

Handwritten signature or initials.

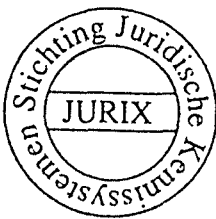
Legal Knowledge Based Systems
Aims for research and development

Legal knowledge based systems

Aims for research and development

Editors:

C. van Noortwijk
A.H.J. Schmidt
R.G.F. Winkels



1991

KONINKLIJKE VERMANDE BV, LELYSTAD
uitgevers sinds 1750



De uitgever is zich ervan bewust dat, ondanks de zorg die auteur(s) en uitgever besteden aan de samenstelling van de uitgave, onvolkomenheden kunnen ontstaan. Hiervoor kunnen de uitgever en de auteurs helaas geen enkele aansprakelijkheid aanvaarden. Voor suggesties aangaande verbeteringen van de uitgave houdt de uitgever zich aanbevolen.

Ontwerp omslag C Koevoets

ISBN 90 6040 948 5/CIP

© 1991 The Foundation for Legal Knowledge Systems

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vervoelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand, of openbaar gemaakt, in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnamen, of enig andere manier, zonder voorafgaande toestemming van de uitgever. Voor zover het maken van kopieën uit deze uitgave is toegestaan op grond van artikel 16B Auteurswet 1912¹° het Besluit van 20 juni 1974, Stb. 351, zoals gewijzigd bij het Besluit van 23 augustus 1985, Stb. 471 en artikel 17 Auteurswet 1912, dient men de daarvoor wettelijk verschuldigde vergoedingen te voldoen aan de Stichting Reprorecht (Postbus 882, 1180 AW Amstelveen). Voor het overnemen van gedeelte(n) uit deze uitgave in bloemlezingen, readers en andere compilatiewerken (artikel 16 Auteurswet 1912) dient men zich tot de uitgever te wenden.

CONTENTS

INTRODUCTION	1
Practical application of knowledge based systems to law: the crucial role of maintenance <i>T.J.M. Bench-Capon, F. Coenen</i>	5
A prototype for the retrieval of case law <i>R.J.A. Berg, P.M.G. Weghorst</i>	18
Towards a workbench for the legal practitioner <i>J.A. Breuker</i>	25
Development of a knowledge based system for the comparison of national social security systems of member states of the European Community in dynamic perspective <i>K. Debrock, V. Lemmens, F. Robben, B. van Buggenhout</i>	36
An intelligent interface to data bases on environmental law <i>P. Guidotti, L. Lucchesi, P. Mariani, M. Ragona, D. Tiscornia</i>	48
An alternative for deontic logic <i>J.C. Hage</i>	59
Involvement, phases and satellites Aims for research and development of legal knowledge based systems <i>A.W. Koers, D. Kracht, M. Smith, J.M. Smits, M.C.M. Weusten</i>	70
The JURICAS system in practice: decisions in a social security environment <i>C. van Noortwijk, P.A.W. Piepers, J.G.L. van der Wees</i>	79
Computer - aided legislative design: worth while the effort? <i>W.J.M. Voermans</i>	87
Combining analogical and deductive reasoning in legal knowledge base systems: IKBALS II <i>G. Vossos, J. Zeleznikow, T. Dillon</i>	97
The evolution of research aims <i>J.H. de Wildt, A.H.J. Schmidt, J.A. Quast, H.J. van den Herik, R. van Kralingen, P.R.S. Visser, W. de Vries</i>	106

INTRODUCTION

Kees van Noortwijk, Aernout Schmidt, Radboud Winkels

Legal knowledge based systems: Aims for research and development

This book contains the proceedings of the third international conference of the Dutch Foundation for Legal Knowledge Systems (JURIX). The conference was held in Leiden on December 17, 1990. Its theme was: Aims for research and development of legal knowledge based systems.

In the last decade, many universities, administrations and even a few software houses have shown a commitment to research and development of legal knowledge based systems (KBS). This gives rise to a number of questions. What are the reasons for this commitment, what are the aims in initiating research projects and designing specific legal KBS? Is it already possible to assess the accomplishments of existing systems and to indicate a direction for future research?

In this book, eleven different research groups present their vision with regard to some of these questions. They represent 5 different countries (The Netherlands, Belgium, The United Kingdom, Italy and Australia). In the next section, an overview will be given of the subjects that are covered by their contributions.

Overview of contributions in this book

The contributions to the conference can be divided into four categories. The oldest branch in the combined research efforts in computer science and law focuses on *information retrieval*. Traditionally, information retrieval applications concentrate on the use of thesauri and on the use of word by word indexation of entire texts. More recent research aims at the use of AI techniques for retrieval methods that are to a certain extent semantically driven. Two papers in this area of research have been contributed to the conference. Both contributions (by Berg *et al.* and by Guidotti *et al.*) investigate the use of such techniques for the retrieval of precedent cases.

The second category consists of research contributions on the problems with regard to design and development of *support systems for the legal practitioner*. This type of research was initiated in the second half of the 1970s by among others L. Thorne McCarty in his TAXMAN project. Six papers in this area of research have been contributed to the conference. Their contents range from the evaluation of actual advisory systems in practical use (Van Noortwijk *et al.*) to the description of more abstract prerequisites and knowledge representations for the development of a complicated and advanced "workbench" for the legal practitioner (Breuker). Koers *et al.* report on their combined theoretical assumptions and practical efforts, resulting in the interesting LASYS system. Vossos *et al.* deliver their views on the integration of two problems often encountered with regard to legal knowledge systems: the processing of case based reasoning and the processing of deductive reasoning. De Wildt *et al.* step back and analyze what has become of their and our initial research aims in the field of legal AI. They point to a shift in aims: from general research concerning complete systems to more specific research aims.

Bench-Capon *et al.* elaborate on such a specific research aim: they analyze the problems engendered by changes in the law and suggest a few methods for meeting this difficulty.

A recent development in research into legal KBS deals with *support systems for the legislator*. Voermans' contribution gives a theoretical analysis of the problems in this area. Debrock *et al.* suggest the use of decision tables in the design of support systems. Their research efforts are inspired by the growing importance of international legislation. The developments in the EC in this area generate many problems that may be solved by legal-AI applications.

One contribution to the conference lies within the field of formal legal philosophy. Hage gives an overview of the deficiencies that are shown by standard deontic logic. He consequently proposes an alternative approach.

The foundation for Legal Knowledge Systems

The Foundation for Legal Knowledge Systems provides a forum for scientists in the Netherlands in the field of law and artificial intelligence. Each year, the Foundation organizes three to four meetings on research topics that are of interest to its members. At these meetings, ongoing research projects are discussed and relevant software is demonstrated and examined. Furthermore, the Foundation organizes an annual international conference. The first conference (Amsterdam, 1988) was dedicated to paradigms in computers and law. The proceedings of that conference contain descriptions of the paradigms of ten university research groups in the Netherlands (*RI-paradigmata*, Lelystad: Royal Vermande publishers B.V., 1988). The second conference (Utrecht, 1989) addressed the search for criteria for the validation and the practical use of legal KBS (*Legal knowledge based systems, an overview of criteria for validation and practical use*, Lelystad: Royal Vermande publishers B.V., 1989).

This third conference, held in Leiden on December 17, 1990, has as its theme: *Aims for research and development of legal knowledge based systems*.

For more information on the Foundation for Legal Knowledge Systems, please contact the secretariat:

JURIX, Foundation for Legal Knowledge Systems
attn. Mr. C.N.J. de Vey Mestdagh
University of Groningen
Computer/Law Section
Oude Kijk in 't Jatstraat 26
9712 EK Groningen
The Netherlands
Tel.: +31 50 635790, Fax: +31 50 635603
Email: SESAM@HGRRUG5.BITNET

Acknowledgements

We would like to thank Ineke Rijkschroeff-Kwa for editing the text of this book and taking care of the layout. It all had to be ready in a very short period of time, and without her this would not have succeeded.

Special thanks goes to the university of Leiden, for supporting this conference in many ways, and to the conference organization committee which consisted of Jeanette Quast, Erik Kroon and Astrid Smorenberg.

PRACTICAL APPLICATION OF KBS TO LAW: THE CRUCIAL ROLE OF MAINTENANCE

Trevor J.M. Bench-Capon and Frans Coenen
Department of Computer Science
University of Liverpool

Summary

Legal Knowledge Based Systems are, by definition, grounded on law. Because law is subject to amendment, significant problems of adaptation are posed for a legal KBS in practical use. If the use of such systems is to become widespread and routine, these maintenance issues must be taken seriously.

This paper describes a research project which is developing tools to assist in the maintenance of legal knowledge bases.

1. Introduction

Research into the application of Knowledge Based Systems to Law has established two main things. First that the potential demand for, and utility of, such systems is great, and second that such systems are feasible. The demand is not confined to lawyers: indeed the wider impact is likely to be on those whose jobs are governed by law (or law-like regulations). For evidence as to the potential we may consider the following quotation written by Paul Duffin [Duffin 88], a prominent member of the UK Central Communications and Telecommunications Agency (CCTA).

The UK Civil Service is the largest single user of conventional IT equipment and services in the UK ... The CCTA has a specific responsibility to research and then encourage the use of appropriate IT to assist in the administrative mechanisms of Government. KBS represents one such technology which CCTA has identified as being of particular benefit ... In terms of government administration, KBS may be the single most significant development to emerge since the computer itself, for it offers a means of streamlining and improving decisionmaking to an unprecedented degree.

The activities that he saw being particularly influenced by such systems go to the heart of administration

Much of government 'mainline business' involves the administration of regulations or the following of set procedures or, frequently, both. These areas of application are amenable to computerized assistance using ES [Expert Systems] techniques, as has been demonstrated [Duffin 88].

If we widen our notion of a "legal" KBS to include not only the "laws of the land" but also the internal procedures and guidelines used by companies to direct the activities of their employees, this remark about the Civil Service of the UK becomes equally relevant to any large organization which performs a good deal of administration - that is, any large organization whatsoever. All administrators must make decisions within the policies and guidelines of their employers, and this activity

is an informal analogue of legal decision making, and susceptible to the application of similar KBS techniques. As an example of this law-like activity, banks have policies on lending, and issue guidelines to their staff to realize these policies. A system which supported a loan scheme would be able to employ much the same techniques as a truly legal system, such as a system to support the adjudication of claims to welfare benefits.

Thus legal KBS are wanted as practical systems, and it is this very practicality that has attracted many researchers to the area. But are they feasible? Much research has been devoted to showing that they are: the British Nationality Act project [Sergot 86] has shown how legislation can be represented in an executable form, and further related work [Bench-Capon 88], has explored the relation of such a formalism to a practically useful system. And indeed there are practical examples of such systems in use, perhaps most notably the Retirement Pension Forecast and Advice System (RPFA) [Spirgel-Sinclair 88] and the VATIA system [Susskind 88].

Thus it can be said both that there is a great potential demand for such systems, and that it has been shown that it is possible, and profitable to build such systems. The question therefore arises as to why such techniques are not part of the routine armoury of large organizations.

Part of the answer lies in issues of knowledge engineering. The traditional consultative model of an expert system is simply not appropriate to support many of the tasks which need to be addressed. The RPFA, mentioned above, does not follow this model, and the need to take the task seriously and to tailor the support provided to the particular task is well documented in [Bench-Capon 90] in which the same legislation is shown to give rise to very different systems when these systems are directed to different tasks founded on that legislation.

These knowledge engineering issues, however, simply mean that it is that much more time consuming and difficult to build such a system, not that it is impossible. It is our belief that the greatest barrier to the routine use of KBS techniques for practical legal applications lies not so much in the problems of building the systems, since this process is becoming better understood, but in the problems associated with the maintenance of such systems. For no one is going to invest the amount of effort involved in building a legal KBS unless he can have some assurance that the system will have a reasonable length of useful life. And since one certain thing is that the law will change over time, this means that there must be a clear strategy to enable the system to cope with these changes.

2. Nature of Changes in a Legal KBS

The way in which the knowledge relevant to a legal Knowledge Based System KBS changes is different from many of the areas to which KBS techniques are applied. In the area of medical diagnosis, for example, where the subject of the domain is the human body, the ills that it may suffer, and the treatments that may cure them, the knowledge that a system must use is relatively stable. Understanding may increase, and new treatments may be developed, but this extends the knowledge that needs to be included without invalidating what already exists. In the legal domain, however, this monotonicity of knowledge is absent. For laws are repealed and amended as well as added to, and a decision in a landmark case may necessitate revision of existing interpretations of the law. And this is a significant difference, posing

significant problems. The situation is analogous to the well known problem of truth maintenance in KBS: so long as information is simply increased there is no problem, but when an additional piece of information requires existing beliefs to be revised the matter is no longer simple, as the variety of truth maintenance systems and non-monotonic logics found in the AI literature demonstrates. Thus while the incremental refinement of the knowledge base typically found in classic expert systems such as MYCIN and XCON, whereby rules are simply added to the knowledge base may be a feasible strategy for those domains, such a strategy is inappropriate to a legal KBS.

Problems arising out of the changes in the law are well known in conventional data processing: changes in tax law, for example, must be announced well in advance of coming into effect so as to allow time for the considerable task of altering the programs which have to apply these laws in payroll and other applications. But the problems with a legal KBS are greater than with a conventional system. In the conventional system the limited range of tasks which such a system can perform tends to restrict the knowledge represented. Thus a payroll system will need to have recorded within it such things as the rates at which tax is paid and the thresholds at which these rates come into effect, but it will not record the sort of expertise and knowledge of precedent that we would expect from a tax lawyer. The kinds of thing which are recorded change at regular intervals, are signalled well in advance, and change in relatively predictable ways. The legal KBS, in contrast, will be expected to incorporate some elements of the lawyer's expertise as well, and this will change in an irregular and unpredictable manner as decisions are made, or as external circumstances change. This means both that detecting that such changes have occurred is a problem, and that deciding on the appropriate response to such changes, and incorporating them into the knowledge base may be difficult.

3. Maintenance Assistance for Knowledge Engineers (MAKE)

The MAKE project, a collaboration between the University of Liverpool, ICL and British Coal is investigating the issues connected with maintenance of regulation based KBS. The specimen application being considered concerns claims for compensation for work related injuries made by employees of British Coal. Such an application is fairly representative of the sorts of application claimed in the introduction to offer the greatest potential for the exploitation of legal KBS.

As expected the need for maintenance is considerable. Regarding the law itself, each year, there are between 10 and 20 court judgements in British Coal cases and another 5 relating to other employers, but with significance for British Coal. There are up to 20 new relevant Statutory Instruments, and 10 technical instructions issued. In addition the policy of British Coal is modified from time to time, and some 10-15 such policy decisions are made in a typical year. All of these alterations need to be assimilated by the clerks dealing with the claims. Other changes in the expertise of these employees arise out of changes in medical views, for example the acceptance that a particular substance can cause dermatitis; policy changes by other bodies, as when a particular firm of solicitors may start to issue writs if the claim is not settled in a certain period of time; and changes in the perception of methods of work or occupations. British Coal estimate that these will require another 30 changes per year. Some of these changes will be relatively minor, but none the less the cumulative effect of these changes indicates the rapidity with which a knowledge base dealing with this sort of application would go out of date. If the advantages

cited for using KBS to support such tasks are to be realized, it is essential that the knowledge be kept up to date, and so a practical system would require continuous updating.

4. Influence of the Representation on Maintenance

One factor that makes the maintenance of a legal KBS difficult is that when the Knowledge Engineer encodes the knowledge that he has elicited from the expert he will often bring together separately presented items in a single rule. Some of the effects of this practice are shown in the following example of Category C Retirement Pension.

The UK Social Security Act states:

39(1) Subject to the provisions of this Act-

(a) a person who was over pensionable age on 5th July 1948 and satisfies such other conditions as may be prescribed shall be entitled to a Category C retirement pension at the appropriate weekly rate.

To interpret this we need also to bear in mind

27(1) In this Act "pensionable age" means -

(a) in the case of a man, the age of 65 years; and

(b) in the case of a woman, the age of 60 years.

Now if we consider the kind of knowledge that an expert adjudicator might apply to decide claims for this benefit, we might see him allowing claims of men aged 101 and women aged over 96. This would certainly pick out the correct group of people and would be the most convenient expression of the knowledge if the claim form gave the age of claimant. It does, however, "compile in" both a certain amount of arithmetical expertise and knowledge of the current date, as well as the interaction between 27(1) and 39(1). If the claim form contained not the age of the claimant but the date of birth, however, this would not be the most convenient expression of the knowledge, since a calculation would now be required to get the age from the date of birth, and the expert would be likely instead to operationalize the knowledge as "men born before 5/7/1888 and women born before 5/7/1893". This still conflates 27(1) and 39(1). The point here is that when experts operationalize their knowledge they will amalgamate knowledge from a variety of sources in the way which is of most use to them, but which may obscure some necessary (from a Software Engineering standpoint) structural information.

