base took place in collaboration with Mr. C. Belair. Mr. Belair is the general secretary of JD, a legal expert and one of the designers of the JD document base. The first step in the task of Knowledge Engineering was one of knowledge extraction from the contents of the abstracts as well as the thinking behind the creation of such abstracts. This knowledge does not necessarily have legal parameters.

4.2. Knowledge Types in JD

Compared to several other legal documents databases, (E.g. "Court of Cassation", CELEX), JD is considered to be one of the most structured. In fact, a JD document or abstract is a succession of well-defined fields where each field has its own signification and its own informative qualities. The knowledge described by these different fields can be observed in three ways :

4.2.1. Legal Knowledge

By regrouping documents of jurisprudence, doctrine and government opinions covering all branches of law, the first function of JD is to take all the different problems encountered into account. According to JD's designers, structuring the documents into a reasonable number of generic fields is the best adapted solution in order to clarify the legal knowledge. Within these fields we find :

1) the JUR field indicating the jurisdiction in which the decision was handed down, its location and formation (name and number of the chamber) ;

2) the ABS field which summarises the legal content of the primary case study, this being the most informative field of the abstract, structured into paragraphs and translating the legal analysis based on the ASs ;

3) the RES field which contains a short summary in natural language briefly describing the factual contents of the corresponding primary case study.

4.2.2. Documentary Knowledge

For this kind of knowledge, JD is really no better than other document bases. The same kind of typical information can be found in the specific fields, such as the date the abstract was created, reference to the primary case study and so on.

4.2.3. Heuristic Knowledge

By heuristic knowledge, we mean all kinds of knowledge that describe know-how or specific expertise of JD designers. Looking at the manner in which the ASs are classified, we see that this classification is not strictly a legal one. By this, we mean that a strictly legal classification is one as described by the legislator. The classification is based in part on the heuristic knowledge. In fact, the division of the science of law adopted by JD is in general rather arbitrary except for certain known legal topics which are unambiguous. This classification, described by an ensemble of thirty legal topics are constructed on several criteria :

- Taking into account the legal classification as prescribed by the legislator in a code (the Civil Code, the Trade Code, the Penal Code, and so on),
- Taking into account the doctrine which deals with subjects that do not have a legal codified existence (Business Law and Environmental Law, for instance),
- As it is a classification of legal decisions, a legal topic is never defined "a priori" but rather based on qualitative and quantitative criteria of the documents, that is, the quantity and nature of the texts found in the database,
- For efficiency, it is not useful to give legal topics that are too generalised as it can cause confusion. A legal topic should be specific enough to better analyse the legal problems which affect it.

The whole set of these legal topics constitutes the first level of the classification, which underwent a bit of refining. In fact, a division within legal topics was made. Each topic had to be redefined by a set of ASs, these being as numerous as legal topic was complex. These ASs, which give a hierarchical chain of concepts, allowed us to establish a judicial field to the meticulous detail. All these criteria were based on thirty legal topics only. Considering the evolution and diversity of legal problems, this number could be subject to change. Consequently, it depends on the evolution of the JD database.

It is important to remember that all knowledge, would it be heuristic, documentary or legal, has to be taken into account when specifying the static knowledge.

4.3. Static Knowledge Specification

An efficient and coherent model of the static knowledge contained in the JD documents must adhere to the following principles :

- Identify the objects which represent the present concepts, whether implicit or explicit, within the documents ;
- Bring up the relevant concepts for the information retrieval. Concepts which can be used to classify the documents in various ways and according to several points of view. These legal concepts can also be extracted from the ASs ;
- Bring forth the relevant factual concepts and try to attach them if possible to legal concepts.

Taking these principles into account, in addition to conforming to the specification method of MOISE, a coherent representational model of knowledge is indispensable. We opted for the "semantic network" model in order to establish the specifications of this static knowledge. (See § 2.2.4)

A representative and complete model of the static knowledge was progressively built up. We started by constructing the corresponding model for each concept identified (not necessarily a legal one). The structuring of the jurisdiction concept within a semantic network is given in Figure 2.3. This is the knowledge described in the JUR field of the abstract.

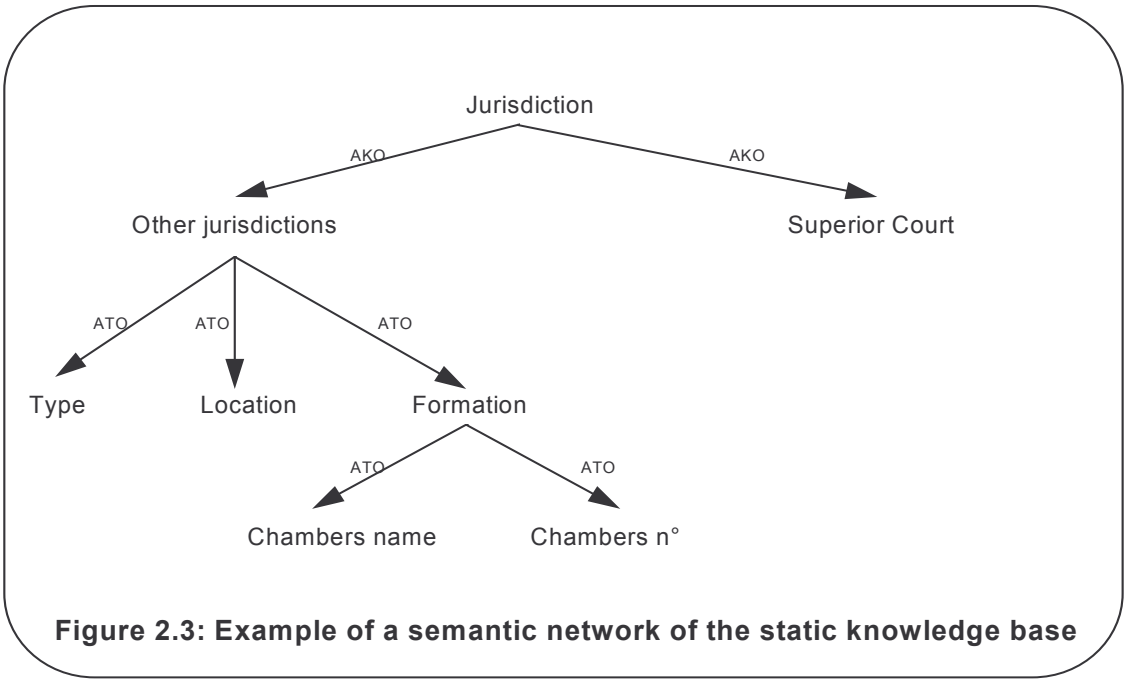

Figure 2.3: Example of a semantic network of the static knowledge base

The semantic network in Figure 2.3 can be read in the following manner :

Two types of jurisdictions can be distinguished, one for the Superior Court jurisdictions (Court of Cassation and the French Council of State) and the others. The latter are characterised by their type (Magistrates' Court, Police Court, and so on), their location (Paris, Bordeaux and so on) and their formation. The formation is itself characterised by two attributes, the name of the chamber (civil chamber, chamber of commerce and so on) and its number.

The AS classification was also structured within a semantic network. Because of space limitations, this network will not be outlined in this article, though this one does undoubtedly form the backbone of the static knowledge base.
By proceeding in this way for each concept identified, we finished by constructing a whole network of knowledge, and in this way we built the static knowledge base.

4.4. Dynamic Knowledge Specification

The dynamic knowledge specification consisted of modelling, using a formal language, the tasks that any user would have to perform. Contrary to the cognitive ergonomics approach [Scapin 1989], the JD dynamic knowledge model was not constructed by observing the behaviour of any user, but rather that of a legal expert, Mr. C. Belair. Choosing to observe the strategies of an expert, we hoped to optimise the use of the knowledge that had been identified on the static knowledge base and to provide users a real "intelligent" interface which would help in their information retrieval.