The conflation of different items of knowledge will require that these be separated out before the knowledge base can be changed. Suppose for example 27(1) was amended, perhaps to equalize pensionable ages. The impact of this on the interpretation of 39(1) could not be recognized in a conflated representation. This suggests that the representation used for a legal KBS should avoid conflating disparate items of knowledge into single structures. This in turn requires that we be able to mirror the structure of the knowledge sources in our representation. A full discussion of this need is to be found in Routen and Bench-Capon [Routen]. This

can often be achieved by a disciplined use of a representation rather than use of a distinctive representation, although certain extensions are required to Prolog (or any first order formalism) if this is to be possible with regard to legislation: again this is fully discussed in Routen and Bench-Capon. Further, we can note that achieving a structural correspondence here will also enable us to record the provenance of all the items of knowledge in out intermediate representation, which is not a simple matter in the absence of such isomorphism, but which is vital if changes are to be followed through from source to knowledge base.

Thus one thing that can be done to ease the problems of maintenance is to use a representation that enables the knowledge base to maintain a close structural correspondence with the original source documents. Moreover, for this to have its best effect, statements in the representation must be truly declarative. While almost all knowledge representation paradigms have declarativeness as an aspiration, in practice the use of, for example, conflict resolution strategies in production rule systems, means that it is not possible to detach a piece of a knowledge base from its context and consider its correctness in isolation. If we want to ensure that localized changes to the source material result in correspondingly localized changes to the knowledge base, we must be sure that there are no ramifications of changes resulting from a subtle alteration of the meaning of the statement deriving from its context in the knowledge base.

We therefore conclude that the form of representation used is an important factor in the production of maintainable systems. The tools developed on the MAKE project are consequently targetted upon a form of representation which exhibits the properties seen as desirable above. This formalism is the representation and inference Toolkit developed on the Alvey-DHSS Demonstrator project, particularly for the representation of legislation [Bench-Capon 90]. In brief, these facilities comprise an inheritance hierarchy, with the classes viewed as logical types, their slots as attributes of these types, and the possible values of these slots specified in the class description. Inheritance was by strict specialization. This hierarchy represents a vocabulary in which constraints expressing the relations between slots can be expressed. These constraints are expressed in a typed logic extended to include arithmetic.

5. Proposed Tools

It is not the intention of the MAKE project to address major maintenance tasks which may necessitate the entire rebuilding of the system. The aim is to address minor maintenance only, i.e. maintenance resulting from the day to day changes in the source material due to changes in legal texts, the application and operation of the law etc. The maintenance tools required can be considered under five headings a) the logical structure of the KB, b) the logical structure of the class hierarchy, c) the source material d) validation and e) maintenance support. Each is discussed in the following Sub-Sections.

5.1. Logical Structure of The KB

As a result of any necessary maintenance the logical structure of the KB must remain (logically) correct. Maintenance on the KB will involve one or more of the following:

- M1 The introduction of a new rule.
- M2 The removal of an existing rule.
- M3 The Modification of an existing rule by adding, removing or adjusting a condition or conditions.

Predicates found in rules are either "leaves", in which case their truth will be ascertained by direct reference to the class hierarchy or the user, "roots", where they are what the system is intended to establish, or "intermediate", where they need to be established from the KB in order to establish some root. The terminology here alludes to a tree representation of a rule base, as illustrated by Figure 1.

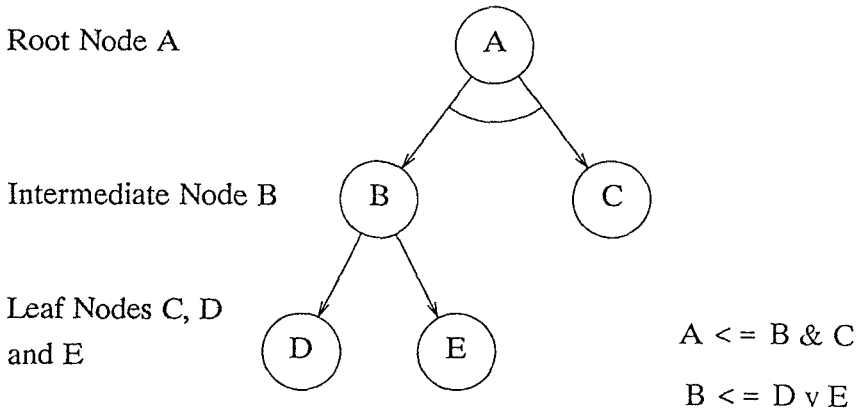


FIGURE 1: AND-OR Tree Representation of a Rule Base

Thus the effect of introducing a new rule may result unwanted redundancy or the creation of a missing branch, a hard contradiction or soft inconsistency. The removal of a rule may also result in the creation of a missing branch or cause a section of the KB to become redundant. The modification of a rule has the same effect as removing a rule and introducing another.

The following tools are therefore proposed:

- T1 KB Folding Tool.
- T2 KB Unfolding Tool.
- T3 Hard Contradiction ID Tool.
- T4 Soft Inconsistency ID Tool.

T1 and T2 are intended to identify redundant and missing branches, and should be implemented on the proposed rule base in its intermediate representation. This is because the existence of redundant and missing branches will only be significant in a task dependent rule base. However, T4 and T5 will have to be implemented at the constraint level.

Before going on to describe these four tools in greater detail it should be noted that a number of KBS development environments currently include some form of inconsistency checking. For example the COVADIS system includes an inconsistency checking tool based on logical inference coupled with expert interaction [Rousset 88].

5.1.1. T1 KB Folding Tool

For a rule not to be redundant it must eventually fold into the root node. In other words the head of each rule in the revised KB must be contained in the tail of at least one other rule in the revised KB, unless it is the root node itself. Thus if we consider a simple rule base of the form given in figure 1:

A if B & C (1.1)
B if D $\vee$ E (1.2)

Where A is the root node representing the goal to be established. If we add the rule:

C if F $\vee$ G (1.3)

We can show that no redundancy exists because the head of Rule 1.3 is contained in the tail of Rule 1.1 and the head of Rule 1.1 is the root. It should be noted however that this tool simply proves that the KB does not contain any redundancy, i.e. that a path exists from each rule in the rule base to the root node. It gives no indication of the correctness of this path.

5.1.2. T2 KB Unfolding Tool

A missing branch exists in a KB if a rule does not unfold into a leaf node the truth of which can be ascertained by direct reference to the class hierarchy or the user. In other words each condition contained in the tail of a rule must either (a) represent the head of another rule, (b) be ascertainable from the class hierarchy, (c) be expected to be supplied by the user or (d) be the root node. If we return to the above example B and C, which are contained in the tail of Rule 1.1, are represented in the heads of rules 2.2. and 2.3. However, D, E, F and G are not present in the heads of any rules, they therefore represent leaf nodes and hence if they are not existent in the class hierarchy a missing branch will exist. As with the KB Folding Tool the Unfolding Tool simply shows the existence of a path from each rule to one or more leaf rules, it does not verify the correctness of this path.

5.1.3. T3 Hard Contradiction ID Tool

A hard contradiction in logical terms is represented by any logical expression which evaluates to false for all possible values of its constituent parts. Thus in its simplest form a contradiction may be represented by a constraint of the form:

A & not A

More simply put a hard contradiction exists if, as the result of a set of legal assertions, an attribute can be inferred to be both true (T) and (F). By representing the rules in a KB in Conjunctive Normal Form (CNF) we can attempt to logically show that a contradiction exists using well established logical proofs. Using these

standard proofs we can assert each attribute in the revised rule base, attribute by attribute and rule by rule and determine, as the rule set is rebuilt, whether any contradiction occurs. This process continues until all attributes have been asserted. This can best be illustrated by considering an example. Suppose we have a knowledge base consisting of the following CNF clauses:

$$P \vee \text{not } S \quad (2.1)$$

$$Q \vee \text{not } P \quad (2.2)$$

$$R \vee \text{not } P \quad (2.3)$$

and we wish to add the CNF clause:

$$\text{not } Q \vee \text{not } R \quad (2.4)$$

Where P, Q, R and S are boolean attributes. We can test for hard contradiction by determining the effect of asserting first not Q, then not R, and then using the union of the inferred values to give the consequences of the disjunction not Q $\vee$ not R as a whole. The result can be drawn up in the form of a Logical Consequence Table as shown in Table 1. In this case no conflict occurs (although not P and not S are both consequences of Rules 2.1-2.4), and so we can say that no hard contradiction exists.

	(I)	Rule 2.4		
		not Q	not R	(I-4)
P	T/F	F (2)	F (3)	F
Q	T/F	F	T/F	T/F
R	T/F	T/F	F	T/F
S	T/F	F (1)	F (1)	F

Table 1

Suppose we now add the CNF clause:

$$S \vee P \quad (2.5)$$

Then asserting each attribute for this rule in sequence we get a Logical consequence Table as shown in Table 2. When we assert S, P is proven to be True from 2.1. Q is then proven to be True from 2.2 and R from 2.3. However, when we consider 2.4, if R is True Q must be False and if Q is True R must be False. Thus by asserting S a contradiction results. However, this does not invalidate 2.5, which states "S $\vee$ P", since if P can be True then the rule may still be correct. But on asserting P, Q and R are again shown to be True from 2.2 and 2.3 respectively, and again according to 2.4 if R is True then Q must be False and if Q is True then R must be False. Therefore a contradiction also results when P is asserted. However, 2.5 states that one of them must be True. A hard contradiction can therefore be said to exist.

	(I)	Rule 5		
		S	P	(I-5)
P	T/F	F (1)	T	F
Q	T/F	X	X	X
R	T/F	X	X	X
S	T/F	T	T/F	T/F

Table 2

5.1.4. T4 Soft Inconsistency ID Tool

What we term a "soft inconsistency" occurs when some proposition is a consequence of the KB where as it is in fact known that its negation is possible. In the simplest possible case we may have two rules:

$$P \Rightarrow Q$$

$$P \Rightarrow \text{not } Q$$

There is no logical contradiction here, but not P is a logical consequence of the KB. If, however, P represented something which we knew to be sometimes true and sometimes false, this would indicate that our KB was in error. In other words in the set of rules where P is a theorem a soft inconsistency exists if we know that not P is possible. More generally a soft inconsistency exists if a rule base contains a logical expression which is a theorem when we know that this should not be the case.

Soft inconsistencies cannot be identified by logical deduction alone, since this will not tell us whether the negation of a logical consequence of our axioms is something that must be possible. However, we can determine the logical consequences of rules as illustrated in the previous Section using Logical Consequence Tables. We can also place annotations on the attribute slots in our class hierarchy indicating which values must be possible for that attribute, and automatically compare the desired consequence with the actual consequence as indicated by the Table. Unfavourable results can then be output to the maintenance engineer. If we consider a set of attributes which can only be set to True or False, three types of annotation may be appropriate:

1. Attributes which must be able to be either true or false. e.g.
(ManagerIsMale literal (true, false)).
2. Attributes which can (or should) be always true e.g.
(allCoalMinesHaveManagers literal (true)).
3. Attributes which can (or should) be always false e.g.
(allManagersManageCoalMines literal (false)).

Thus it would be undesirable if it were a consequence of our regulations that all managers were male, but perfectly acceptable if we organized things so that all mines always had a responsible manager.

If we take Rules 2.1 to 2.4 from the above example this resulted in a Logical consequence Table of the form shown in Table 2. Attributes P and S are proven to be False and attributes Q and R True or False. If we consider P and S first, a logical consequence of False will be acceptable only if both attributes were allocated a Type 3 Annotation. Similarly a logical consequence of True or False would be acceptable for attributes Q and R if allocated a Type 1 Annotation.

It should be noted that the Soft Contradiction Tool will only operate successfully if all hard contradictions have first been removed.

5.2. Logical Structure of The Class Hierarchy

The maintenance associated with the class hierarchy, which in our representation defines the vocabulary of the domain, may involve:

- M4 The modification of an existing slot by introducing a new value.
- M5 The modification of an existing slot by removing an existing value.
- M6 The modification of an existing class by introducing a new slot.
- M7 The modification of an existing class by removing an existing slot.
- M8 The introduction of an entire new class.
- M9 The removal of an existing class.

In practice a value will be added to a slot as a consequence of the introduction or modification of a rule. The allocation of this additional value to an existing slot will not generally effect the operation of any established rules or the existing class hierarchy. The exception to this is if rules exist that use the possible values of a slot to express negation. For example, suppose the class "paint" has a slot "colour" with possible values, red, blue and other. We may then have a rule:

(paint notPrimaryColour true if not((colour red) v (colour blue)))

This would compile to:

(paint notPrimaryColour true if (colour other))

Now adding an extra value to the colour slot, say yellow, would jeopardize this rule.

Thus when introducing a value it is necessary to identify the uncompiled rules in the intermediate representation of the KB that use this slot and hence the compiled clauses in CNF which are "jeopardized", and then to recompile these clauses so that account can be taken of the extended range of values for the attribute.

Alternatively removing a value from a slot jeopardises all rules which make use of that value, and therefore these must also be deleted from the KB.

A tool to identify jeopardized rules as a result of removing and introducing values to and from slots in the class hierarchy is therefore proposed:

T4a Rules Jeopardized by Slot Changes ID Tool.

The identification of jeopardized rules can be simply achieved by searching through the rule base and identifying all rules where the head or tail of a rule contains the changed attribute. If we have a rule base of the form:

$$\begin{aligned} Aa &\leq -Bb \vee Ca \quad (1) \\ -Bb &\leq Da \& Eb \quad (2) \\ Ca &\leq Fa \vee Ga \quad (3) \end{aligned}$$

Where A to G are attributes in a class hierarchy which can have values a,b or c and Aa indicates that a has the value a etc. We may extend the range of possible values for (say) attribute B to include d in which case rules 1 and 2 will be jeopardized. Rule 1 because B is contained in the tail (and we need to consider whether we want Bd to imply Aa) and Rule 2 because B is contained in the head and so we need to consider whether we want Da and Eb to imply Ba $\vee$ Bc $\vee$ Bd or only Ba $\vee$ Bc. Simply recompiling Rule 1 will give the effective rule

$$Aa \leq Ba \vee Bc \vee Bd \vee Ca$$

but since it was originally written in ignorance of the possibility Bd, it may be that

$$Aa \leq Ba \vee Bc \vee Ca$$

is what is required. Similarly recompiling Rule 2 will give

$$Ba \vee Bc \vee Bd \leq Da \text{ and } Eb$$

whereas we may want only

$$Ba \vee Bc \leq Da \text{ and } Eb$$

Notice also that the answers may differ in the two cases: the recompiled version may be right for some rules and wrong for others.

The addition of a new slot to an existing class or the introduction of an entire new class will not affect the operation of the existing KB.

The removal of a slot or an entire class is effectively the same as removing a sequence of values and can be addressed in the same manner.

5.3. *Changes in Source Documents*

In the above Section we proposed a tool to identify rules jeopardized by changes to the class hierarchy representing the vocabulary of our domain. At a higher level, rules will also be jeopardized by changes in the source documents, as when legislation is amended. By including links in the intermediate representation from the sources to the various elements of the class hierarchy and KB, we can exploit these links to show which elements will be jeopardized by changes in the source material.

The knowledge analysis tools used to build the specimen application, KANT [Storrs 89], includes linking facility of this type. This allows the knowledge engineer to trace changes in the source material through to the KB and class hierarchy and thus identify the attributes and rules that will require attention. At present this is

implemented manually. It should be possible, however, to automate this process so that the rules and attributes in the class hierarchy jeopardized by changes in the source material can be listed automatically, and integrated with the tools already described. Thus:-

T4b Rules Jeopardized by Source Changes Identification Tool

5.4. *Validation*

So far only the verification of the KB and class hierarchy have been considered. However, it is also necessary to validate the KBS after maintenance has taken place. Currently the most usual way of validating KBSs is to run test cases and to compare the results produced by the system to those produced in "real life". If a conflict arises some corrective maintenance will be required. The identification of the nature of this maintenance can only be carried out by manual intervention. However, a monitoring tool to monitor batches of test cases may be appropriate to look for comparisons and to identify common factors amongst test cases that do not produce a satisfactory result. Thus:

T5 Test Case Monitoring Tool.

5.5. *Maintenance Support*

When adding new rules it is desirable where ever possible to utilize existing rules and slots rather than create new ones. A useful tool to assist in the identification of appropriate existing rules and slots would be a data dictionary where by existing rules and slots can be identified by entering keywords. Data Dictionaries have been used as an important software development support tool in conventional systems for some time. For example as a quasi-formal method of describing the content of information items in Data Flow Diagrams [Pressman 88]. Two data dictionaries, one for the KB and one for the class hierarchy, thus suggest themselves:

T6. KB Data Dictionary.

T7. Class Hierarchy Data Dictionary.

6. **Conclusion**

In this paper we have identified the maintainability of legal KBS as an important factor in their successful exploitation. Unfortunately this issue has attracted too little attention to date. In the MAKE project we are producing a coherent strategy for the maintenance of such systems, embracing a methodology for knowledge analysis, recommendations for representation principles, and a set of tools to support the amendment of the knowledge base. Some useful tools for the maintenance of a KB have been sketched in this paper.

7. **Acknowledgement**

The work described above was carried out as part of the MAKE Project, supported by the Information Engineering Directorate of the UK Department of Trade and Industry and the UK Science and Engineering Research Council. The project collaborators are ICL, the University of Liverpool and British Coal. The views

expressed in this paper are those of the authors and may not necessarily be shared by the other collaborators.

8. References

- [Bench-Capon 90] T.J.M. Bench-Capon and J.M. Forder, Knowledge Representation for Legal Applications in T.J.M. Bench-Capon (ed.), *Knowledge Based Systems for Legal Applications*, Academic Press, forthcoming 1990.
- [Bench-Capon 88] T.J.M. Bench-Capon, Applying Legal Expert Systems Techniques: Practical Considerations. In *KBS in Government 88* (ed. Duffin), On Line Publications, 1988, pp. 205-214. (I).
- [Duffin 88] P.H. Duffin (ed.), *Knowledge Based Systems: Applications in Administrative Government*, Ellis Horwood, Chichester, 1988. p. 7.
- [Pressman 88] R.S. Pressman, *Software Engineering, A Practitioners Approach*, 2nd Edition. McGraw-Hill 1988.
- [Rousset 88] M. Rousset, *On the Consistency of Knowledge Bases: The COVADIS System*. Proceedings of ECAI 88.
- [Routen] T.W. Routen and T.J.M. Bench-Capon, Hierarchical Formalizations. Forthcoming in *International Journal of Man-Machine Studies*.
- [Sergot 86] M.J. Sergot, F. Sadri, R.A. Kowalski, F. Kriwaczek, P. Hammond, H.T. Cory, The British Nationality Act as a Logic Program. *Communications of the ACM* 29, 5 (May 1986), pp 370-386.
- [Spirgel-Sinclair 88] S. Spirgel-Sinclair and G. Trevena, The Retirement Pension Forecast and Advice System, in [Duffin 88], op cit. pp 34 - 40.
- [Storrs 89] G.E. Storrs, C.P. Burton. KANT, A Knowledge Analysis Tool. *ICL Technical Journal*, Vol 6, No 3, May 1989.
- [Susskind 88] R. Susskind and C. Tindall, *Ernst and Whinney's VAT Expert System*, Proceedings of the Fourth International Expert Systems Conference 1988.

A PROTOTYPE FOR THE RETRIEVAL OF CASE LAW

Radboud J.A. Berg, Paul M.G. Weghorst
Utopics b.v. Informatie-architecten
Deldenerstraat 36
7551 AG Hengelo
The Netherlands

Summary

In this article a prototype is described for the retrieval of case law that is stored in a database. For this purpose a concept was developed that combines artificial intelligence techniques with database technology. The concept is based on searching for a match between case models, which are based on characteristics stated in a dynamic classification model. With this article we aim at describing useful techniques for making case law accessible with maximum flexibility and minimal maintenance overhead.

1. Introduction

In the field of juridical knowledge based systems often the distinction is made between law-based systems and case law-based systems. The approach discussed in this paper can be classified as case law-based. In analogy to the CAMO research [Visser 90] our approach focuses on the modelling and retrieval of case law in contrast to reasoning on basis of case law (case-based reasoning), as described in for instance [Hage 90] or [Wildt 89]. We have built a prototype of a case law based knowledge system. In this paper we focus on the functionality of such a system and the concept behind the approach we used to model and implement the system. The second paragraph deals with the functionality of the prototype and the classification model used to classify the case law. The third paragraph explores the commitment to artificial intelligence (AI) and explains why we have chosen to integrate AI with 'conventional' database technology. In the fourth paragraph the concept behind the used approach is explained. Paragraph five is a further elaboration of the AI aspects. Emphasis will be put on the capability to deal with a changing classification model. To counter this problem our approach uses a dynamic classification model. Furthermore we will discuss the reduction of the search space and the possibility of specifying mismatch rates. The last paragraph finally presents some conclusions.

2. Functionality

The purpose of our research was to build a prototype to identify relevant case law for a case at hand. There were a few constraints to be satisfied:

- *interactivity:*
experience shows that users have difficulties specifying exactly what case law is considered relevant in a particular case. Therefore the system should provide for the possibility to narrow down or adjust the specifications of the relevant case law. After each retrieval of relevant case law, the results are presented. The user can decide whether the results are satisfying or he can decide to alter the case specifications and let the system retrieve the case law corresponding to this new specifications. This way the user reaches the desired results in an interactive and iterative way;

- *performance*:
the retrieval of the relevant case law should be quick in order to make an interactive and iterative process possible and to make working with the system agreeable;
- *completeness*:
in order to withhold the user from tedious manual search in voluminous archives it must be guaranteed that the results will be complete and correct. The user must be convinced that all case law corresponding with the given case law specifications will be retrieved. Without this conviction the user will not use the system alone (or not at all) but sideways with old search methods. In this case we would not have reached a workload reduction but a workload increase;
- *dynamism*:
the system must have the possibility to adapt to a changing (view on) case law. Otherwise the system will be outdated after a short period of use. In order to achieve this a dynamic classification model is used.

In the remainder of this paragraph we will elaborate how these constraints were met.

The first problem we encountered, was how to model the case law. For this purpose we used a dynamic classification model. The classification model consists of several characteristics which are used to describe the case law. The classification model is a dynamic one because it is adaptable to a change of case law itself or to a change in the view on case law. In the paragraph on AI aspects we will further elaborate this dynamic aspect.

Let us look at the characteristics used to describe the case law. We initially distinguished the following characteristics:

- court (administer of justice);
- statute/legislation (related to the conflict);
- parties (involved in the conflict);
- type of conflict;
- catchwords (hierarchically structured);
- dictum;
- motivations of the court.

Suppose we wish to determine all relevant case law for a newly risen conflict. A search profile should be constructed that characterizes the conflict. This search profile can for example characterize the conflict as an appeal before the Crown of a *trade organization* against a decision of *the Secretary of State* concerning legislation on *chemical waste*. The conflict can further be characterized with a number of catchwords; e.g. the kind of chemical waste. By matching this search profile with the case law in the database it is possible to determine all case law which satisfies the search profile. A more precise description of the matching process will be covered in the paragraph on AI aspects. On the basis of the retrieved case law one can decide to adjust the search profile. Now the matching process can be repeated. In this way it is possible to retrieve, in an interactive process, the desired case law.

A special feature of the prototype is the possibility to select case law which differs from the search profile on one or a few characteristics. Detailed information is given by the system to explain to the user why the selected case law did not match the search profile

exactly. Although the thus selected case law does not correspond exactly with the case at hand, it can be very useful, because of the similarity between the cases.

3. Artificial intelligence and databases

When dealing with this subject, many implementation environments are useful. Keywords for a system to be used in the field are:

- *flexibility:*
since the view on case law may differ between organisations and may change in time, it is important that the system and the models can be modified easily, the latter even by a privileged user;
- *integration:*
data and knowledge are often already stored in information systems. In order to be able to use this data, the system must be provided with means to access those data sources with powerful interfaces;
- *creativity:*
the user should not be limited in modelling cases and interrogating the system. Powerful on-line facilities for interrogating and modelling must therefore be at hand;
- *effective and efficient in dealing with large data/knowledge bases.*

In our opinion these constraints reduce the possibilities to two: expert system shells and fourth generation database management systems. Other possibilities might be third generation languages, but with these it is very difficult to build systems which satisfy the flexibility demand, or languages like Prolog and Lisp, but systems built with these language environments conflict with the flexibility and effectiveness/efficiency claims.

To start with the first mentioned: expert system shells. Major advantages in developing knowledge based systems with these products are:

- the support of several reasoning strategies;
- the presence of knowledge modelling structures and explanation features.

Less easy to find in the current range of expert system shells are:

- efficient handling of large amounts of data;
- unlimited modularity;
- query optimization;
- easy portability on the top range microcomputers.

Fourth generation database management systems offer the following characteristics:

- flexible user interfaces;
- portability to a wide range of computers;
- modularity;
- query optimization;
- efficient handling of large amount of data.

They are, however, less tailor-made in the craft of:

- structuring knowledge;
- reasoning (including explanation).

In summary: both have complementary (dis)advantages. The disadvantages of the expert system shells are hard to get round. In dealing with large databases, the reasoning time of expert system shells will be considerable. Therefore we decided to develop a tailor-made inference engine within SQL, to be used within the fourth generation relational database management system Oracle.

4. Concept

To make case law accessible, the choice was made in favour of a concept already used in another domain, the domain of subvention programs see [Herwijnen 89] and [Houten 89]. Central issues are a classification model and a case model, cf. [Visser 90]. Both are expressed by means of information items. The classification model consists of all applicable information items and their allowed domain values (dependent on the information item type, these can be alphanumeric, boolean or enumerated).

The case model consists of a set of case items. A case item is a combination of an information item and a specific domain value. For a case model it is of course allowed to use the same information item several times. The information model is given in figure 1.

Information item values may have a hierarchical relation. A hierarchical relationship between two domain values means that one domain value spans at least the other domain value, e.g. when we look at the information item *Court*, *Civil Court* is a predecessor of *Cantonal Court*. When an information item value was entered for a specific case, the system knows that all descendants are valid too (of course the consistency will be maintained if the hierarchical structure is changed). For example: when the clause "Valid for *Civil Court*" is true, "Valid for *Cantonal Court*" will be true too. In figure 1 this structure is depicted by the arrow indicated by 'hierarchical structure'.

With the building blocks given by the information items of the classification model, it is possible to construct logical expressions that make up a description of a case. An example is given below:

(Court = Raad van State) and

(Statute = Wet Administratieve Rechtspraak Overheidsbeschikkingen) and

(Article = 5) and (Article = 7) and

(Catchword = Belang)

The system is used as follows. One classifies the case for which case law is to be found. The case description is called the search profile. The search profile is stated in term of the classification model. For instance: case law should be found for which the court is the 'Raad van State' and the finding should be based on the statute 'Wet op de Raad van State' or the 'Wet Administratieve Rechtspraak Overheidsbeschikkingen'. In addition the finding should be dated after the beginning of the year 1989. Stated in terms of the classification model the search profile reads like this:

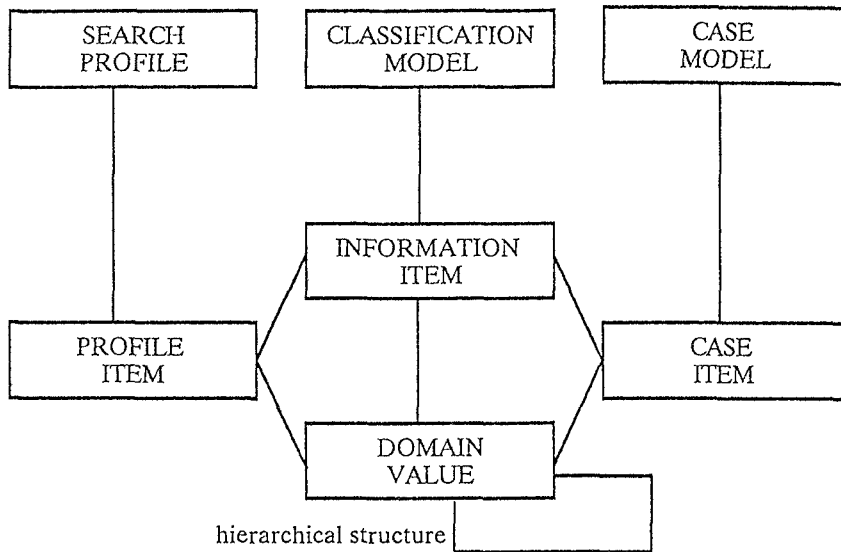


Figure 1 Information model

(Court = Raad van State) and

(Statute = Wet op de Raad v.St. or Statute = Wet Adm.Rechtspr.Overheidsbesch.) and

(Date $\geq$ 890101)

This example shows also the use of operators such as ' $\geq$ ' and the logical AND and OR.

When the search profile is entered, the inference engine tries to find relevant case law with due of the following strategy. Case law is relevant if the case model in question of the case law in the database contains no conflicting information items and corresponding values with respect to the given search profile. Information items that were not used in the search profile but were used in the case law description are considered 'don't care' in the search profile and vice versa.

5. AI aspects

Although the prototype was implemented in the fourth generation relational database management system Oracle, concepts from the field of Artificial Intelligence are embodied in the prototype. First, the reasoning component was realized within SQL (and is therefore part of the software). The classification model and the case models of the case law, however, are stored in the database. This means that the classification

model is dynamic. All information items, their types and their domain values can be changed by the client (in fact, by a privileged user and not by any user). This means the classification model can evolve in time without intervention of the knowledge engineers. Flexibility is necessary, because we experienced in previous projects that in the knowledge acquisition phase experts think they will need many information items to describe cases sufficiently. In practice however, queries are most of the time composed of at most six to eight items. In daily use the classification model shrinks to an optimal size. Therefore the dynamic aspect of the classification model is a major feature over traditional approaches where the model is part of the software.

The next aspect to be mentioned is the reduction of the search space. Brute force search through all case law descriptions will take too much time, especially when several thousands of cases are stored in the database. Therefore the search space is reduced as follows. The process is depicted in Figure 1.

Firstly, the information item/domain value combination from the search profile with the lowest hit rate (number of occurrences) is determined, with respect to the fact that one information item may be used several times in a profile (the profile contains 'or' constructions). The set of cases for which the case model contains this combination is called set A. Secondly, determine the combination with the second lowest hit rate. Call all case models containing this combination set B. Determine by means of an evaluator whether it is worth while to reduce the search space to the intersection of set A and B and repeat the process or to evaluate all cases included in set A completely. When the process is repeated, the cases containing the combination with the third lowest hit rate will be considered, called set C, in relation to the intersection of sets A and B.

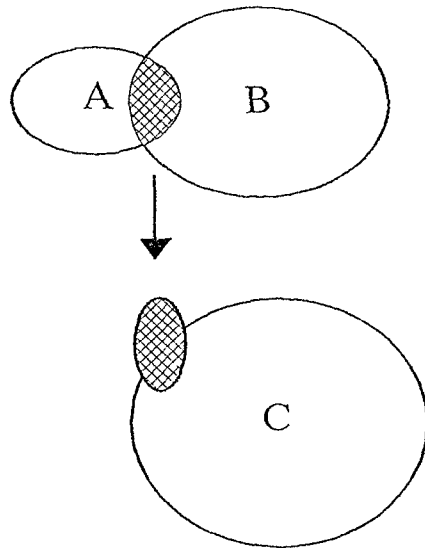


Figure 2 The principle of the search space reduction process.

The use of case law in the field of the legal profession is based on similarity between cases. Similarity is an arbitrary concept. Therefore flexibility of the similarity concept is a desirable thing. It should be possible to retrieve cases that are slightly different from the case at hand. In the terms of the prototype: it is desirable to fetch case models which differ a little bit from the search profile. This flexibility is implemented by a so called 'mismatch rate', which influences the search process. The user can specify how many mismatches are allowed in comparing the information item/domain value combinations between the search profile and the case law descriptions. With this facility the number of cases to be selected can be influenced. When a query only selects one or two cases, the use of the mismatch facility can cause more cases to be selected, cases with

a high level of similarity with the search profile. Of course the presentation of these case is accompanied by an explanation of the mismatch.

6. Conclusions

The prototype is very easy to work with and has proven to be useful for lawyers. Lawyers often do not need a system to formulate a decision for a given case, but a system to retrieve all cases closely related to a newly prepared case. The major advantages of the prototype are therefore not to be found in reasoning on basis of case law. They are rather to be found in the dynamic character of the classification models and in the use of AI techniques in the search space reduction.

The dynamic character of the classification model was experienced by the experts as a major advantage. This dynamic character creates the possibility to let the classification model evolve and adapt to the changes in the experts view on case law, as the system is being used. This way a step wise perfection is reached.