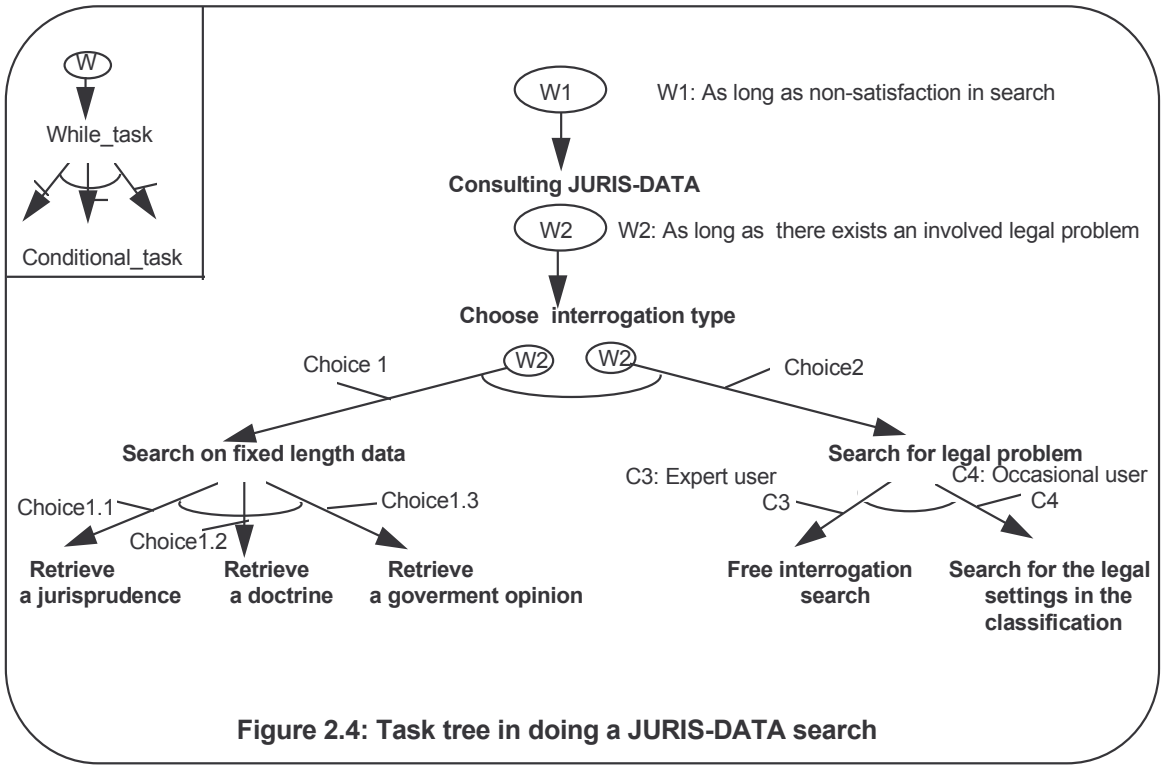

Figure 2.4: Task tree in doing a JURIS-DATA search

The specification of the dynamic knowledge was made using a formal language called the "task language". This followed a descending type of analysis. The starting point is the whole task to be performed by the system. This task is then divided into parts or sub-tasks which are themselves divided into other sub-tasks. The process continues until it creates a task tree whose final nodes are the finished tasks which determine the specific interactions within the environment. Task control was included in the decomposition, for example, ask a question or search for a corpus of case studies , and so on. Each node of the tree represents a task of the strategic plan of an expert in order to solve the problem at hand. In our case, it is the search for a case study.

We can distinguish the following types of tasks :

- The sequence_task, divided into several sub-tasks which must be executed in a specific order ;

- The conditional_task, divided into several sub-tasks which are subject to activating conditions ;

- The repetetive_task, described by a generic task and an organised list of objects which gives the successive entries within the generic task ;

- The while_task, similar to the "while...loop" in traditional programming languages.

A practical example of the specification of the dynamic knowledge is shown in Figure 2.4. A complete specification of the dynamic knowledge base would require about ten pages of graphic representations.

A brief description of this tree follows :

To perform a JD search, the designed interface proposes two options :

- Either a search on the fixed length data, such as the date the decision was handed down, the names of the parties involved, the jurisdiction which rendered the decision, and so on or a search for several specific legal problems.

For the first option, depending on whether they wish to search jurisprudence, doctrine or government opinions, users have three forms available with different data inputs. Each form allows to input the relevant information according to the type of document requested. These data input forms constitute a kind of assisted search mode in

addition to acting as a control mechanism which is both syntactic and semantic in order to guide in the search. In fact, by choosing this option, the user's task is limited to simple data input. Since it is up to the system to structure the data input requested, the user is almost assured of syntactically correct input. In addition, the fields which are proposed by a certain form coincide with specific documents only. Therefore, contrary to natural language searches, the legal parties involved in a case cannot be included in a search on doctrine for example, because of this semantic control system.

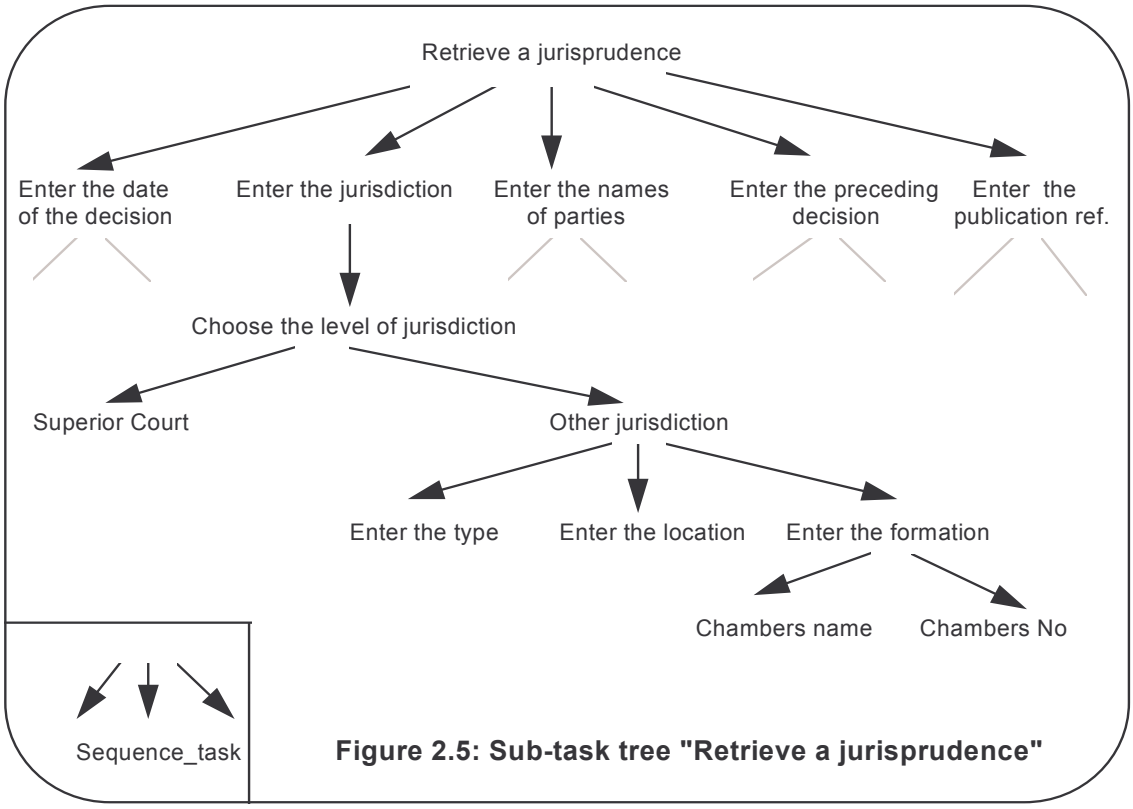

Figure 2.5: Sub-task tree "Retrieve a jurisprudence"

Example:

Suppose users wish to find all the decisions handed down by the civil chambers of Court No. 23 by the Appeals Court in Paris. By choosing the option, "Retrieve a jurisprudence", and entering in the jurisdiction field of the corresponding input form according to the task tree in Figure 2.5, the system begins by initiating the different nodes of the semantic network of the static knowledge relating to the jurisdiction concept before generating the appropriate request and beginning the search. (See Figure. 2.6)

When using the second option, searching for legal problems, both frequent and occasional users can make two kinds of searches.
If someone is a frequent user, a *free request* presupposes that the user will be able to correctly formulate a request.