In the near future research will be carried out on two aspects. First, methods will be developed to refine the evaluator function automatically during the operational use of the system, in order to base the estimation on the actual content of the database. A more ambitious project to be carried out is to classify cases automatically based on textual descriptions of the case.

7. References

- [Visser 90] Visser, P.R.S., Van den Herik, H.J., Schmidt, A.H.J. and de Wildt, J.H. (1990), *Het modelleren van Casus*. Proceedings of the third Nederlandstalige AI-conferentie '90 (eds. H.J. van den Herik and N.J.I. Mars), Kerkrade, 13th and 14th June 1990.
- [Herwijnen 89] Van Herwijnen, J., Van Houten, E.G., Houtsma, M.A.W. and Romkema, H.M. (1989), Implementatie van een regel-gebaseerd kennissysteem in een relationele database-omgeving. *Maandblad Informatie*, Vol. 32, No. 1, Kluwer Bedrijfswetenschappen, Deventer.
- [Houten 89] Van Houten, E.G. and Van Bruggen, R.D., *Subsidieregelingensysteem: kennis in een database*. Proceedings of the second AI Toepassingen Conferentie '89 (ed. H.J. van den Herik), The Hague, 28th and 29th November 1989.
- [Hage 90] Hage, J.C., *Meta-Kennis voor juridische kennissystemen*. Proceedings of the third Nederlandstalige AI Conferentie '90 (ed. H.J. van den Herik, N.J.I. Mars), Kerkrade, 13th and 14th June 1990.
- [Wildt 89] De Wildt, J.H., Quast, J.A., *Kenmerken van LEIDRAAD, een kennissysteem voor de werkloosheidswet*. Proceedings of the second AI Toepassingen Conferentie '89 (ed. H.J. van den Herik), The Hague, 28th and 29th November 1989.

TOWARDS A WORKBENCH FOR THE LEGAL PRACTITIONER

Joost Breuker
Laboratory for Computer Science & Law
University of Amsterdam
Kloveniersburgwal 72
1012 CZ Amsterdam
The Netherlands

Summary

The major thesis of this paper is that in representing regulation knowledge, the knowledge about the world to which the regulations refer should be separated out from the representation of the regulations proper. These two types of knowledge can communicate by common terms, i.e. the abstract view of the legislator on how regulations can constrain and control 'what can or should happen in the world'. In this sense, the legislator has some (implicit) model of how a world works. A world is a domain of legislation, e.g. transactions on property, public organisations, traffic works. Legal principles and institutions are intended to constrain the possible behaviours in a domain to accomplish abstract goals (e.g. justice, safety, profit, etc.). This legal view on the world is very practical and should not be too easily confused with the role of 'common sense' in legal reasoning. Common sense problems refer to the fact that the model may not (directly) fit reality. If the model is too often incorrect, the regulations are ineffective. Preliminary however is the question of making these models explicit and if possible: generative in the sense that they can generate or predict behaviour. This is what AI is about: building 'qualitative models' [Clancey 88].

This view underlies the specification of functions and structure of a prospective legal practitioner's workbench. The core of this workbench consists of a legal ANIMATOR which generates or constructs event structures which are interpretable by formal regulations. In order to be able to serve multiple functions, intelligent processing modules (classifiers, planners) can be used as pre- and post-processors to the ANIMATOR. In particular, an intelligent dialogue front end which enables 'regulation tale spinning' is required for users who seek legal advice. The actual development and implementation of this framework is beyond the scope of our laboratory (it is part of an Esprit proposal), but it functions as a general paradigm for current research as e.g. in legal contract generation (in the ALDUS Esprit project), or testing implications of a new traffic regulation (RVV-90).

1. Background and motivation

Traditionally, two major categories of applications of informatics, and in particular AI, to the domain of law have been identified [McCarty 82]:

storage and retrieval of legal sources (texts)

legal analysis systems

Examples of these two categories are abundant in the literature, e.g. see [Bing 84], [Sergot 87b]. Two less frequent types can be added:

legal design tools (e.g. drafting contracts and regulations)

(intelligent) teaching and training systems

These types of applications -except for the last one- cover to a large extent practice in law, i.e. they support activities in the law office [Sergot 87b]. In contrast with the conventional systems for information management functions, like text processors and data bases, these knowledge based systems (KBS) have highly specific functionalities and architectures.

For storage and retrieval data-base technology is used, 'intelligently' enhanced with conceptual, dialogue front-ends [Bing 87]; [McKeown 85]. Legal analysis systems may have a larger variety of architectures, but in practice straightforward decision trees are prevalent over logical [Sergot et al. 86b] or conceptual modelling based ones [McCarty 89]. Aids in legal design are an underdeveloped area and are text based with very limited functionality (contract drafting). Intelligent support for designing and drafting regulations I have not found in the literature. Systems for testing consequences of (newly designed) regulations can be an important component of a more general tool (e.g. ExpertiSZe [Nieuwenhuis et al. 89b], see also further). Except for (important) issues in logics and for approaches like case based reasoning these architectures are of little (AI) research interest. Standard knowledge engineering techniques are used.

This is most likely due to a rather restricted view on the required functionality. These restrictions seem to be aimed at *minimizing the role of common sense knowledge in legal reasoning*. Indeed, where multifunctionality is required as in LEX common sense knowledge is explicitly represented [Haft et al. 87]. Most applications constructed so far deal adequately with small sets of specific and well defined tasks, but users require a wider range of functionalities.

First, most legal analysis systems are aimed at processing cases, i.e. to provide decision support about the legal consequences of a real or hypothetical case. However, this is not the typical situation in legal *advice and information serving*. In general there is no case yet, and much legal advice is about possible courses of actions, which are enabled, excluded or prescribed by regulations. For an advice the intentions of the client are the pivoting issue. These may vary from some very global constraints ("What can I do to prevent them building a highway in front of my house") to an actual case ("What will the judge say"). Other typical advice questions are: "What is the cheapest way to get a divorce?", "Can I deduce donations to the church from my taxable income?". These questions aim at the *construction* of a plan which, for instance, optimises personal benefit without coming into conflict with the law, instead of assessing or 'judging' a case (see e.g. EPS-II [Schlobohm & McCarty 89]). To answer those questions the knowledge has to support multiple tasks, like planning, classification, assessment. Moreover, obtaining the intentions and concerns of clients, which is of ultimate importance to provide control to potential inferences is not an explicit, flexible function in these systems, which requires negotiating capabilities [Pollack 86].

A second issue is that both legal analysis systems and design support systems should be complemented with retrieval and storage components to be of practical use. Law

practice is largely case based, whether the cases are the client's or come from precedence. In case based reasoning systems some integration of these functions is accomplished [Rissland & Ashley 87].

In the next sections I will present a short description of the abstract functions involved in many legal reasoning tasks, and an architecture which supports these functions in an extensible way.

As indicated earlier a major reason for having rather specialised legal reasoning (support) systems is probably to reduce the role of common sense knowledge and reasoning as possible. E.g. [McCarty 83b] has explicitly chosen a domain of law for the TAXMAN systems which common sense knowledge plays a minor role. Other ways to diminish the problems related to the use of common sense knowledge is to select domains where this knowledge is highly 'operationalised', i.e. where the application and control of regulations have been delegated to agents with no specific background in law, e.g. to civil servants for administering social security benefits [Nieuwenhuis 89a]; [de Hoog 89]; [Bench-Capon et al. 87], or (income) tax law. The open texture of many concepts have been operationalised in such detail to allow decision tree like structures, which contain mixtures of legal and common sense knowledge. A third way to make legal decision support systems feasible is to leave as much as possible the common sense reasoning aspects of a domain to the user, because that is exactly where humans have little problems, and where the machine (AI) gets lost.

However, the term common sense knowledge has been used too loosely in the literature. The distinctions between real common sense (see e.g. [Steels 88b] and the particular views that laws, regulations, or legal practice have on the world have been confounded. In order to construct regulations, one has to have an abstract *model* of how a particular world -domain of law- works and what effects a (new) regulation would have. Therefore, it seems better to distinguish between regulation knowledge, a model of the world to which these regulations apply, and common sense knowledge. For all practical purposes one can see strict common sense knowledge as those issues which are left to the user to decide what is the case.¹

In many legal reasoning systems the representation of laws and regulations is intertwined with extensions to the world to which the regulations refer. Mostly this is done in an implicit way. However, in more research driven systems as TAXMAN and LEX, there is an explicit recognition of the role of world (common sense) knowledge, but there is no sharp distinction. McCarty states that: "There are many common sense categories underlying the representation of a legal problem domain: space, time, mass, action, permission, obligation, causation, purpose, intention, knowledge, belief and so on" [McCarty 89, p 180].

There is no doubt that this is the case, but it is also the case that among these categories e.g. permission and obligation appear to be far more inherent to what law is about, than for instance space and mass. What I propose here is to separate far more strictly regulation knowledge from the particular interpretation of objects, states and actions in a part of the world that is implied and required by these regulations. Regulations are applied to (potential) events or situations in the world. The interface between the regulations and the world to be regulated is provided by those high level concepts as obligation, liability, etc. as analysed by philosophy of law [Hart & Honoré 85]. Hamfelt & Barklund (1990) consider these concepts to be part of metalevels of a (logical) regulation knowledge base, but it may as well be a

metalayer of the world knowledge base, which views the world of human action from a legal point of view. The difference between this conception and McCarty's (1989) RLL is the separation which enables modelling of worlds in a more transparent way, and distinguishing between events in the world and application of regulations.

2. A functional view on legal reasoning²

2.1. *Legal reasoning as modelling*

Legal reasoning systems are thus far highly task specific, which provides an emerging field of research with the identification of some of the major problems, and bottlenecks, but as pointed out above, it can also imply that the solutions proposed are too specific. Reflecting on what legal reasoning means and requires in practicing law may provide a wider and more 'ecologically valid' horizon. These broad perspectives may suffer from underestimating many of the lower level problems. Therefore, the functionality and architecture to be discussed are a framework for individual research and development projects, rather than an aim in itself.³

In practice one would like to see that practitioners of law could have systems-workbenches- at their disposal, which would support them in much the same way as CAD/CAM does for civil engineers. These workbenches consist of more or less tightly coupled tools of a very generic nature, which can be specialised and tailored to individual needs by user defined libraries of information and knowledge. Essentially, what such workstation environments allow the user to do is first to construct a model of some complex object or system and then run the model, i.e. to derive implications.

2.2. *Legal tale spinning*

In legal analysis the model to construct represents a case or a client's concerns and situations in terms that can be interpreted by the regulations. This model is an event structure [Haft et al. 87]. Event structures obtain their meaning -or rather: theme- by abstracting out the relevant issues. In narrative discourse this is called the plot. Models of discourse have been developed which separate out these plots or macro structures partly on the basis of conventions (story grammars) and general theories about human action, intention and physical worlds [van Dijk & Kintsch 83]. In an analogous way one could view constructing the plot of a legal problem as 'legal tale spinning'. Besides general conventions of what constitutes legal argument, the legal tales may differ widely from one domain to another one. For instance, in tax law one is not (directly) concerned with the question whether incomes have been obtained in an 'honest' way, i.e. the tax law practitioner may not see criminal points in his tax-law tales. Running the model consists of applying the regulations (see section 3).

Constructing event structures looks different from constructing models of systems as e.g. in civil engineering. There the functions are given and the structure has to be modelled; in legal reasoning, the 'behaviour' of a system or world has to be modelled in functional terms on the basis of an -often implicit- structural model of that system. This structural model is a description of how the world 'works' in terms of actions, agents, objects, etc. from the point of view of the (objectives of) a domain of regulation. For instance, a tax law view on financial transactions is a particular abstraction about the worlds of economy and of personal income.

2.3. Legal animation

The core of this workbench consists of an 'ANIMATOR' which allows the user to model or experiment with situations and legislation in order to 'compute' their consequences. In this sense, the ANIMATOR can be the basis for both a legal analysis system which analyses the consequences of existing regulations, and a tool which supports the design of regulations, dependent on whether the user specifies the description of situations and events, or tests changes in regulations by generating situations and events semi-exhaustively.

This core can be enhanced by specific interpreters -e.g. planners for planning courses of actions, and specific user interfaces -e.g. a dialogue front-end which enables a user to specify his intentions, concerns and data in a goal directed way. Each of these extensions supports different tasks or functions which are based upon legal reasoning.

The major role of the ANIMATOR is to allow making inferences about states and events in a world. This can be illustrated by making explicit what it requires to answer simple question about traffic regulations: e.g. whether it is permitted to park a car on the lefthandside in Holland.

First there is no regulation which forbids or enables explicitly this action. In legal reasoning it is hardly ever the case that a legal fact immediately confirms or rejects a proposition. By consulting a legal knowledge base (KB) the answer could be either "don't know", or even "allowed-by-being-not-forbidden" (c.f. negation by failure). However, by considering what it implies in the physical world to park a car there is a specific conclusion. Parking on the lefthandside would imply crossing the road on the side of the oncoming traffic, which is forbidden, etc.⁴ However, this answer is incomplete, and if the concern of the user is whether there is any possibility, the case of a one way street is one of these, and there may be many others. The point is that in order to answer this seemingly simple question requires large amounts of inferences of what possible actions and what possible objects can be involved in the world.

In this respect, legal reasoning differs from problem solving in other types of domains, because any systematic exploration of possible actions and situations may easily lead to combinatorial explosions. In typical problem solving tasks these potential explosions are controlled by problem solving strategies. However, legal reasoning rather resembles question answering, in which the intentions, concerns and data of users have to be exploited as to provide local goals and constraints in exploring all possible ramifications in the world and their legal consequences. This is why a naked ANIMATOR requires relatively intelligent front end tools, in particular those which are capable of recognizing plans and intentions of users [Luria 87]; [Breuker 90b].

3. Architecture of the legislation ANIMATOR

As can be interpreted from figure 1 the major partitioning of knowledge bases of the ANIMATOR rests on the distinction between the representation of the regulation knowledge (REGULATION KB) and the description of the world (WORLD KB) which the regulation refers to and/or which is assumed.

These KB have a different functionality. The WORLD KB is used to generate or comprehend the description of situations and events. The REGULATION KB contains the descriptions of the regulations. Probably, the most suitable medium of representation for regulations proper is a logical one, but not necessarily one that involves modal operators, because what these modal operators predicate may be more easily represented as consequences in the world: an obligation or an interdiction restricts legal actions.

3.1. Representing worlds

The WORLD KB contains a number of abstract knowledge types ('epistemological primitives'; cf. Brachman 79) which can be used by the SITUATION SPECIFIER. This module contains the required inference mechanisms and strategies to construct or classify specific situations in terms of its world knowledge. In restricted domains where the WORLD KB refers to a closed world and can be complete, the SITUATION SPECIFIER'S inference engine can be a simple instantiator or classifier.

The SITUATION SPECIFIER constructs a dynamic data-structure: the SITUATION DESCRIPTION. The situation specifier knows how to compose situations from states, how to interpret actions as changes of states and combine these into events and how to identify roles of agents and objects in these events, resp. states.

This requires not only inference mechanisms for construction and identification of these entities and structures, but also a framework to keep track of which of these events or situations are past, current, hypothetical, etc. In other words a time calculus and truth maintenance should be part of the SITUATION SPECIFIER resp. WORLD KB. The SITUATION DESCRIPTION is an event structure, and consists of two 'orthogonal' parts: time and (pseudo) causation dependent sequences of events, i.e. the history and a relevant, e.g. current situation, describing the states of objects and agents. This description may be the description of a particular instance ("My car parked in Amsterdam on the left side of the ... street on .. at ... hrs") or of a generic situation ("A car parked on the left side of a road (in a city) in Holland").

3.2. Applying legal knowledge

This situation is the point of departure of computing the regulation CONSEQUENCES, i.e. what may or should follow given the state of affairs. The event structure, i.e. the history provides the relevant antecedents for computing these consequences.

CONSEQUENCES are a dynamic data structure generated by the REGULATION APPLIER which applies one or more regulations from the REGULATION KB to the situation. Note that this is only possible if the concepts (terms) used to describe situations map onto those of the regulations.⁵ Specifying the WORLD KB and the REGULATION KB therefore cannot occur independently, and requires editing tools which preserve the identity of concepts and terms.

Consequences may also be based on precedence. Although the ANIMATOR is based upon the representation of explicit regulations -whether documented or abstracted from cases-, the ANIMATOR should be capable of bringing to bear the relevant

cases: not only to refine the CONSEQUENCES, but may also to serve as concrete examples to the user to understand the meaning of the CONSEQUENCES.

The fundamental problems in matching cases with a SITUATION DESCRIPTION are well known and discussed in the literature, ranging from case based reasoning to connectionist matching [e.g. Rose & Belew 89], which does not mean that they are solved, but that pragmatic solutions exist, which may not be foolproof.

The ANIMATOR can be used by domain specialists to experiment with, or compare regulations by computing their consequences. In this way the ANIMATOR is a workbench for designing regulations. In ALDUS, an Esprit pilot project in which we participate, these functions for drafting commercial contracts will be further investigated.⁶ In figure 1 the architecture of the ANIMATOR is depicted.

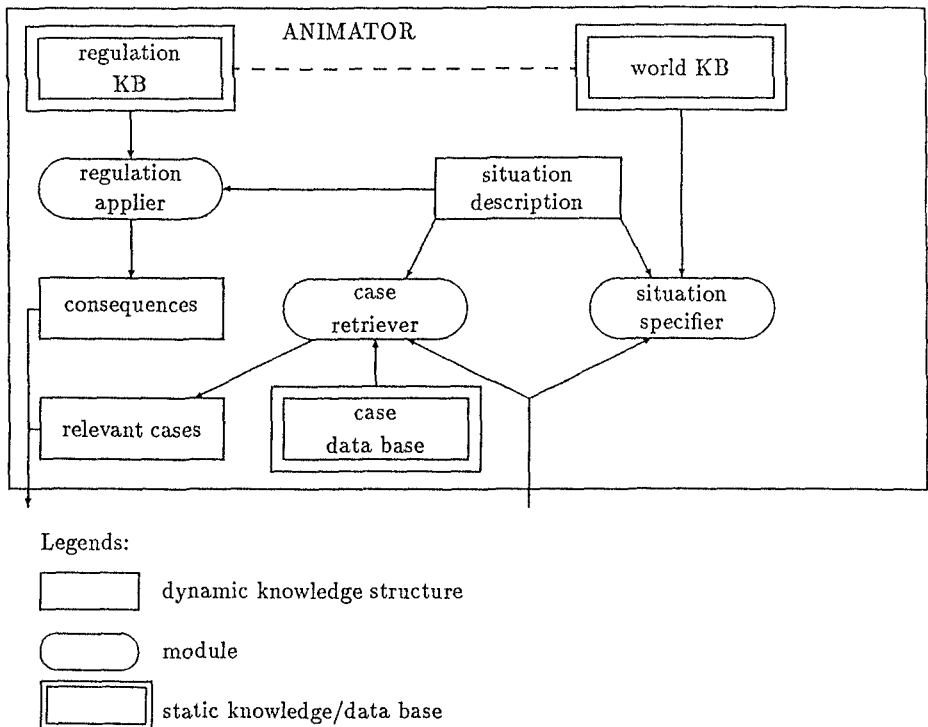


Figure 1: Architecture of the regulation ANIMATOR

4. An example: the animation of traffic regulations⁷

To illustrate how things may work in this workbench I will present some examples of the domain of (Dutch) traffic regulations, i.e. the RVV-90 which is intended to replace the current RVV. The RVV is only concerned with the rules of traffic behaviour; not with issues of civil and criminal law.

The problem is how one can test the consistency and intended consequences of this body of regulations, i.e. how one can find out that there are no omissions or not intended side effects. A more practical question is to check in what respects the regulations of the new RVV-90 are different from the current one. Regulations can only be fully compared by their implications; not by their structure. However, it is impossible to check all these implications 'by hand', as the example about lefthandside parking may have shown.

The principle of testing potential differences between regulations by the machine is relatively simple. There are two REGULATION KB. One containing a representation of the current RVV, the other one of the RVV-90. It should be noted that the structures of both regulations are completely different. The RVV-90 is highly abstract and generic, whereas the current RVV contains many redundancies, is relatively concrete and less systematic. Both refer to almost the same world as can be assumed from the terms used and in particular from the sections which contain the definitions of terms.

By defining this world of traffic and GENERATING EXHAUSTIVELY the possible traffic situations, one can assess what each regulation has to say about that situation. The generation of all possible situations is based upon the distinctions as implied in the RVV. For instance, there are only 5 types of roads; there is only one type of crossing (taking symmetry into account); there are only 5 relevant speeds; there are only 12 types of actions, which can be abstracted in 4 major types, etc [Nysingh 90]. However, a traffic situation consists of a combination of these types, and generating blindly all these combinations may still easily lead to combinatorial explosions, in particular when the traffic signs are taken into account. Therefore, further reductions have to come from analysing which objects or actions have no context dependent effects in the world, i.e. are not implied in any other action or object. Note that the inverse function -identifying a 'real life' description of a traffic situation- would not yield the same problems in combinatorial explosion, but rather problems in matching terms used and terms known by the system. This is a reduced common sense knowledge problem. For instance, it can be left to the user to state that his Trabant is a car. Problems in understanding what the combinations of terms used by the user mean are the classical natural language interpretation problems [Haft et al. 87].

5. Conclusions

The major problem in this view of constructing multi-functional tools for the legal practitioner is the model of the world that has served to provide the meaning to regulations and has lead to decisions in cases. In principle, building a model of the world to which the regulations refer looks like solving a puzzle. The knowledge engineer may know what the picture is about -e.g. traffic-, but he can only reconstruct the perspective and grain size level by looking at individual pieces of this puzzle: the rules, or rather, the terms and definitions in these rules. Because many terms are implicit, because they are not object of regulations, this enterprise is best labeled as a 'rational reconstruction' of the world the designers had in mind when drafting regulations. In this respect domains of law are not different from other types of domains, e.g. medical ones [Clancey 88]; [Breuker 89]. The major difference is that in other domains the nature of models and abstractions is often made explicit in natural language and/or mathematical models. In law the emphasis is on the parts of the world that can be or should be controlled; the autonomous parts are left

implicit. The question is also how much support the study of law and the practice of law can contribute to this rational reconstruction. Philosophy, history and comparison of law are in this respect major sources.

6. Notes

1. Theoretically this means that common sense knowledge is extensionally defined, whereas model knowledge has intensional semantics.
2. This section and the next one are largely based upon an ESPRIT-II proposal, REGALE, aimed at constructing a shell for regulation information servers, which was technically accepted, but which has not been granted (for commercial reasons). I acknowledge the contributions of Martijn den Uyl (Bolesian) and Marek Sergot (Imperial College) to these ideas.
3. Developing the core of such a workbench would require the size of an Esprit project, i.e. about 60 manyears.
4. Of course, there are exceptions to this interdiction, as during overtaking, but this action cannot imply parking.
5. A one to one correspondence is not required, because the situation description may contain some terms which are not relevant for a particular situation.
6. This project is aimed at the study of functions of and markets for commercial contract generation. Partners are: University of Amsterdam (LRI), Imperial College (GB), BIKIT (B), DiDaEl (I), Kluwer (NL), Machine Intelligence (GB).
7. This section is based upon a request of the Stichting Wetenschappelijk Onderzoek Verkeersveiligheid (SWOV; Foundation for Research in Traffic Safety) for a feasibility study on testing the consequences of a new body of regulations for traffic control (RVV-90).

7. References

- [Bench-Capon et al. 87] T.J.M. Bench-Capon, G.O. Robinson, T.W. Routen, and M.J. Sergot (1987). Logic programming for large scale applications in law: A formalisation of supplementary benefit legislation. In: *Proceedings of the First International Conference on Artificial Intelligence and Law*, pp. 190-198, Boston, May 1987.
- [Bing 84] J. Bing (1984), *Handbook of Legal Information Retrieval*. North-Holland, Amsterdam.
- [Bing 87] J. Bing (1987) Designing text retrieval systems for 'conceptual searching'. In: *Proceedings of the First International Conference on Artificial Intelligence and Law*, pp. 43-51, Boston, May 1987.
- [Brachman 79] R.J. Brachman (1979) On the epistemological status of semantic networks. In: N.V. Findler (ed.), *Associative Networks*, New York, Academic Press.

- [Breuker & Wielinga 89] J.A. Breuker and B.J. Wielinga (1989) Model Driven Knowledge Acquisition. In: P. Guida and G. Tasso (eds.), *Topics in the Design of Expert Systems*, pp. 265-296, Amsterdam, 1989. North Holland.
- [Breuker 90] J.A. Breuker (1990b) (ed.). *EUROHELP: developing intelligent help systems*. EC, Kopenhagen.
- [Clancey 88] W. J. Clancey (1988) The role of qualitative models in instruction. In: J. Self (ed.), *Artificial Intelligence and human learning*, London, 1988. Chapman and Hall.
- [Dijk, van & Kintsch 83] T.A. van Dijk and W. Kintsch (1983), *Strategies of Discourse Comprehension*, New York, Academic Press.
- [Haft et al. 81] F. Haft, R.P. Jones, and T. Wetter (1987) A natural language based legal expert system for consultation and tutoring: The LEX project. In: *Proceedings of the AFIPS National Computer Conference*, pp. 75-83, Chicago, May 1981.
- [Hamfelt & Berklund 89] A. Hamfelt and J. Barklund (1989) Metaprogramming for representation of legal principles. In: *Proceedings Meta90*, pp. 105-122. Katholieke Universiteit Leuven, 1990.
- [Hart & Honoré 85] M.L.A. Hart and T. Honor (1985), *Causation in the law*. Oxford University Press, Oxford, GB, second edition.
- [Hoog 89] R. de Hoog (1989). Een expertsysteem: bijstand voor bijstand. *Informatie en informatiebeleid*, 7.
- [Luria 87] J. Luria (1987) Concerns. In: *Proceedings of the 10th IJCAI*, pp. 1025-1031, Los Altos, C.A., Morgan Kaufman.
- [McCarty & Sridharan 82] L.T. McCarty and N.S. Sridharan (1982), *A computational theory of legal argument*. Technical Report LRP-TR-13, Laboratory for Computer Science Research, Rutgers University, Brunswick, NJ.
- [McCarty 83b] L.T. McCarty (1983b) Permissions and obligations. In: *Proceedings of the 8th IJCAI*, Palo Alto, CA, 1983. Morgan Kaufmann.
- [McCarty 89] L.T. McCarty (1989) A language for legal discourse I: basic structures. In: *Proceedings of the 2nd International Conference on AI and Law*, Vancouver, 1989. ACM.
- [McKeown 85] K.R. McKeown (1985) Discourse strategies for generating natural language text. *Artificial Intelligence*, 27:1-41.

- [Nieuwenhuis 89] M.A. Nieuwenhuis (1989). *TESSEC, een expertsysteem voor de Algemene Bijstandswet*. Kluwer, Deventer, NL.
- [Nieuwenhuis et al. 89b] M.A. Nieuwenhuis, G.J. van 't Eind, M.G.K. Einerhand, and J.G.J. Wassink. (1989b) ExpertiSZe: een expertsysteem voor beleidsvoorbereiding. In: J. van der Herik (ed.), *Proceedings van de AI-Toepassingen Conferentie*. Stichting Informatica Congressen.
- [Nysingh 90] J.G. Nysingh (1990). Verslag van verkeersacties in het RVV 1990. Master's thesis, Lab. for Law & Computer Science, University of Amsterdam, 1990.
- [Pollack 86] M.E. Pollack (1986). Inferring Domain Plans in Question Answering. PhD thesis, University of Pennsylvania, 1986.
- [Rissland & Ashley 87] E.L. Rissland and K.D. Ashley (1987) A case-based system for trade secrets law. In: *Proceedings of the First International Conference on Artificial Intelligence and Law*, Boston, May 1987.
- [Rose & Belew 89] D.E. Rose and R.K. Belew (1989) Legal information retrieval: a hybrid approach. In: *Proceedings of the 2nd International Conference on AI and Law*, Boston MA, 1989.
- [Schlobohm & McCarty 89] D.A. Schlobohm and L.T. McCarty (1989) EPS-II: Estate planning with prototypes. In: *Proceedings of the 2nd International Conference on AI & Law*, pp. 1-10, Northwestern University, Boston MA, 1989.
- [Sergot 87b] M.J. Sergot (1987b) Logic programming and its application to law: a brief tutorial introduction. In: C. Walter (ed.), *Computing Power and Legal Language*. Greenwood/Quorum Press.
- [Sergot et al. 86b] M.J. Sergot, F. Sadri, R.A. Kowalski, F. Kriwaczek, P. Hammond, and H.T. Cory. (1986b). The British nationality act as a logic program. *Communications of the ACM*, 29:370-386.
- [Steels 88] L. Steels (1988) Steps towards common sense. In: Y. Kodratoff (ed.), *Proceedings of the ECCAI 88*, London, Pitman.

DEVELOPMENT OF A KNOWLEDGE BASED SYSTEM FOR THE COMPARISON OF NATIONAL SOCIAL SECURITY SYSTEMS OF MEMBER STATES OF THE EUROPEAN COMMUNITY IN DYNAMIC PERSPECTIVE

Debrock, K., Lemmens, V., Robben, F., Van Buggenhout, B.¹
Institute of Social Law and Centre for Law and Computer Science (I.C.R.I.)²
Katholieke Universiteit Leuven
Tiensestraat 41, B-3000 Leuven
Belgium

Summary

At the Institute of Social Law and the Centre for Law and Computer Science, a computer supported model is developed that will attempt to measure the legal and economic effects of changes in national regulations and of co-ordinating E.C.-measures in the field of social security. This paper reports on the aims and basic principles of the methodology that is applied.

1. Introduction

Partly as a result of the strongly increasing scientific research in the area, legal knowledge systems have been introduced, during the last years, to a large group of potential users. Usually the attention focusses primarily on the function of legal knowledge systems as an aid to the supply of situational legal advice.

In this context research has been carried out since 1986 at the Faculty of Law of the K.U. Leuven,³ on the possibilities of the use of informatics in legal advising,⁴ more specifically in the area of social security. This sector applies indeed in particular for automation. The complexity and extensiveness of the subject, the diversity, technicity and fast evolution of the regulations, and the lack of co-ordination and codification are elements that complicate the accessibility and the application of social security. The introduction of informatics, not only for rationalization of the data administration in social security administration, but also, in a more advanced way, using intelligent knowledge systems, can optimize the accessibility and efficiency.⁵

More specifically, in previous research of the M.I.J.A.-project, a methodology for the development of legal advisory systems was elaborated, and was applied with success to a number of specific domains.⁶ Starting from the basic idea that it should be possible to apply the same formalized knowledge for various purposes, given the development of different, adapted user interfaces, the research group has aimed since the end of 1989 at a generalization of the methodology, in view of its application for purposes of preparation and evaluation of legal and socio-economic aspects of policy-making.

The value of a knowledge system in legal policy preparation and evaluation is primarily situated in the possibilities which it offers for consistency checking and for the detection of connections. Subsidiary, knowledge systems can be used as a support for descriptive or analytic comparative law.⁷ As knowledge systems contain information in a clearly structured form, comparison and a general understanding of different regulations is simplified. In a further stage one can think of the support of functional comparative law, where the descriptive comparison is followed by a

synthesis, proposing a solution for particular problems, e.g. a co-ordination, reform or unification of law. Particular fields in different national regulations, formalized in knowledge systems in a uniform structure, could be confronted with each other in a computer-supported way in view not only of an analysis of the causes of certain differences or problems, but also of the elaboration of solutions.

For the estimation of the socio-economic effects of policy alternatives knowledge systems open up new perspectives with regard to micro-simulations. Given the continuously increasing differentiation of regulations, the traditionally used macro-approach indeed does not seem to satisfy any longer. More and more, meso- and micro-models are needed, in which the variation of the results is no longer aggregated to a single average decision, but in which the attempt is to simulate, in an as accurate way as possible, the final decision with regard to a (group of) client(s). This micro-simulation function demonstrates a strong affinity with the interpretational function of a legal knowledge system, and can be executed by such a system, given a sufficient flexibility of the knowledge base and an adapted user interface.⁸

The conviction of the M.I.J.A.-research group that legal knowledge systems can have a supporting function in the field of preparation and evaluation of the legal and socio-economic aspects of policy, is reflected in a research project⁹ that was started in October 1989. Starting from the above described experience relating to legal advisory systems, the research project aims to elaborate a computer-supported model in order to measure the legal and possible economic effect of (planned) changes in national regulations and of co-ordinating E.C.-measures in the field of social security law. As an example the model will be applied to the unemployment regulations of a number of member states.

2. Description of the scope and aims of the research

Since the foundation of the European Community in 1957, one of its most important aims has been the realization of an internal market with free movement of persons, goods and capital. During the last years the public opinion has become fully aware of this aim, also because of the publication of the Cecchini-report [Cecchini 88], in which is confirmed that an advanced European unification will enhance the standard of living in a important way.

The advancing European integration, with an enhanced mobility of persons, and the growing internationalization of professional and societal life, has led to the application of national social security systems beyond the national borders, and subsequently to the need for a better knowledge of these systems. Documentary knowledge bases or, even better, intelligent computer-advice systems are valuable assets in relieving this need for information.