An *assisted request* generally means that the user is a novice. Therefore, with the aid of the AS classification system, the user is guided to one of the legal headings which best describes the area needed. As all JD abstracts originate from a specific AS, and these have already been classified, by linking the JD documents to the AS classification a veritable taxonomy of the abstracts can be constructed. This in fact was accomplished without encountering any real problems. The specification of such an interface is shown in Figure 2.7.

At least with a basic idea of the problem to be solved, the user begins by giving up to three key words which are considered significant. The system will generate all the possible requests by a combination of these three words. Thus, the search has begun. The results are posted on the screen in the form of a tri-dimensional form, each line containing a combination of the key words used in the request, the number of case studies found, and the corresponding legal topics. These are nothing but the nodes of the first heading of the taxonomy of the abstracts. By taking into account the number of proposed documents and legal topics, the user can direct his search by

choosing the search heading which are considered relevant. A search heading corresponds to the whole set of key word combinations, number of corresponding documents and corresponding legal topics.

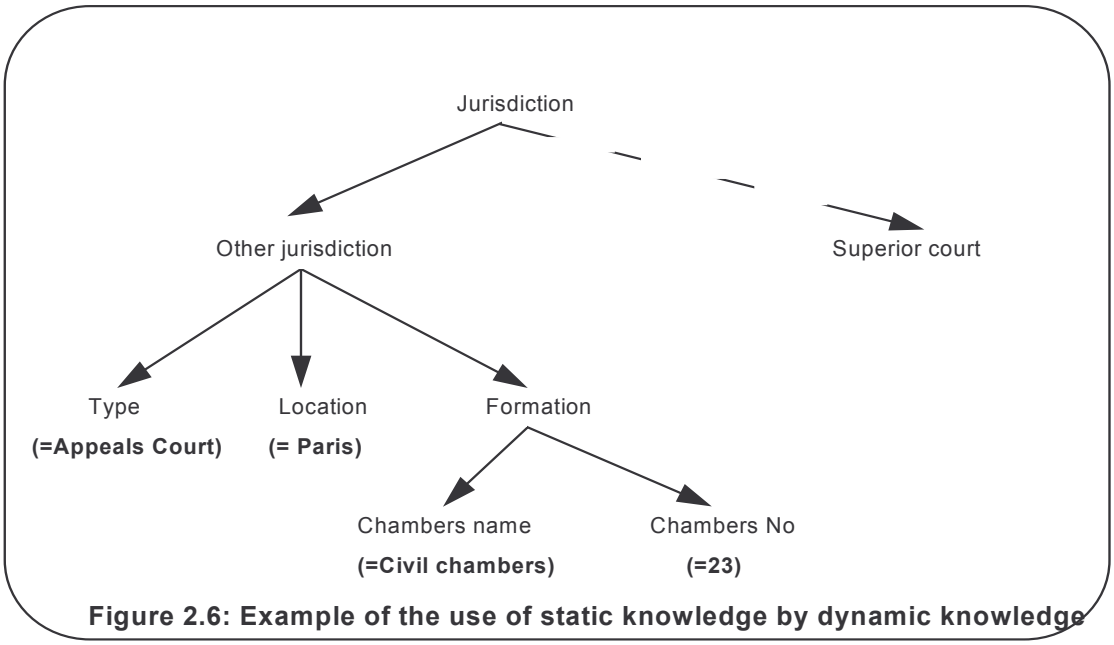

Figure 2.6: Example of the use of static knowledge by dynamic knowledge

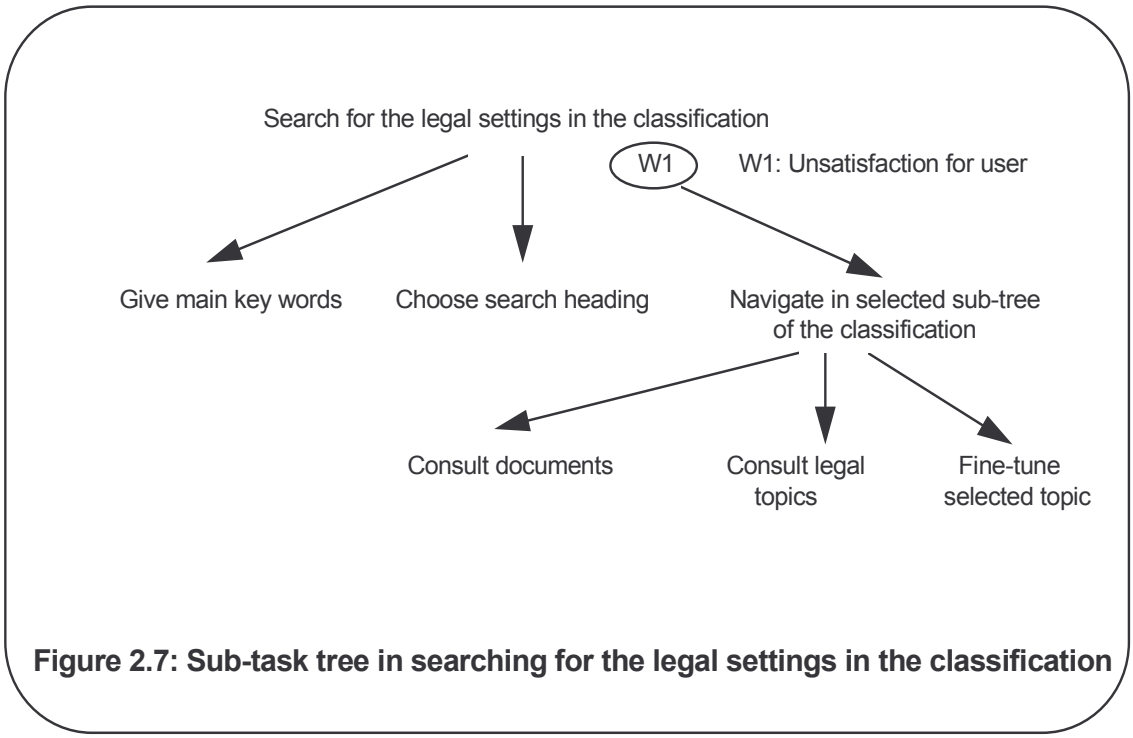

Figure 2.7: Sub-task tree in searching for the legal settings in the classification

Once the search heading has been chosen, the user can refine the search by exploring the branch of the classification system which covers the selected legal topic. At every sub-heading classification, a new, more precise request is generated and the search begins again. The user can backtrack and select a different legal subtopic.

The user can stop the exploration of the taxonomy at any time when the number of case studies found is sufficient to allow their consultation, thereby aiding in choosing the most relevant case studies. This navigation through the classification system expresses the legal reasoning an expert would make in formulating a request.