In addition to the possibilities in the field of situational advisory systems, there is also room for integrated knowledge systems with information on the various national social security systems, in view of policy support. An internal market without limitations on movement of persons, goods and capital will indeed be advantageous for the most attractive regions. In that view, the level of social protection and the amount of the social benefits are important elements. The abolition of the internal borders in the European Community must therefore be accompanied by adapted co-ordinating and (possibly) harmonizing measures in the area of social security. This is

the responsibility both of the member states and the European Community itself. In order to gear these measures optimally to the changing circumstances, a flexible method is required to compare the social security systems of the different member states in a dynamic way, and to determine, in an integrated way, the legal and economic influence of (planned) measures at national or European level. Moreover, a system could be developed which detects problems through a confrontation of the different legal systems, and which gives a clear insight into the causes of these problems, and thus in possibilities for solutions.

The research project regarding the development of a computer-supported model for the comparison and evaluation of national social security systems in dynamic perspective, which will be described hereafter, is subdivided into two parts. A first part contains the comparative study of the social security systems of a selected number of E.C.-member states, and the modelling of the legal knowledge which is collected. A second part concerns the development of a knowledge system that contains this knowledge and permits to measure the effects of hypothetical or real changes in national or European regulations.

In a first stage the determination and evaluation of the effects of legal nature will be focused. A further purpose however is the application of the comparative tool for the determination and evaluation of micro- and macro-economic effects.

In what follows the elaborated method for the modelling of the legal knowledge will be described and justified. Further, the computer-technical implementation of this model will be discussed shortly and illustrated with a concrete example.

3. Description of the method for the modelling and representation of the legal knowledge

3.1. Points of departure

The various national unemployment regulations are analyzed and modelled in view of the legal and possible economic comparison that will be realized afterwards. Deliberately, the option has been taken not to make a structural comparison in abstracto of the different legal systems, certainly not in a first stage. The difficulties arising in the elaboration of a reliable supporting knowledge system in this domain do not seem to be compensated by its practical relevance. However, there has been taken into account that the applied method of knowledge representation could also be used in a further stage for such purpose.

The aim of the present research is the development of an instrument which permits to

- compare the result of the application of the different legal systems to representative random samples or typologies;
- give insight into the causes of differences;
- investigate possibilities for the elimination of unjustified differences;
- investigate the result of (planned) changes in legal systems introduced by normative actions of a national state or of the European Community.

As the research aims are situated primarily on the methodological level, the emphasis is put on the elaboration of the instrument vertically, i.e. concerning available functionalities, rather than horizontally, as regards the contents of the

knowledge base. Thus the purpose is a prototype with advanced functionalities within a limited and well-defined part of social security law. Target group of the knowledge system are institutional users with a certain foreknowledge of informatics and social security.

3.2. Limitations of the knowledge system resulting from the postulated points of departure

3.2.1. Specification of the knowledge system as regards content

The knowledge system will deal with part of the unemployment regulations of a selected number of member states of the E.C., more specifically the regulations concerning the determination of the amount and the duration of the unemployment benefits. Conditions that concern the existence of entitlement to the benefit are presupposed to be fulfilled. When assuming the existence of entitlement, this implicates that one source of law, viz. case law, will be mainly left aside as source of knowledge for the research.

The selection of the national systems which are involved in the research, goes back on the classification of the different national social security systems in basic types. For each basic type a representative system was chosen, with attention for the available possibilities concerning the gathering of information. In a first stage the following member states have been selected: the Netherlands, Belgium, Great Britain and Germany.

3.2.2. Extent of profundity of the represented knowledge

The instrument that is being developed, is a policy supporting knowledge system, not a legal advisory system. The aim is not to build a set of separate advisory systems for the selected national regulations, in order to determine amount and duration of a benefit in individual cases, in an exhaustive and exact way. The objective of the research is in the first place the determination of differences and similarities when legal systems are applied on typologies or representative cases.

The degree of abstraction and the level of precision, reflected in the representation of the regulations, is influenced by these elements. The rule modules in the knowledge system will only represent the main patterns of the regulations, not the detailed rules which determine every single situation. Special attention is being paid to a uniform method for performing abstractions in the different legal systems.

3.2.3. Explanation facilities and user interface

The transparency of the system for the user is of an extreme importance, in view of the comparative process. The explanation facility must be available in such a way that can be determined how and why a particular conclusion has been formed, at as many stages of the decision-making process as possible.

However, in order to limit as little as possible the application possibilities of the policy supporting system, and given the fact that the target group consists of institutional users with a certain foreknowledge of informatics, less attention will be paid to the development of an easy manageable user interface, which would inevitably limit the application possibilities for the user.

3.3. *The modelling of the comparative elements*

3.3.1. Definition of the problem

In order to realize a valid comparison, the existence of comparable elements is an essential condition. However, legal rules include a large number of concepts whose content and meaning are specifically defined in the legal system to which they belong. Thus, these concepts can have a completely different content or meaning in different legal systems.¹⁰ For the application of legal rules on factual situations, the legal translation of the facts to legal concepts, i.e. the process of legal qualification, is of vital importance. In comparative law the differences in legal qualifications inherent to the various legal systems must be taken into account, in order to achieve a valid comparison. Therefore, when developing a knowledge system that aims to support comparative law, the basic concepts which are used for the description of the different legal systems, must necessarily have the same meaning throughout the entire system.

3.3.2. The method of modelling comparative elements

Because of the problem of qualification, a method has been elaborated, permitting to model the comparative elements and to reveal differences in legal qualifications. The method is based on the reduction of legal qualifications to underlying factual situations. Every legal qualification indeed consists of one, or a combination of several factual elements. If these elements can be retrieved from the various regulations, each rule can be described in terms of identical elements, no matter from which legal system it originates.

For a first general modelling of the investigated parts of the different national unemployment regulations, the attempt has been made to determine all factual elements which are (or may be) of importance in the different systems for the determination of the amount and duration of the unemployment benefits. These factual elements, subsequently called "**basic elements**", are represented as entities, which can be brought in relation with each other. This set of basic elements forms the underlying structure for the description of all regulations that are inserted in the knowledge system. Thus every rule which is inserted, can be described exclusively in terms of these basic elements.

With regard to the selected field of the unemployment regulation, two kinds of basic elements are used: entities and relations between entities. Both entities and relations between entities can have particular characteristics.

The relevant entities are: a person, an occupational situation and an income situation. A person has a number of characteristics: age, sex, his actual living alone or living together with other persons. An occupational situation is described in terms of: the nature of the occupational relation (private or public employer, self-employment, voluntary employment...), in terms of reason of absence of employment (a.o. strike, bad weather, care of children, military service...), begin date and end date, number and nature of the unit used to measure the occupation (hours per day...). Finally an income situation is represented with characteristics such as: nature of the income (income directly connected to a particular employment, substitution incomes...), the quantity of the income, the period of time which covers the income, and the fact of the income being effectuated or not.

Relations can be defined, possibly with consideration for the aspect of time, between persons, between persons and occupational situations, between persons and income situations, and between occupational and income situations. In this way it is possible to represent the (subsequent) occupational and income situations of a person, and the relations between persons. E.g. a person can be related to others by kinship, partnership, can form an economic unity with persons, can be in charge of other persons, or can himself be supported by others.

3.3.3. Basis for comparison: typologies or representative random samples

As the aim of the research consists in the determination of general differences and similarities in the application of the different investigated legal systems, the data which will be inserted into the knowledge system, will not refer to particular, individual cases, as is usual in legal advisory systems, but will refer to representative situations. The data will be determined by elaborating abstract typologies, or by taking representative random samples.

3.4. *Modelling of the legal rules*

The modelling of the various regulations is performed by using the interactive software program PROLOGA,¹¹ which offers extensive facilities concerning consistency checking and optimization of units of rules which are inserted in the shape of decision rules.¹² The legal rules are thus transformed in decision rules, expressed in a strong specification language, specific for PROLOGA. The condition and action part of the rules should only be formulated in terms of the basic elements as described above, or in terms of elements of a specific regulation, derived from these basic elements. Once the consistency check and optimization has been realized with PROLOGA, the set of rules can be generated automatically in production rules, that can be interpreted by the inference engine of the used expert system shell.

4. Description of the computer-technical implementation

4.1. *Technical specifications of the used expert system shell*

The expert system shell that is used, is ADS (Aion Development System).¹³ ADS is an hybrid expert system shell, that supports rulebased as well as object-oriented representation of knowledge, and offers extended possibilities concerning pattern matching, class hierarchy with inheritance of slots and methods, explain facilities, customization of user interface and procedural programming.

The implementation of the research project is done on a micro-computer with a 80386-processor under the operating system MS-DOS. Since ADS is also available on mainframe-environment, the applications are transferable to mainframe-environment if necessary.

4.2. *Computer-technical implementation of the knowledge model*

4.2.1. General description

The knowledge base is built in a strictly modular way, in order to enable a better maintenance and organization. Different knowledge representation techniques are used: production rules for the representation of the legal rules, object oriented

representation for the basic elements. Besides, a mere rulebased knowledge representation would not be sufficient to meet the requirements of a comparative system. Comparing legal systems by comparing their separate rules seems indeed to be an almost impossible task, for each legal system has its own organization, consistency and structure of norms, structure that will only rarely, and always in a fragmentary way, meet the structure of another regulation.

The model of basic elements, as described above, is represented in the knowledge base by means of classes and slots. This structure is common to all regulations and forms the basis for comparison of the different legal systems. The representative samples or typologies, used for the comparison, are described in this structure.

The regulations of the various systems are represented in production rules, which can be generated automatically by the software program PROLOGA, after the necessary consistency check and optimization as described above, or which can be entered manually. The rules are grouped in separate modules and submodules for each national regulation system. As mentioned before they can only manipulate the basic elements or elements derived from the basic elements, characteristic for a specific regulation.

4.2.2. An example

The implementation method can best be illustrated with a concrete example. For this example those rules of the Belgian and Dutch unemployment regulation are extracted that determine the influence of the composition of the family of a person applying for benefits on the amount of the benefit (by determining the category to which he belongs).

4.2.2.1. Description of the relevant basic elements

- class person denotes an entity with a number of properties, viz.:
 - * slot lives_alone: boolean (true = the fact of living alone; false = living together with other persons)
 - * slot sex: multiple choice (man, woman)
- class kinship_between denotes a relationship between 2 persons with the properties:
 - * slot applier_of_benefit: pointer to the instance of class person which applies for the benefit
 - * slot other_person: pointer to an instance of class person
 - * slot nature_of_kinship: multiple choice indicating the nature of the kinship (in this example the choice is delimited to 'spouse' and 'other or no kinship')
- class economic_unity_between denotes a relationship between 2 persons with the properties:
 - * slot applier_of_benefit: pointer to the instance of class person which applies for the benefit
 - * slot other_person: pointer to an instance of class person

4.2.2.2. Description of the Belgian legislation

Art. 160 § 2¹⁴ classifies an applier in one of the categories "living alone", "head of the family" or "person living together" for resp.

"every employee who proves to be in one of the following situations:

1. living alone;
2. living together with a spouse, who does not have any occupational income, nor any replacement income; living together with other persons who have or have not any income, is not of any importance;
3. living together, without spouse (...)."

The same article 160 states in §8 that: "For the application of this article, are regarded as spouses, the persons of different sex who form a factual family." Finally an administrative regulation of the "Rijksdienst voor Arbeidsvoorzieningen" (State Service for Employment)¹⁵ states that "living together requires living under the same roof and sharing the same material interests concerning the household", i.e. having a economic unity.

A number of abstractions is presupposed, in view of the simplification of the example, viz.: the income situation of the partner is not taken into consideration, the situation of single parent families and persons living together with relatives by kinship is excluded from the example, and the applier is presupposed to live with only one person.¹⁶

4.2.2.3. Analysis of the rules in a decision table, generated by means of PROLOGA

applier lives alone	Y		N			
kinship between applier and person with whom he lives	-	spouse		other or no kinship		
economic unity between applier and person with whom he lives	-	Y	N	Y		N
applier and person with whom he lives have different sex	-	-	-	Y	N	-
cat. head of family	-	x	-	x	-	-
cat. living together	-	-	x	-	x	x
cat. living alone	x	-	-	-	-	-
Column:	1	2	3	4	5	6

Figure 1 - Belgian unemployment regulation

4.2.2.4. Generation of the production rules of the decision table within the shell ADS

Every column in the decision table has an according rule in ADS. As an illustration, the 4th column of the table is represented below. Since a lot of rules use basic elements, represented as classes, often pattern matching rules will be used. So, all instances of the class which satisfy the conditions mentioned in the condition part are traced.

The syntax in ADS for the manipulation of slots of classes or instances is the following: <class name> (<instance name>) . <slot name>

Pattern matching rules are expressed by the operator IFMATCH instead of IF. The symbol "->" after a variable means that the variable is a pointer at an instance of the class.

Column 4:

```
IFMATCH kinship_between with
    applier_of_benefit =
    actual_applier_of_benefit
    nature_of_kinship = 'other or no kinship'
economic_unity_between with
    applier_of_benefit =
    actual_applier_of_benefit
kinship_between, economic_unity_between with
    kinship_between.other_person =
        economic_unity_between.other_person
and NOT (person(actual_applier_of_benefit->).sex=
    person(economic_unity_between.other_person-
        >).sex)
and person(actual_applier_of_benefit-
    >).lives_alone = false
THEN category = 'head of family'
END
```

4.2.2.5. Description of the Dutch legislation

The "Toeslagenwet"¹⁷ also distinguishes the situations where the applier is married resp. unmarried, and applies different consequences concerning the amount of the benefit.

Unmarried persons of different or equal sex, who have a permanent, common household, are considered to be married for the application of these regulations.¹⁸

Again, a number of abstractions is presupposed, viz.: the applier is presupposed to live with only one person, the situation of single parent families and persons living together with relatives by kinship is excluded from the example, no iterative element is taken into account.

4.2.2.6. Analysis of the Dutch regulation in a decision table

applier lives alone	Y	N		
kinship between applier and person with whom he lives	-	spouse	other or no kinship	
economic unity between applier and person with whom he lives	-	-	Y	N
cat. married person	-	x	x	-
cat. living alone	x	-	-	x
Column:	1	2	3	4

Figure 2 - Dutch unemployment regulation

4.2.2.7. Conclusion

When both decision tables are put together, it appears a.o. that the factual element of the sex of the applier and the person with whom he is living together without being married, plays a determining role in the Belgian regulation, and is of no importance in the Dutch regulation. When the separate modules of rules are triggered for a typical situation, the existence of an additional condition in the Belgian regulation will be shown quite clearly.

5. General conclusion

Most of the legal knowledge systems that are operational today, are dealing with the field of situational advice-giving or the application of the law. It is our view that, given an adequate method concerning knowledge modelling and knowledge representation, legal knowledge systems can also be used to support preparation and evaluation of policy. More precisely in the research project as described above, a knowledge system is being built that supports the functional comparison of legislation within the E.C. It is thereby of fundamental importance that the regulations of the various legal systems are described according to an underlying structure of factual elements, common to all legal systems. The elaboration of such a structure is rather difficult as it demands a clear understanding of the various legal systems.

The legal knowledge systems developed for functional comparative law can, in our opinion, rather easily be used for social-economic micro-simulations, since the functions necessary for these simulations show a rather close relationship with the legal interpretation function.

In the next stage of this research project, the methodology will be refined, the regulations of the selected legal systems will be inserted into the knowledge base and representative samples or typologies will be elaborated to be subject of comparison.