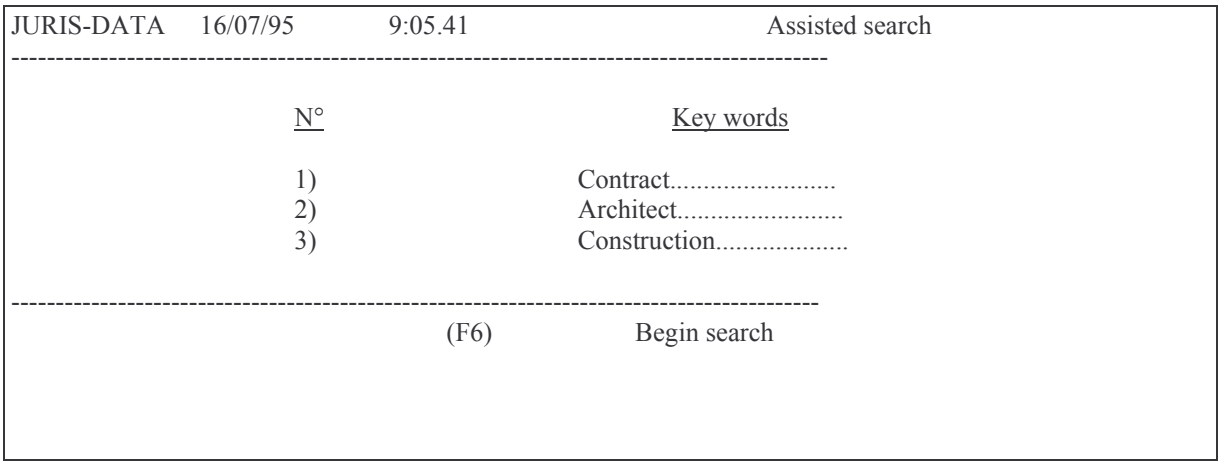

Figure 2.8 : Interface display corresponding to the sub-task "Give main key words"

In the example in Figure 2.8, the user has typed three key words: "Contract", "Architect", "Construction", which are echoed in the Key words window. The system has responded in three ways (see Figure 2.9). In the Request column, the system has displayed the conjunctions of key words used as requests. It has also printed the Number of documents that it has judged to be relevant. In the column labelled Legal topic, the system has listed the legal topics covering the documents found. The user can now choose one search heading.

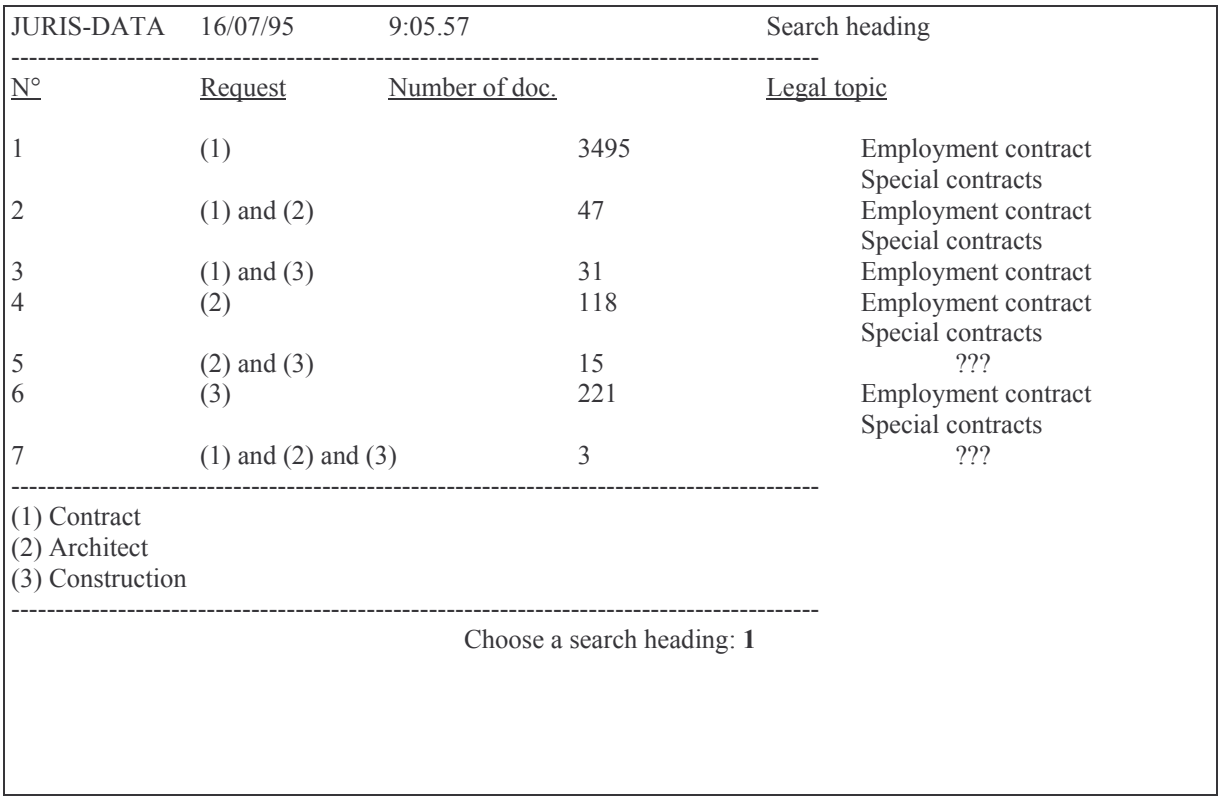

Figure 2.9 : Interface display after the user types three key words

In our example, the user chooses the first search heading (number 1) which causes a new screen to be displayed (see Figure 2.10).

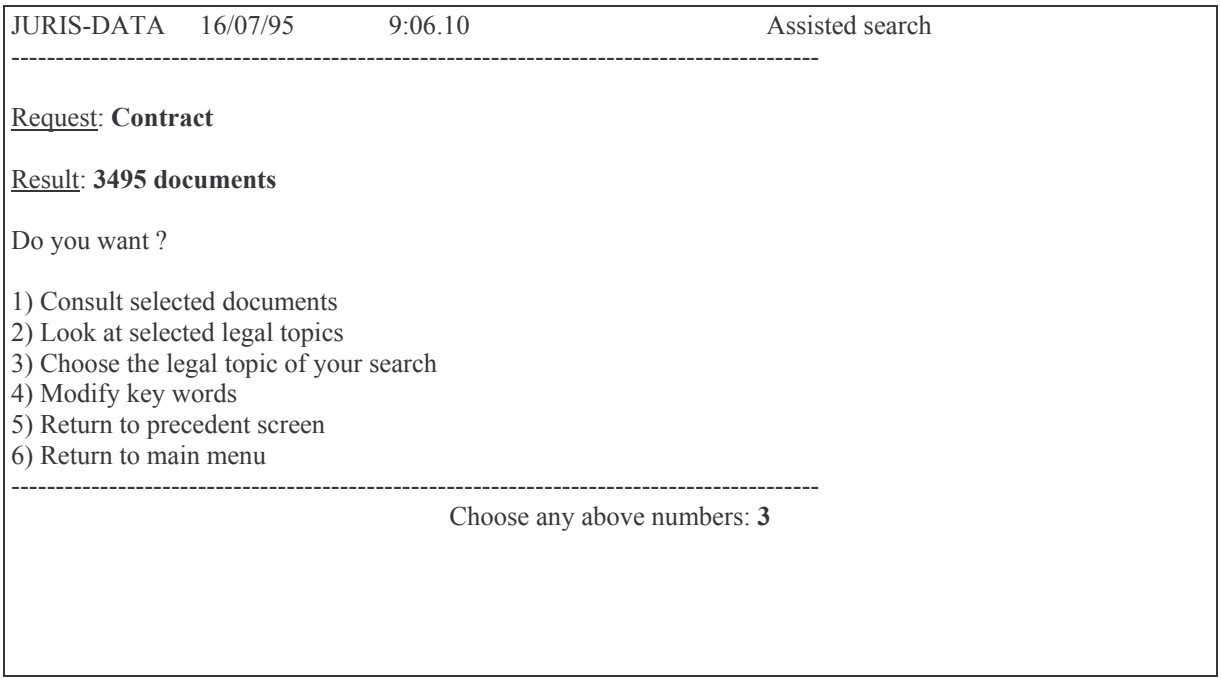

Figure 2.10 : Interface display after the user selects the search heading

The user now selects "Choose the legal topic of your search" (choice n° 3), and then selects the topic "Employment contract" (sea Figure 2.11), which leads to the display shown in Figure 2.12.