6. Notes

1. Promotor of the project is Prof. B. Van Buggenhout; initiator and project leader is F. Robben; K. Debrock and V. Lemmens are responsible for the actual realization of the project.
2. I.C.R.I. stands for "Interdisciplinair Centrum voor Recht en Informatica".
3. The research is carried out in the M.I.J.A.-project (acronym for Model voor de inzet van Informatica bij Juridische Adviesverstrekking), that was initiated at the Institute of Social Law. In view of the recent foundation at the K.U. Leuven of a Centre for Law and Computer Science (I.C.R.I.), the M.I.J.A.-project has been taken in functionally into the activities of this Centre.
4. See for a description on prior research of the M.I.J.A.-project: [Robben 88a], [Robben 90a].
5. See on the use of informatics in the field of social security: [Aarts 89], [Robben 90b].
6. Computer advisory systems were developed in the area of financial benefits for handicapped persons, the minimum level of subsistence and unemployment regulation. For a detailed description of HANDIPAK, the legal advisory system concerning financial benefits for handicapped persons, see [Robben 88b].
7. With descriptive or analytic comparative law is meant the method in which legal rules concerning a particular subject in different legal systems are compared without the purpose to propose a solution for particular problems. The aim is thus purely the supply of information and the amelioration of knowledge.
8. See on knowledge systems in policy preparation: [Aarts 89], p. 79-87.
9. The research is financed by the Research Fund of the K.U. Leuven and is a first part of a more extended research program, that, as far as its motivation is concerned, was elaborated between the M.I.J.A.-research group, the werkgroep Jurimetrie van de Sociale Zekerheid of the Rijksuniversiteit Leiden and the werkgroep Sociale Zekerheidswetenschap of the Katholieke Universiteit Brabant.
10. A good example in this matter concerns the importance of the element of (different or equal) sex in the equal treatment of non-married partners who form an economic unity, and married persons, when determining the amount of the unemployment benefit. In Belgian law the condition of different sexes of the partners is a constituting element, while in other legal systems, e.g. Dutch law, this condition does not exist (see the example mentioned *infra*, in 4.2.2.). If a concept as "living together without being married" is used for the description of the regulations when describing the different legal systems, without further analysis, a similar difference would not be revealed.
11. See on this software program: [Vanthienen 86].
12. This technique and its use in the domain of the development of legal advisory systems have been treated in detail in [Robben 90a].
13. ADS is a product of Aion Corporation and is distributed in the Benelux by Software Generation Benelux.
14. Art. 160 §2 K.B. Arbeidsvoorziening en Werkloosheid, 20.XII.1963 (*B.S.*, 18.I.1964).
15. Adm. Onderrichting R.V.A. (Rijksdienst voor Arbeidsvoorziening), art. 160, nr. 82.3c of 5.III.1982, art. 160, nr. 86.81c of 31.XII.1986, p. 10-11.
16. This because of the difficulties in representing iterative elements and hierarchy of tables in the context of an example.
17. Art. 2, eerste en derde lid; art. 8, eerste en derde lid, Wet 6.XI.1986, *Stb.* 562 ("Toeslagenwet").
18. Art. 1, derde alinea, Wet 6 november 1986, *Stb.* 562.

7. References

- [Aarts 89] Aarts, L. (ed.), *Informatica en Sociale Zekerheid*, 's Gravenhage, VUGA, 1989, 103 p.
- [Cecchini 88] Cecchini, P. (ed.), *1992 Le défi: nouvelles données économiques de l'Europe sans frontières*, Paris, Flammarion/Commission Européenne, 1988, 249 p.
- [Robben 88a] Robben, F., "Description du projet M.I.J.A. - Model voor de inzet van Informatica bij Juridische Adviesverstrekking", in Schauss, M. (ed.), *Systèmes experts et droit*, Précis et travaux de la Faculté de Droit de Namur no. 6, Brussel, Story-Scientia, 1988, 87-96;
- [Robben 88b] Robben, F., "HANDIPAK: legal advisory system concerning financial benefits for handicaped people", in Van Buggenhout, B., Robben, F., Leus, I., Casteleyn, H., Hertecant, G. and Demeester, W., *Het nieuw gehandicaptenrecht. Commentaar bij de nieuwe wetgeving en recente evoluties in het beleid*, Recht en Sociale Hulpverlening, Brugge, Die Keure, 1988, 21-26.
- [Robben 90a] Robben, F., Herbosch, E., Van Buggenhout, B. en Van Bulck, K., "The computer supported development of juridical advice systems based on the decision table technique", in The Foundation for Legal Knowledge Systems, *Legal knowledge based systems. An overview of criteria for validation and practical use*, Lelystad, Koninklijke Vermande, 1990, 50-56.
- [Robben 90b] Robben, F., "De Informatisering van de sociale zekerheid: doelstellingen en gevolgen", in Persyn, C., Pieters, D., Robben, F., Van Bulck, K., *Actuele uitdagingen voor de sociale zekerheid*, II, Brugge, Die Keure, 1990, 1-23.
- [Vanthienen 86] Vanthienen, J., *Automatiseringsaspecten van de specificatie, constructie en manipulatie van beslissingstabellen*, doctoral thesis, Faculty of Applied Economics, 1986, 378 p.

AN INTELLIGENT INTERFACE TO DATA BASES ON ENVIRONMENTAL LAW

Paolo Guidotti, Luciano Lucchesi, Paola Mariani,
Mario Ragona, Daniela Tiscornia

Systems & Management s.p.a.
Vicolo S.Pierino 2, Pisa (Italy)
Istituto per la Documentazione Giuridica del CNR
Via Panciatichi 56/16, Florence (Italy)

Summary

This article features some solutions adopted to develop an interface to data bases on environmental law which adopts string search methods. Emphasis is placed on the structure of the semantic network used for knowledge representation of the domain and an explanation is given of its use.

1. Introduction

Interest in problems related to ecology and, more generally, to the "environment" has been growing in our society in recent years. This issue has also become very important within the panorama of legal science: there have been numerous legal studies on the environment and it is now possible to talk about "environmental law". Although those involved in this field are interested in being able to have rapid and easy access to environmental law documentation, the situation regarding information retrieval systems in this sector in Italy cannot be considered satisfactory. Whilst, in fact, many data banks exist with an enormous overall total number of documents on-line, such as those for example in the data banks of Italgire system, environmental law information is currently spread throughout many different data banks and is almost always organized under the classifications found in standard legal sources (national legislation, regional legislation, civil "massime" (summary of the principles in a case) of the Italian Corte di Cassazione, criminal "massime" of the Cassazione, etc.), held at different documentation centres (Italgire, Celex, IUCN, etc.).

The difficulty is not, therefore, related to the lack of information but rather to being able to have easy access to it. The user who wishes to obtain information by on-line searching faces the following problems:

- the proliferation of data banks,
- the structural differences in the individual data banks (full text or only reference data banks),
- the absence of standardization in link-up procedures,
- the diversity of query languages,
- the multiplicity of conceptual retrieval tools (thesauri, classifications, etc.).

The purpose of our research, supported by Environment Committee of Italian CNR, is to build a system which facilitates user/computer interaction during the search of environmental data banks, thereby overcoming the problems discussed above. It should, therefore, give the user the possibility of searching the documents stored in different data banks in a logically uniform way and it should aid him in formulating his query and search strategy.

Although there are various expert systems implemented for I.R. described in recent

publications [Anderson 76], [Bennet 79], [Marcus 81], [Yip 81], [Meadow 82], [Pollit 86], [Lebowitz 83], [Vickery 83], [Tong 84], [Ceri 88], [Zarri 88], [De Jaco 88], there are no such applications in the legal field. In this field many papers [Hafner 81], [Hafner 87], [Cross 87], [Bing 87] stress that conceptual retrieval techniques must be used in the legal area if an efficacious selection of documents, by using their semantic content, is to be made. It goes without saying that the documents must be interpreted if they are to be given a semantic representation which can be employed by these search techniques. In our case, as it is not feasible to modify the structure or the classification of the documents, which are already largely stored in existing data banks, we are forced to use the string search method found in on-line systems.

We will illustrate the main features of the system in this paper and will, therefore, pay special attention to the A.I. technology utilized for permitting the user to search for documents according to their domain, while retaining the string search structure under which they are stored.

2. Functionality and Structure of the System

In order to define the required functions of the system, it is useful to analyze the task carried out by a human intermediary who helps the user satisfy his information needs. In brief, he carries out the following tasks:

- a) he understands the user's request: he talks to the user until he eliminates any ambiguity in the request;
- b) he formulates and implements the query: he translates the user's request into the formal data base query language and defines a search strategy;
- c) he analyses the results: he shows the user the retrieved information and points out which parts of the query are still unanswered. If need be, he checks why certain answers are wrong.

The proposed system, called CABALA (Consultazione Assistita di Basi Dati di Leggi Ambientali) has four main features which enable it to perform some parts of the previously listed tasks:

- navigation on a semantic network,
- management of the dialogue with the user,
- construction of the query,
- definition of the search strategy.

Its main aim is to display the conceptual structure of the data base described by a semantic network, and to assist and guide the user in navigating it so that he can formulate his queries properly. The query is then translated into a suitable form for the specific information retrieval system involved.

CABALA can be divided into three main logically independent parts which communicate amongst themselves only by exchanging messages:

- Query Generator,
- Data Base Query Manager,
- Data Bases.

In this paper, we are only going to take into consideration the Query Generator, currently the only part under development.

The objective of the Query Generator is to enable the user, regardless of his experience in the domain, to easily formulate a valid query from both the legal and common sense points of view. Once the user has formulated his query, this is translated into an intermediate language and both its correctness and efficacy are evaluated. The Query Generator is, therefore, subdivided into three modules:

- the Navigator

- the Query Constructor
- the Query Evaluator.

The three modules act simultaneously, supplying, during any step, the query equivalent to the specifications supplied by the user and its evaluation. This enables the user to immediately check the work done during the search session. This operational procedure is achieved through a blackboard-like architecture, as can be seen in Fig. 1.

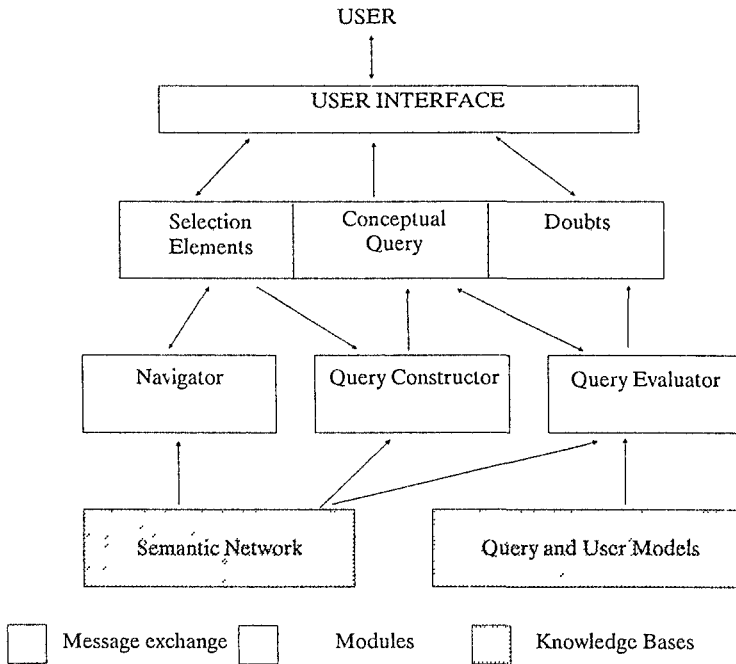


Fig. 1

3. Domain Knowledge Formalization

The basic criterion used for the choice of the domain knowledge formalization is the facility to codify due the width of the field taken in consideration. In our system deep models are not necessary as for the advisory systems in fact more complex formalisms such as production rules have been rejected, and a semantic network as the representation framework has been adopted. In practice a thesaurus has been created by extracting a series of significant terms from legal texts, representing the nodes in the network. The choice has also been suggested by the fact that another equipe at IDG was already working on the classification of environmental law. Anyhow the relations, which we use, are semantically enriched whereby we prefer to consider it a semantic network. The first relation defined between the terms is the Broader Term relation (BT); which is transitive. The BT relation lays down hierarchies (trees) among terms, each belonging to a specific argument that arises in the definition of the hypothetical case or defines legal aspects pertaining to the domain. In the field of noise pollution, the subdomain chosen for building the

prototype, the following BT hierarchies were identified: Sources of emission, Type of noise, Ambits for safeguarding, Prevention and control, Bodies, Normative sources, Sanctions, Pertinent facts. In fact, these hierarchies constitute a classification of the conceptual domain stored in the databases below mentioned.

It must be noted that the BT hierarchies point out the basic concepts that make up the query. Then the hierarchies, representing the nodes of a graph, may be linked by oriented arcs, identifying the relations existing between them. Fig. 2 illustrates the hierarchies graph (HG graph) related to noise pollution. Through the HG graph it is possible to capture the semantic of the user's queries. Let us suppose that the user selects terms in more than one hierarchy: the path on the HG graph that links the hierarchies univocally identify the query. For example, on the basis of the terms used and on the relations between hierarchies, it is easy to capture the meaning of the following three questions:

- Which judicial bodies are competent to control and prevent noise pollution?
- Does the judge have the power to verify the tolerance level of the noise?
- Before which bodies must workers lay their claim to be protected against noise pollution in the working place?

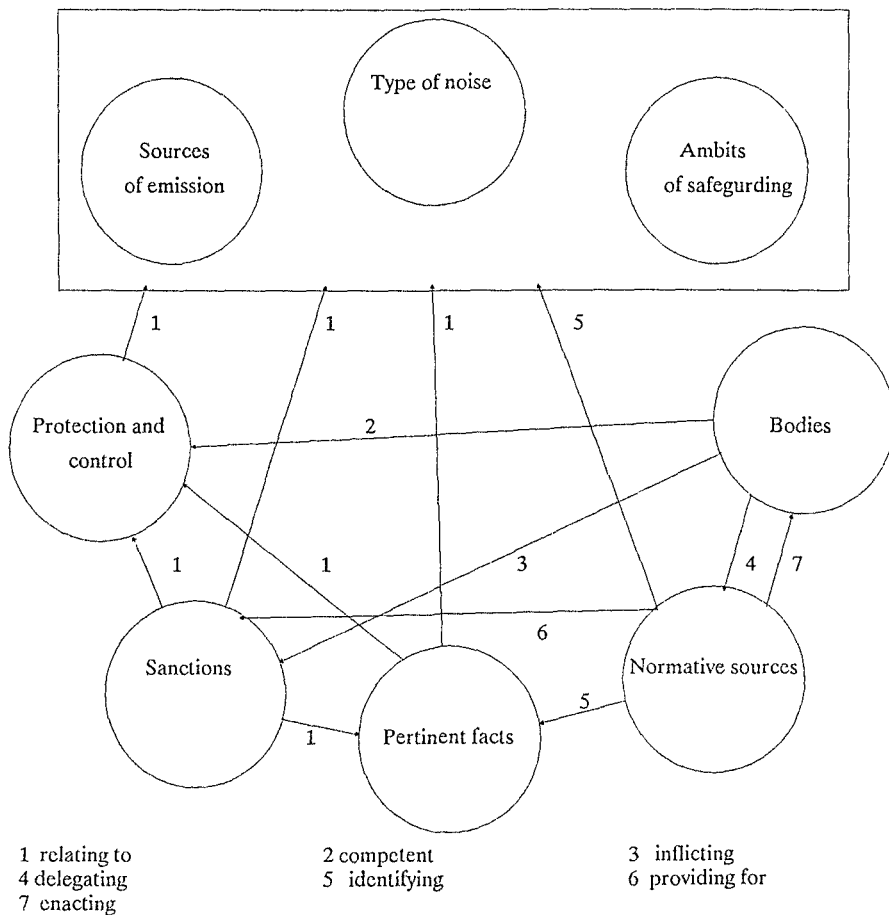


Fig. 2

We use another class of relations to codify knowledge resulting from a partial interpretation of the norms and from the case law found in the data bases. We call a relation of this class a Related Term (RT) because it is used by the Query Constructor as it was symmetric and transitive, but all RT relations between terms of two hierarchies take the name and the direction of the relation between the hierarchies on the HG graph. An example of this type of knowledge is:

In the case of noise pollution, only fines and arrests can be inflicted as sanctions by the judge.

Moreover, this relation enables the representation of general knowledge on law and legal authority of the type:

*the ordinary courts inflict criminal sanctions,
the administrative bodies inflict administrative sanctions.*

These examples underline two meanings of the RT relation. The former is only valid according to the context of the case in question: in the case of noise pollution the sanction is inflicted by the judge. In the present state of our research, the context is represented by the type of pollution. Let us, therefore, consider RT relations as being divided into two groups:

- primaries: validity is independent of the context,
- secondaries: validity depends on the context.

As an example of the knowledge which can be extracted and formalized in this way, see Fig. 3 in which some parts of the hierarchies relating to the Bodies and Sanctions involved are represented according to the RT relations existing between them. It is demonstrated in the Figure that the administrative courts only inflict administrative sanctions whereas ordinary courts only inflict criminal sanctions while, among the Executive bodies, it is the Mayor who inflicts administrative sanctions. Lastly, in the case of noise pollution, the judge can only fine offenders or have them arrested.