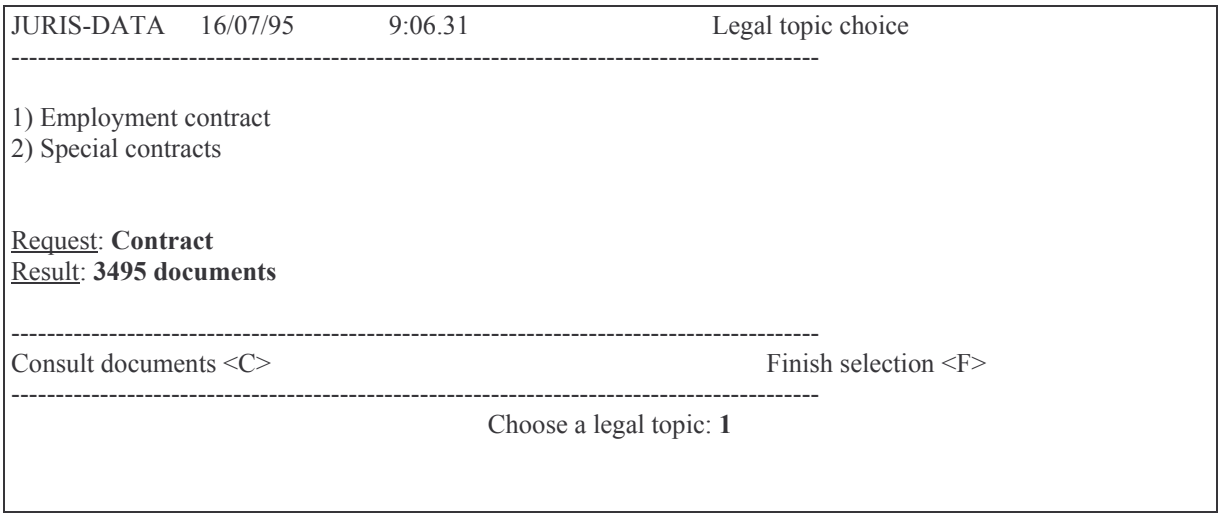

Figure 2.11 : Interface display after the user makes "Choose the legal topic of your search" choice

The user choose the subtopic "Apprenticeship contract" which causes a new subtopic level to be displayed. The new request and the number of documents found reflect the results of the user's two subtopic selections.

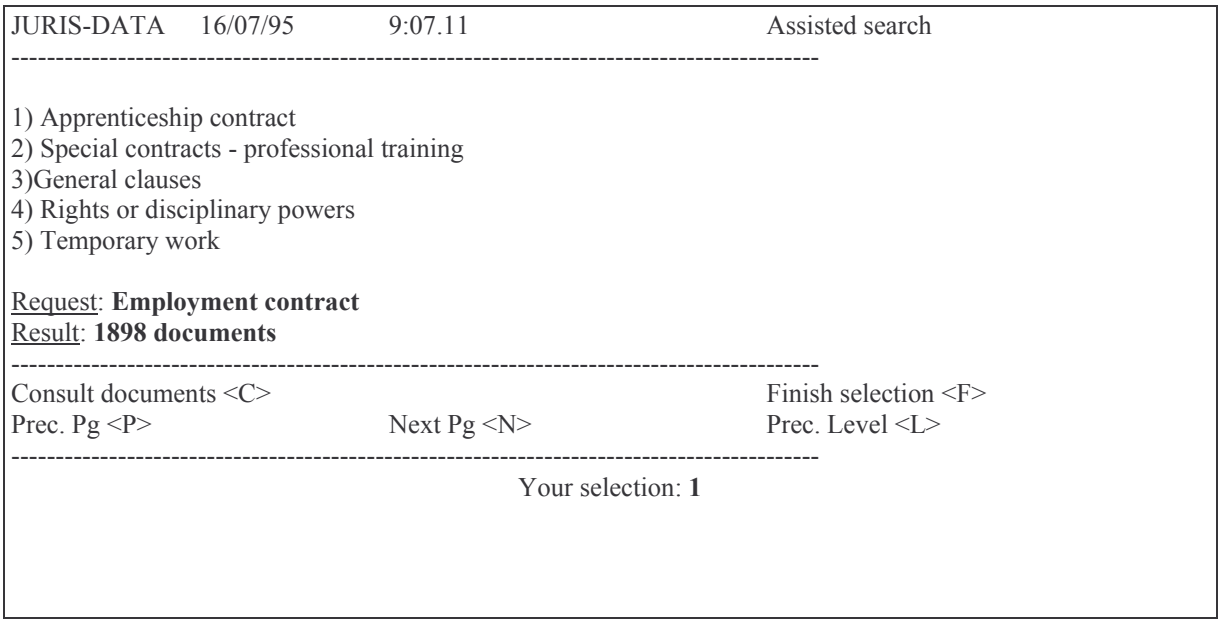

Figure 2.12 : Interface display after the user makes legal topic selection

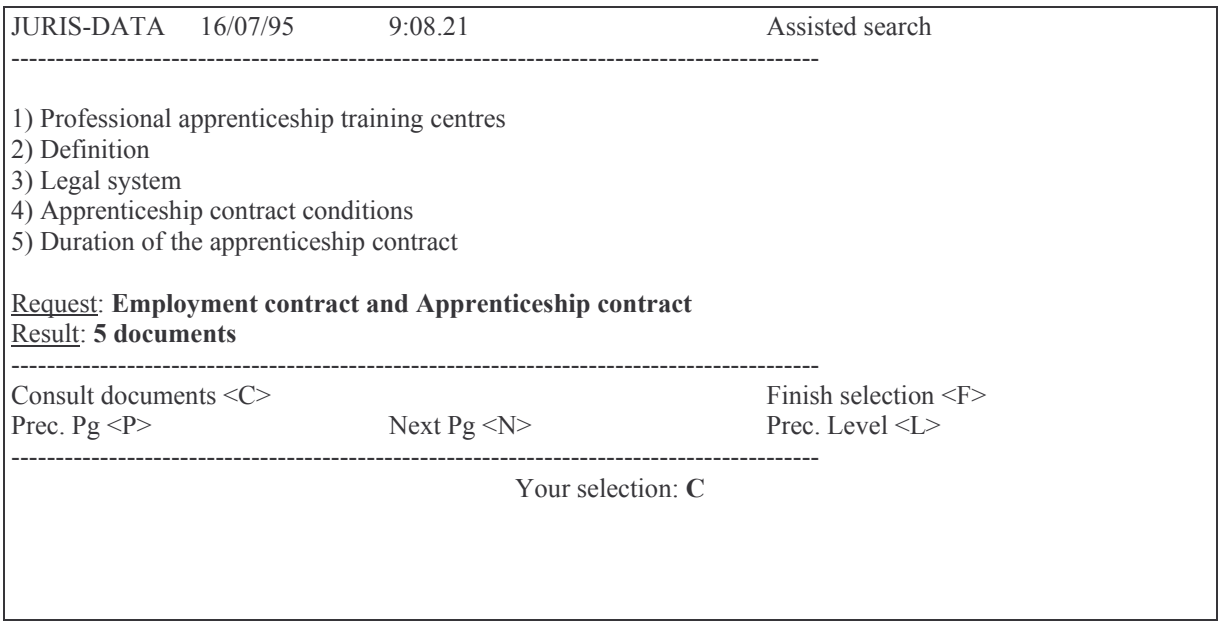

Figure 2.13 : Interface display after the user makes subtopic selection

The user does not need to scrutinise numerous documents; the subtopics navigation mechanism allows the user to easily descend to more specific levels, or backtrack to follow other parts. The number of retrieved documents changes accordingly, the more specific levels subtopics are selected, the fewer documents remain.

4.5. Conception, Coding and Validation

The first interface which originated from this specification was a generic programs instantiating semantic networks, by processing step by step the hierarchy of the network. A formal specification of a legal concept was entered into a module, leading ultimately to the corresponding semantic network. Making a request corresponded "grosso modo" to attributing values to one or several semantic networks through the use of a recursive module of a tree search. JD users considered this interface too sophisticated and incomprehensible.

A more flexible and comprehensible interface was then developed. The dynamic knowledge tree was faithfully coded in algorithms. The flexibility of use of this interface is due to the simplicity of its procedures : numeric

menus, data-capture forms, a few function keys, and so on. The goal of this interface is to be both efficient and as simple as possible. This simplicity is necessary for two reasons : first of all, jurists are not necessarily used to sophisticated graphic interfaces--such as the Windows interfaces ; secondly, this software is designed to function in a limited, non-graphic environment, the French Minitel network.

This interface was developed in C language on UNIX. It was tested on a significant part of JD : About 20 %, i.e. around 110,000 case studies arbitrarily chosen. Four legal topics, out of the thirty defined by JD, were successively examined and build the classification system on which the finished interface is based. The four legal topics are employment contracts, special contracts, civil liability and insurance. These four matters alone cover 60,000 case studies.

To validate our interface, we proceeded in two phases : validating completeness and validating correctness.

4.5.1. Interface Completeness

This validation aimed at making sure that all the tasks and sub-tasks of dynamic knowledge are really carried out by the interface. To do this, a direct correspondence was established between the various sub-tasks of the general tree and the screens and functions of the interface.

4.5.2. Interface Correctness

The goal here was to verify the appropriateness of different types of requests in relation to the problems which are the subject of these requests. To do this, the appropriateness of the requests obtained by using the interface was evaluated for each type of users. This evaluation was based on comparisons between requests obtained by interface users and reference requests determined by an expert. The results of the tests carried out were summarised in validation test-sheet of the type shown in Figure 2.14.

Actor	Expert	Initiated	Novice
Legal problem			
Free request			
Assisted request			

Figure 2.14 : Validation test-sheet

The columns represent three user types : a JD expert, an initiated user and a novice. The first line indicates the legal problem to be resolved expressed in natural language. The second line expresses the formulated request by using the "free interrogation" option of the interface, and the third represents the request obtained through use of the assisted mode. It should be noted that each validation test-sheet includes a reference request given by an expert, Mr. C. Belair, as well as the requests obtained by the user in response to a specific legal problem.

Example : Let us take the example of a novice doing research on an article of doctrine entitled, "The Presiding Judge". If our user uses *free request*, his request would be something like "I would like to have the article of doctrine entitled 'The Presiding Judge'". As we can see, this formulation is incorrect as the involved database registers are not pointed out (natural language is not sufficient, see request given by the expert). On the other hand, selecting the 'Retrieve a doctrine' option enables the user to produce the right request (See Figure. 2.15)

Actor	Expert	Initiated	Novice
Legal problem	**Search for article of doctrine entitled *the presiding judge***		
Free request	*AUT: doctrine AND TIT: The presiding judge*		I want to retrieve an article of doctrine entitled The Presiding Judge
Assisted request			AUT: doctrine AND TIT: The presiding judge

Figure 2.15 : Example of a validation test-sheet

Fifteen users more or less familiar with computers and/or law experimented in this way with the interface. This enabled us to create fifty validation test-sheets. Once completed, these test-sheets proved the overall correctness of our interface.

5. Generalization of the Query Interface for other unstructured legal document bases

We have seen that the query interface described above is founded on a knowledge base essentially built up from a classification of jurists' documents. In order to transpose the interface to other case-study bases, it is absolutely necessary to have the corresponding taxonomies. Unfortunately, some case-study bases, such as those based on the Full-text, have no classification structure. Consequently, it is necessary to determine an organised methodology of constructing classification structures of legal documents. This would make it possible to standardise the use of the interface for other case-study bases. Of course, there are tools designed to accomplish this type of task, such as the CABARET system [Skalak 89]. However, our wish to formalise the expert know-how and to master the tool led us to develop our own method.

5.1. Methodology for the Construction of Classification Structures

5.1.1. Introduction

The ultimate goal of this part of the work was to define a methodology for construction of judicial classifications by using the know-how of an expert. Building taxonomy of the reference field with an expert is a reliable and convenient way to approach the structure to be given to the knowledge base [Vogel 1988]. Furthermore, analysis and normalisation of the expert's resolution of the classification problem is by itself an organised methodology of classification structures construction.

We will now explain the steps of this method. It is interesting to note that this method is in fact a formal specification of the approach used by a judicial expert, Mr. C. Belair, to construct a classification of articles 229 to 310 linked to Title VI of divorce, Book I, concerning "persons", of the French Civil Code (Articles 229-310).

Before going any farther, we should give some definitions:

- *Primary Lexeme:* a term whose meaning cannot be deduced from its parts.
- *Secondary Lexeme:* made up of several primary lexemes. For example; a noun associated with one or several adjectives.
- *KWIC*: ***K**ey **W**ord **I**n **C**ontext*, one or several terms in the text. A KWIC can be a simple word, a primary lexeme, a secondary lexeme, a complete sentence or a phrase.
- *KWOC: **K**ey **W**ord **O**ut of **C**ontext*, a synonym or analogue of a KWIC in the text.
- *KWAC: **K**ey **W**ord **A**t **C**ontext,* which is an explicit description of an implicit concept.

5.1.2. Steps of the method

⇒*Step 1:* Extraction of KWICs
→ **Input:** Judicial documentary units to be classified
← **Result:** Judicial documentary units reduced to the KWICs extracted.

The expert begins by examining the documentary units* in order to identify the KWICs that it has judged to be relevant for the construction of legal classification. This first step relies essentially on the expert's ability to recognise the KWICs, thereby summarising in a sense the entire documentary unit.

⇒*Step 2:* Determination of the KWOCs.
→ **Input:** Judicial documentary units reduced to the KWICs extracted.
← **Result:** Judicial documentary units reduced to the KWICs extracted and corresponding KWOC.

The expert determines the corresponding KWOCs for each KWIC identified in Step 1. To do this, he can either use his own knowledge of judicial vocabulary or refer to a thesaurus. Experience has shown that several key expressions can have the same KWOC.

* In this case, a documentary unit is an article of the French Civil Code

⇒*Step 3:* Selection of Key Expressions
→ **Input:** Judicial documentary units reduced to the KWICs extracted and corresponding KWOCs.
← **Result:** Judicial documentary units reduced to selected key expressions.

The expert chooses the key expressions to be considered in the following steps from all of the KWIC and KWOC selected so far. These expressions will contribute to creating a classification vocabulary. Expressions which are not selected can be used to construct an alphabetical index.

⇒*Step 4*: Determining KWAC
→ **Input:** Judicial documentary units reduced to selected key expressions.
← **Result:** Judicial documentary units reduced to selected key expressions and determined KWACs.

The expert determines the KWACs evoked by each judicial documentary unit. Identifying a KWAC means either describing explicitly an implicit concept or grouping certain key expressions onto the basic relationship existing between them (link of causality, link of subordination, etc.) to obtain a new concept.

The aggregation of key expressions in KWAC consists of recognising discontinuities and/or semantic similarities. The expert goes across the list of expressions and detects a variation of meaning. This variation of meaning corresponds to a variation of "semantic traits" or "sememes" which enables him to elaborate a KWAC. The determination of a new KWAC by the expert is the result of the recognition of discrimination criteria between different key expressions. This phenomenon of semantic discontinuity, identifiable by a specific "sememe" is one of the basic elements of the theory of meaning known as semiotics, the theory of signs. This is called an "isotopic break" [Greimas 1966, Ermine 1989]. The "isotopy" is the redundancy of a semantic trait; the "meaning effect" is produced by isotopic breaks. A simple example is the sentence, "Everything is blue" (or "black"), which has two different meanings, depending on which trait is added. If the sentence, "I am in a bad mood" (which possesses a human trait) or the sentence, "In that room" (which possesses the non-human trait), is added, the isotopic break eliminates the ambiguity of meaning.

⇒*Step 5:* Construction of Classification Structures
→ **Input:** Judicial documentary units reduced to selected key expressions and to determined KWACs.
← **Result:** Classification structures

In order to construct classifications, the expert works by aggregation. This is an ascending approach, aggregating sub-groups into groups. It consists of first bringing the entities to be classified together into small batches, then regrouping the small batches in larger batches. This continues until finally there is one general group. This type of approach is based on the principle of "inductive generalisation", the rule of ascending generalisation defined by [Michalski 1983].

In general, the regrouping of sub-groups into groups is done on the basis of their mutual semantic relationships.

In our methodology, the classification is progressively constructed. First of all, the expert establishes the link between the KWICs/KWOCs maintained in Step 3, and the KWAC aggregated onto them. This is done following the principle of "isotopic break". Then links are established between KWACs of the same level and those of higher levels in the hierarchy.

This classification is the starting point for the construction of the knowledge base underlying the information query interface described above.

5.1.3. Methodology for constructing documentary languages from documents

We have seen a methodology to build a classification structure by analysing the texts. In that case, the classification structure is a kind of documentary language, enabling the intelligent indexing and consulting of documents. Another well-known problem is the construction of an index, another kind of documentary language for indexing and consulting texts. The above methodology has been adapted to index construction. The changes are straightforward, the first steps of the method are the same, and only the output is different.

5.2. Automation of certain steps of the method

As we have observed, the methodology of constructing classification structures relies essentially on the participation of an expert jurist. This fundamentally "human" approach represents a practical and reliable way of extracting all knowledge in a field and organising it in classification structures. Through the judicial knowledge of the expert, it is possible, for example, to identify implicit concepts or evaluate the relevance of a concept to a legal topic under examination. In our opinion, this cannot be done perfectly with automation, and we do not agree with G. Salton [1972] who maintains in his article that this is possible.

If an expert is not available, different automatic techniques of extraction and knowledge structuring can be used. These techniques often can give relatively satisfying results, although they cannot replace the expert. We will now examine some of these techniques.

5.2.1. Automatic extraction of KWICs

Certain tools of acquisition of terminological units, generally used for automatic indexing systems, can be used to automate this task.

The SPIRIT system [SPIRIT 1992] automatically produces a thesaurus and a list of key words and performs document retrieval [Andreewsky 1983a) and 1983b]. The extraction of the terminology used in the elaboration of the thesaurus uses linguistic methods:

- A morphological analysis searches for the lexical identity and the possible grammatical values of each word;

- A grammatical analysis eliminates ambiguities unresolved on the morphological level;

- A morpho-syntactic filter extracts from the corpus all linguistic configurations corresponding to a given morphology or syntactic structure. These results are then manually filtered.

Other terminological extractors such as TERMINO [David 1990] or LEXTER [Bourigault 1992] can also be used.

5.2.2. Automated classification structures construction

This section of the paper is not a complete study of types of classification and the various algorithms of automatic classifications (for this, consult a study such as [Benzecri 80] or [Vogel 1988]. The goal here is rather to propose several methods based on certain types of research, and to see how they can be adapted so we can automatically construct judicial classification structures.

"DISCAN", A Computer System for Content and Discourse Analysis [Maranda 1992]: This software designed by Pierre Maranda*, allows for two types of analysis of a textual corpus:

- A content analysis which breaks up the text into lexical units and then provides statistics on their frequency, the percentage of their appearance, and so on;

- A discourse analysis which provides a graph of concepts making it possible to determine the interaction which exists between two semantic components in the text.

Let us now take a more detailed look at how DISCAN creates this graph of concepts and how to transform it into a classification structure:

1) **Setting up the corpus for analysis**

Before launching a discourse analysis the text to be analysed undergoes certain preparations by the module of content analysis. A content analysis can treat either a raw text -the articles as they appear in the Civil Code- or a filtered text. In our example, the textual corpus will be made up of articles reduced to selected KWIC. Consequently, the lexical unit for DISCAN will be the KWIC.

* Professor of Anthropology, Laval University, Quebec, Canada

2) **Creation of a thesaurus**

This is the most delicate step of the method. The analyst associates one - and only one - descriptor to each lexical unit of the original corpus. The goal is to reduce the lexical diversities of the raw forms, the KWICs, to semantic groups - also called semantic fields. These descriptors are in fact nothing but the famous KWOCs and KWACs.

It is important to note that the relevance of the results obtained from the discourse analysis phase depends a great deal on the quality of the thesaurus. Therefore, the analyst should choose his or her descriptors very carefully according to the desired goal.

3) **Tagging the corpus**

This phase consists of replacing the raw forms of the corpus by the corresponding descriptors. The result of this substitution is a new corpus, generally called "Secondary Level" or "Standardised Corpus". Thus, the discourse analysis will bear on tags rather than raw forms.

4) **Discourse Analysis**

The analyst is now ready to start the discourse analysis phase. To do this, DISCAN uses the techniques of Markov Analysis. These techniques involve calculating the transition probabilities from one "*state*" to another. In this context, a *state* can be defined as a descriptor defined in the thesaurus. This phase does not analyse the raw forms of the original corpus, but rather their semantic content- expressed by the descriptors of the standardised corpus.

The result of this discourse analysis is a weighted and oriented graph of concepts in which the nodes represent the descriptors or concepts and the links express their coocurrence*. The score given to each link represents the probability factor of the succession of descriptors in the standardised corpus. The relative *position* of one descriptor to another is indicated by the arrow of each link.

5) **Construction of a classification structure**

A classification structure is set up from a graph of concepts by constructing the corresponding "*covering tree*". For a graph, a *covering tree* is a free tree connecting all the nodes. This step, not included in DISCAN, consists first of optimising the graph of concepts by eliminating undesirable links -cycles or links with low scores. Then, an algorithm to produce a *covering tree* is applied, which finally leads to a classification structure.

Other methods like semantic proximity [Barakat 92] can also be adopted.

5.3. Conception and Coding

A software framework using the two methodologies of documentary language construction (classification and index) have been implemented. It groups a set of modules, each of which carrying out one of the step shown above. The methodology is then semi-automatic, with a lot of facilities (text processing, database and legal thesaurus links, on line helps...). The programming language used for development in this software framework is C++.

A graphic interface conception tool called ZINC is also used. This tool enables the use of the software in several different environments: DOS, WINDOWS, MACINTOSH and UNIX.

6. A Legal Case Studies Engineering Framework (LCSEF)

The fundamental problem in any documentary system is to establish a connection between a user and a file. The goal is to make it possible to detect information and to select the document considered relevant. This type of system is largely based on two procedures: classification and information retrieval.

If we try to evaluate the solutions proposed above from a strictly documentary point of view, we can observe that they are integrated into a documentary chain which is a veritable LCSEF. For JD, the tool for help in writing abstracts, which we have not presented in this article, when associated with an indexing module, represents the *storage* link of a documentary chain described above. Use of the intelligent query interface makes for a better

* The notion of coocurrence corresponds to the spatial proximity between two descriptors on the linear scale of the normalized corpus

connection between the user and the document base. Through this interface users can retrieve the information that interests them.

We can thus offer archivists the skeleton (backbone) of a veritable framework of Judicial Documentary Engineering (See Figure 4.1) which is a tool for writing abstracts, as for JD, an intelligent query interface and tools supporting the methodology of documentary languages construction. The archivists will always be required to follow and update the case studies database.

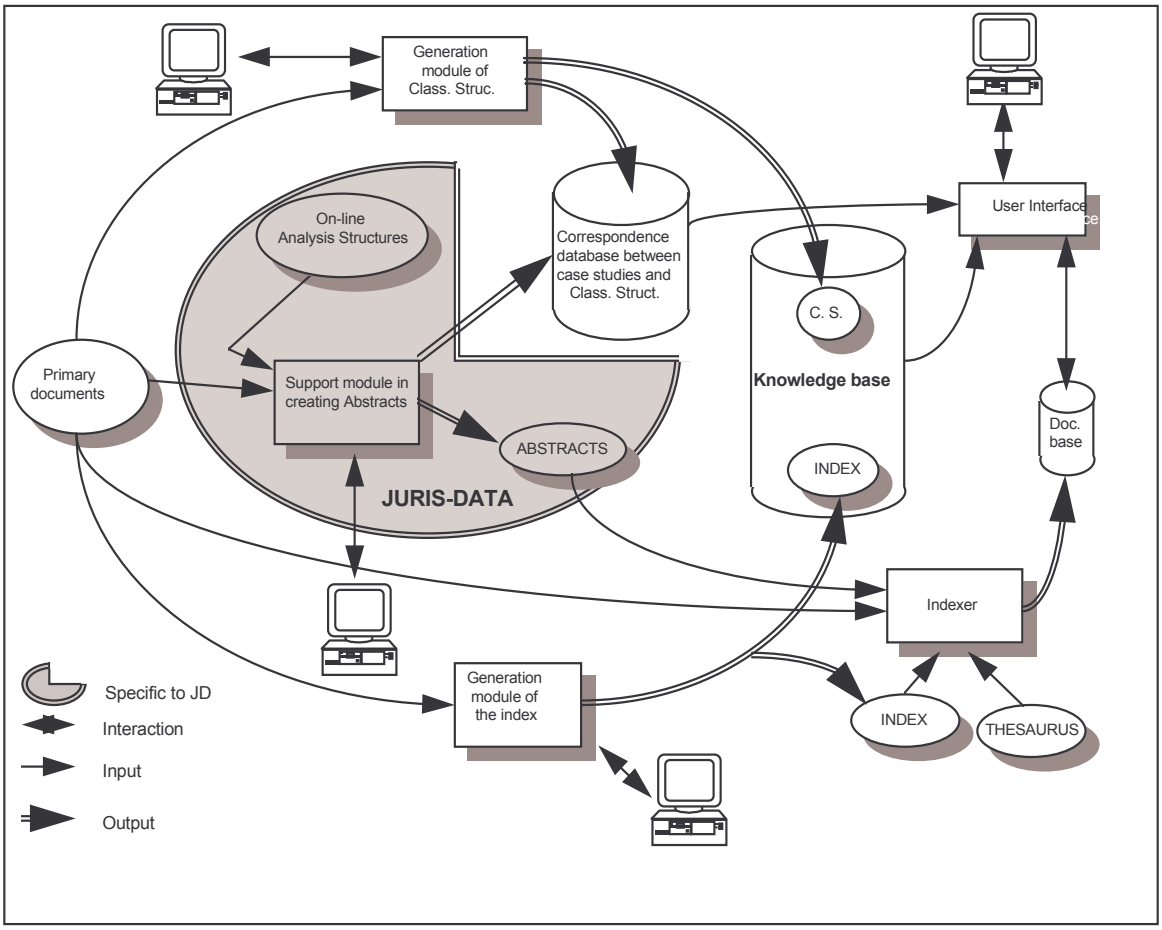

Figure 4.1: Architecture of the Legal Case Studies Engineering Framework

7. Conclusion and Perspectives

Legal documents - laws, judgements and decisions, articles and commentaries - are neither bibliographical documents nor commercial, scientific or technical documents, any software program created for this kind of documents cannot be easily transferable to the legal field.

The legal field, being rich in knowledge, means arbitrarily that archivists competence rely heavily on that bank of knowledge, and therefore any legal document system must be based on a knowledge-based approach.

Thus, in collaboration with Mr. C. Belair, we have developed the skeleton of a real Case Studies Framework which includes:

1) A support system in creating abstracts for the case-studies database (the JDs);

2) An intelligent, generic interface to help in searching for information. This interface is based on a knowledge base specific to the legal database we hope to manage;

3) For unstructured legal document bases, a semi-automated methodology of constructing documentary languages (Classification structures and index) from the documents was developed and implemented. Classification structures and index serve to set up the knowledge base to be used by the information retrieval interface. Such an engineering framework should serve to integrate the whole legal information chain from the production and management process (follow-up and updating) up to a more intelligent final product.

References

Alkhatib B., B. Bergeon, J.-L. Ermine, C-M. Falinower, M. Monsion : *Génie logiciel et Génie cognitif pour l'élaboration d'une base de connaissances en automatique*, 9ième Congrès Reconnaissance des formes et Intelligence Artificielle, RFIA'94, Paris 11-14 Janvier 1994, Vol 2, pp. 734-738, Paris, 1994

Andreewsky A., Binquet, Debili F., Fluhr C., Ponderoux : *L'interrogation en langage naturel dans le système SPIRIT*, Journées Internationales de l'Informatique et de l'Automatisme, pp. 322-332, 1983.

Andreewsky A., Debili F., Fluhr C.: *Apprentissage - syntaxe - sémantique lexicale*, Revue du Palais de la découverte, Vol. 9, N°83, décembre, 1983

Barakat Barbieri B. : Vers une c*onstruction automatique de graphes de concepts*, Thèse de Doctorat de l'École Centrale, 1992

Benzécri J.P : *L'analyse des données, Tome 1 : la taxinomie* - Éditions DUNOD, 1980

Bourcier D.: *Méthodes pour une approche cognitive du droit*, Les sciences cognitives en débat, B. Vergnaud Ed, CNRS Editions 1991

Bourigault D. : *Lexter, vers un outil linguistique d'aide à l'acquisition des connaissances*, 3èmes journées d'Acquisition des connaissances du PRC-IA, Dourdan, Avril, 1992.

Brunet E., Ermine J.-L : *Problématique de la Gestion des Connaissances des Organisations*, Ingénierie des systèmes d'information, Vol. 2, n° 3, pp. 263-291, AFCET/Hermès, 1994

Charreton B., J-L Ermine: From knowledge specification to executable specification, Knowledge Engineering and Modeling Languages, KEML'96, Paris, 15-16 janvier 1996

David S., Plante P. : *De la nécessité d'une approche morphosyntaxique en analyse de texte*, 25 pages, Rapport interne UQAM, Québec, 1990.

Ermine J.-L : *Systèmes experts, théorie et pratique*, Collection Tec et Doc, Lavoisier Ed, Paris, 1989

Ermine J.-L : *Génie Logiciel et Génie Cognitif pour les systèmes à base de connaissances*, Collection Tec et Doc, Lavoisier Ed, Paris, 1993

Greimas A.-J. : *Sémantique structurale*, Larousse, Paris, 1966 (repris chez P.U.F. Paris, 1986)

Hickman F.R., Killin J., Land L., Mulhall T., Porter D., and Taylor R.M.: *Analysis for knowledge based systems, a practical guide to the KADS methodology*, Ellis Horwood books in information technology, 1989

IGL Technology: *SADT un langage pour communiquer*, 1989 - Éditions EYROLLES

Maranda P., Nze-Nguema F-P. : *L'unité dans la diversité culturelle,* Les Presses de l'Université Laval, Sainte-Foy Québec, 1994

R.S. Michalski, J.G. Carbonell, T.M Mitchell. : *A theory and methodology of inductive learning*, Eds, Machine learning: an artificial intelligence approach, Tyoga, pp. 83-129, 1983

Salton G.: *A new comparison between conventional indexing (MEDLARS) and automatic text processing (SMART)*, Journ. of the Americ. Soc. for Inf. Sci. Vol. 23, N° 2, March-April, pp. 75-84, 1972

Scapin D., Pierret-Golbreich C. : *Towards a method for task description: MAD*, Work with display units 89, L. Berlinguet, D. Berthelett Eds, Elsevier Science, North Holland Publishers, 1990

Skalak D.B.: *Taking advantage of models for legal classification*, Second International Conference on Artificial Intelligence and Law, pp. 234-241, ACM New York, 1989

SPIRIT of SYSTEX - the linguistic skill, Rapport interne SYSTEX, Bâtiment Appolo, Espace Technologique, 91195, Saint-Aubin cedex, France, 2 pages, 1992.

Vogel C. : *Génie cognitif* - Éditions MASSON, Coll. Sciences cognitives, 1988.