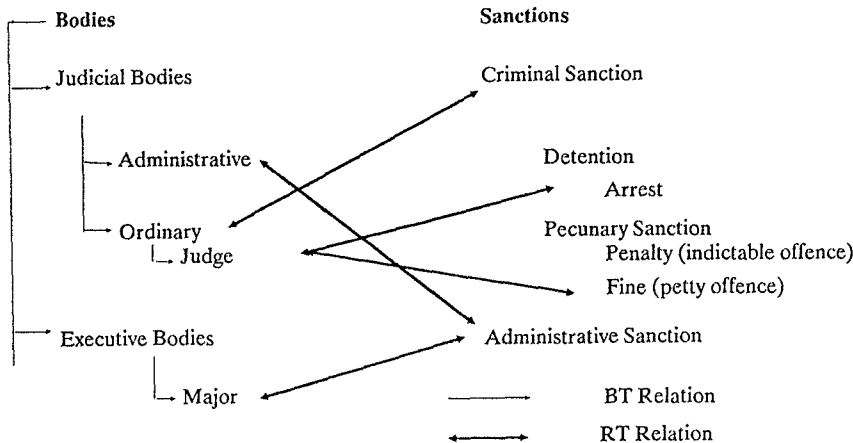


Fig. 3

4. Use of the Semantic Network

The semantic network is utilized by both the Navigator and the Query Constructor. The Navigator allows the conceptual structure of the documents stored in the data base to be visited: the user specifies his query from the menu in which he selects some pre-defined terms (items). The menus are derived from the hierarchies defined by the BT relation, which means that specifying the query is the same as selecting nodes in those hierarchies. In this way, the set of hierarchical nodes performs the role of the system's vocabulary. This considerably simplifies the user-system interaction: the system, in fact, never has to analyze an unknown term as may occur by freely interacting in natural language. The method which is adopted, however, leaves it up to the user to express his query in the terms of the system's vocabulary. This is counterbalanced by the fact that being hierarchically organized terms, the user is offered a classification table of the domain which acts as a map of the conceptual structure of the data base. We left the user the possibility to select the hierarchies without any predefined order informing him, however, about the hierarchies correlated to the selected one.

The RT relation is utilized by the Navigator for displaying the related terms and for guiding the user in formulating his query. It should, also, be noted that additional information is given by the absence of a RT relation between two selected terms. In fact, if two terms belong to hierarchies having RT relations but they are not linked by a RT relation or do not belong to sub-trees whose roots are linked by RT arcs, an error in the selection can be assumed and the user can, therefore, be informed about it. This is the case, for example, where the user selects "mayor" and "criminal sanctions", see Fig. 3.

In the Evaluator implementation, we thought that it was more suitable to use simple heuristic criteria like those generally followed in developing the system, which can be set up in an easier way than statistical models. The latter are more reliable but more complex and need a large amount of data. The Evaluator, as said above, has the task of verifying the validity and the efficiency of the query. Therefore, the first thing that the Evaluator has to do is to verify whether it is possible to find a path linking the selected hierarchies on the HG graph, if not the Evaluator suggests how to complete the path. Anyhow, it is always possible to force the system to process the query.

It's more difficult to evaluate the query's possibility of success with our model. Particularly, we think it is very difficult to evaluate query precision, a standard which measures the intermediary's capacity to reject non pertinent documents. Anyhow it is easier to evaluate query recall, a standard used to measure the intermediary's capacity to retrieve all documents pertinent to the query. For example, the documentation on noise pollution is poor and fragmented, therefore very specific queries may not have any answer.

We can use the deepness of the selected terms from the hierarchy's roots as a measure of the specificity of the query. Since the various types of normative action and the bodies competent to carry them out are common to all sector of environmental law, the most significant elements seem to be those related to areas where regulations are applied, i.e. Sources of Emission, Types of Noise, Ambits for Safeguarding. Therefore it is natural to allow more specificity in the selection of terms for these hierarchies than for the others. To accomplish this, we assign a weight to each hierarchy, which multiplied by the deepness of its selected terms, gives a number that we use to find a pounded measure of the specificity. The sum of the numbers calculated for each selected hierarchy must be lower than a fixed number.

The Query Constructor accepts the selection about the hierarchies made by the user and transforms it into a query for the textual data base. Let us analyze a series of examples of the method for constructing a query of the following kind:

give me all the documents which contain <expr>,
 where <expr> represents the set of terms generated by the Constructor starting off from the selection made by the user.

Let us assume that we have one hypothetical textual data base in which the query language represents the system's intermediary language.

In order to understand the use of the BT relation, let us consider the case where only one node in the hierarchy has been selected. There are two alternatives: either the node is a leaf or the node has children. In the former case, the node represents itself and is picked up immediately. In the latter case, let us look at the part of the hierarchy illustrated in Fig. 4.

Let us assume the user has selected "executive Bodies". As "Minister", "President of Regional Council" and "Mayor" have executive power, the search is correctly expanded by including these terms. We will, therefore, use all the sub-tree which has the node selected as its root in the query construction. Let us call N the set of nodes we obtain by visiting the tree of the root n.

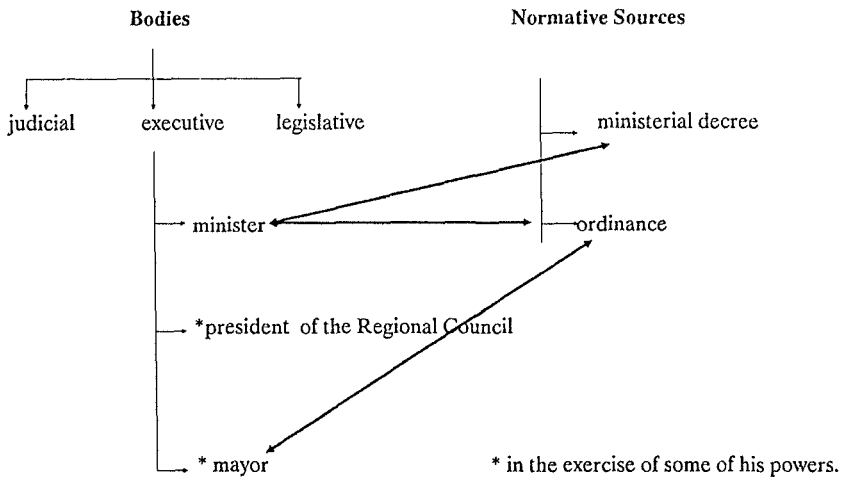


Fig. 4

If the user selects the nodes n_1 and n_2 , with the same sub-trees N_1 and N_2 , in the same hierarchy, the set M of terms used for constructing the query is reached by uniting N_1 and N_2 . The elements of the set M, linked by the logical operator *or*, form a term of <expr> relating to the activated hierarchy; it is equivalent to the following query:

give me all the documents which contain one or more elements of M.

The terms generated by the sets $M_1, \dots, M_n$, belonging to different hierarchies, are linked through the logical operator *and*; that is equivalent to the following sentence:

*give me all the documents which contain
one or more elements of M_1*

.....

one or more elements of M_n

The RT relation is, instead, utilized by the system for completing and making the query formulated by the user more exact. In order to understand it, let us look at the parts of the hierarchies illustrated in Fig. 4, relating to the Bodies involved and to Normative sources. Let us assume that the user has selected the node "minister" in the hierarchy relating to Bodies, which we can, therefore, say has been activated. The information given by the RT arcs is that the minister issues ordinances and ministerial decrees. In order to broaden the system's retrieval capacity, we can include the terms "ordinance" or "ministerial decree" in the query; this improves the system's recall. The risk is, however, to retrieve a large number of non pertinent documents, for example, those relative to mayoral ordinances: it is, therefore, necessary to exclude documents in which the terms "ordinance" and "mayor" appear; this increases the system's precision.

Finally, let us consider the case where a RT arc links a selected term to another term belonging to an activated hierarchy. In this case we cannot utilize the latter term because we distort the user's intention. Let us, in fact, assume that the user has selected the terms "minister" and "ordinance" in the two hierarchies represented in Fig. 4 or, in other words, that he has requested documents where both "minister" and "ordinance" appear. It is, obviously, wrong to add the term "ministerial decree" to the query, in this case. Therefore, the more hierarchies activated by the user the less the system will use the RT arcs, which will be completely unutilized, in the query construction, if the user has selected a node for every hierarchy.

4.1. Algorithm of the Query Constructor

We can define the algorithm for the Query Constructor in a final summary which utilizes both types of knowledge simultaneously. We can see how it functions in a final example illustrated in Fig. 5.

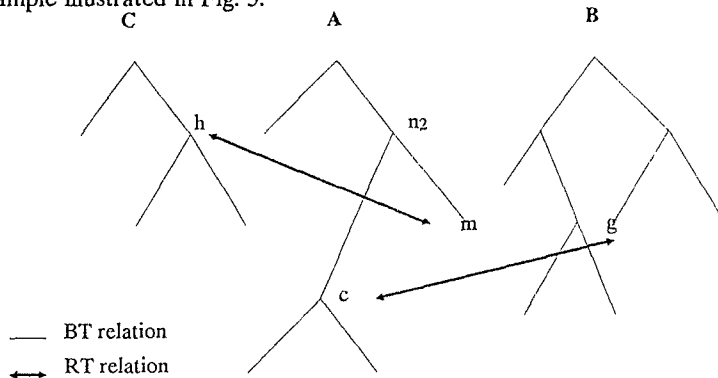


Fig. 5

Let us assume that the user has selected node n_2 in hierarchy A from which RT arcs emerge. As we have already seen, we must, nevertheless, consider all the sub-trees of the root n_2 in A. The descendants of n_2 in A are linked through the RT arcs to elements of the C and B hierarchies in which the user has not selected any node. Given that the descendants of n in A are included in the query, then the terms in C

and B linked by the RT relation must also be included. Firstly, let us formalize the construction of the term related to an activated hierarchy.

Let $n_1 \dots, n_m$ be the nodes selected in the k-th BT hierarchy and $N_1 \dots, N_m$ be their respective sub-trees. We presume:

$$M_k = \bigcup_{i=1}^m N_i$$

Let $[f_1, \dots, f_n]$ be terms, belonging to non activated hierarchies, obtained from the nodes in M_k following the RT arcs and let $[b_{i1}, \dots, b_{ih}]$ be terms related through the RT relation to f_i so that every $b_{ij} \notin M$. Let us indicate by F_i the sub-tree of f_i and by B_{ij} the sub-tree of b_{ij} so that $B_{ij} \cap M_k = \emptyset$. Let us assume $B_i = B_{i1} \cup B_{i2} \dots \cup B_{ih}$ and let us construct the term B_i of $\langle expr \rangle$ by linking the elements of the set through the *or* operator. We will recursively construct the term:

$$A_k = F_1 \text{ and-not } B_1 \text{ or } \dots \text{ or } F_n \text{ and-not } B_n.$$

where F_i is obtained by applying the procedure we have just seen for M_k to each F_i and setting activated the hierarchy of F_i . Let be

$$T_k = M_k \text{ or } A_k,$$

where M_k is the term obtained by linking the elements of M_k through the *or* operator.

For each M_k we will construct the related term T_k , as explained. The query will be:

give me all the documents which contain $\langle T_1 \text{ and } T_2 \text{ and } \dots \text{ and } T_z \rangle$.

The conceptual Query Constructor's algorithm is, therefore, completely specified.

5. Conclusion

One of the most important aspects in defining a knowledge-based interface to retrieval systems is to decide the level of "deepness" at which knowledge must be represented. This means identifying all the concepts for ensuring the completeness of the retrieval which are necessary and sufficient in order to avoid transforming the system into a decision-support system the semantic network has, in this sense, the advantage of providing a great deal of elasticity in defining the deepness level. The functions of the evaluator can undergo further developments. Particularly we think that the comprehension of the query by the evaluator can furtherly? be used allowing a dialogue with the user. Such situation till now is not provided. We also plan to use the HG graph in the Query Constructor in order to improve the query's precision [Brajnik 87], [Rabitti 88]. At present, a CABALA prototype, including the functions of the Query Generator described here, is being implemented. The tool used to implement it is Epsilon, an expert system shell based on Prolog, which enables a partition of the knowledge into various modules, each with its own deduction mechanism [Coscia 86], [Levi 87], [Coscia 88].

6. References

- [Anderson 76] Anderson R.H., Gillogly J.J., *Rand Intelligent Terminal Agent (RITA): Design Philosophy*, Rand Report n. R - 1809 ARPA 1976.
- [Bennet 79] Bennet J.S., Englemore R.S., *SACON: A Knowledge-based Consultant for Structural Analysis*, in: Proceedings of 6th International Joint Conference on Artificial Intelligence, Tokyo - Japan, pp. 47-49, 1979.
- [Bing 87] Bing J., *Designing Text Retrieval Systems for 'Conceptual Searching'*, Proceedings of the 1st International Conference on Artificial Intelligence and Law, Boston, May 1987, pp. 43-51.
- [Brajnik 87] Brajnik G., Guida G., Tasso C., User Modeling in Intelligent Information Retrieval, in: *Information Processing & Management*, Vol. 23, 1987, No. 4, pp. 305-320.
- [Ceri 88] Ceri S., Gottlob G., Tanca L., *Logic Programming and Databases*, Atti della Giornata di Studio "Evoluzioni dei sistemi per basi di dati e di conoscenza", AICA, Milano, 1988, pp. 3-24.
- [Coscia 86] Coscia P., Djennaoui S., Franceschi P., Kouloumdjian J., Levi G., Lei L., Moll G.-H., de San Victor I., Sardu G., Simonelli C., Torre L., *The Epsilon Knowledge Base Management System: Architecture and Data Base Access Optimization*, Workshop on Logic Programming and Data Bases, Venice, December, 1986.
- [Coscia 88] Coscia P., Franceschi P., Levi G., Sardu G., Torre L., *Meta-level Definition and Compilation of Inference Engines in the EPSILON Logic Programming Environment*, Fifth International Conference and Symposium on Logic Programming, Seattle, August, 1988.
- [Cross 87] Cross G.R., de Bessonnet C.G., *The Implementation of CCLISP*, presented at the 2nd International Conference on Law and Technology, University of Houston Law Center, June 1985, Proceedings to appear in Computing Power and Legal Language, C. Walter, Ed., Greenwood/Quorum Press, 1987.
- [De Jaco 88] De Jaco D., Garbolino G., Pogliano P., *Un sistema di consultazione in materia di diritto ambientale basato su tecniche di intelligenza artificiale: considerazioni preliminari*, Proceedings of the IV International conference "Law and Computers", Rome, 16-21 May 1988, sess. X, n. 23, 16 pp.
- [Hafner 81] Hafner C.D., *An Information Retrieval System Based on a Computer Model of Legal Knowledge*, Ann Arbor, UMI Research Press, 1981.
- [Hafner 87] Hafner C.D., *Conceptual Organization of Case Law Knowledge Bases*, Proceedings of 1st International Conference on Artificial Intelligence and Law, Boston, May 1987, pp. 35-42.

- [Lebowitz 83] Lebowitz M., *RESEARCHER: An Overview*, Proceedings of the Third National Conference on Artificial Intelligence, Washington, DC, 1983, pp. 232-235.
- [Levi 87] Levi G., Modesti M., Kouloumdjian J., *Status and Evolution of the Epsilon System*, in *EPRIT'87*, pp. 593-610, North-Holland, 1987.
- [Marcus 81] Marcus R.S., Reintjes J.F., *A Translating Computer Interface for End-user Operation of Heterogeneous Retrieval Systems: Design*, Journal of the American Society for Information Science 32, 4, pp. 287-303, 1981.
- [Meadow 82] Meadow C.T., Hewett T.T. and Avesa E.S., *A Computer Intermediary for Interactive Database Searching: Design*, Journal of the American Society for Information Science 32, 5, pp. 323-332, 1982
- [Pollit 86] Pollit A.S., *An Expert System Approach to Document Retrieval*, Ph.D. Thesis, Department of Computer Studies and Mathematics, Huddersfield Polytechnic, 1986.
- [Rabitti 86] Rabitti F. (ed.), *Research and Development in Information Retrieval*, Proceedings of the ACM Conference, Pisa, 1986, 249 pp.
- [Tong 84] Tong R., Askman V. and Cunningham L., *RUBRIC: An Artificial Intelligent Approach to Information Retrieval*, Proceeding 1st International Workshop on Expert Database Systems, 1984.
- [Vickery 83] Vickery A., *An Intelligent Interface for Online Interaction*, Journal of Information 9, pp. 7-18.
- [Yip 81] Yip M.K., *An Expert System for Document Retrieval*, M. Sc. Thesis, Massachusetts Institute of Technology, 1981.
- [Zarri 88] Zarri G.P., *Conceptual Representation for Knowledge Bases and 'Intelligent' Information Retrieval Systems*, Proceedings of the Eleventh International ACM Conference on Research and Development in Information Retrieval, Presses Universitaires (PUG), Grenoble, 1988.

AN ALTERNATIVE FOR DEONTIC LOGIC

Jaap Hage
Rijksuniversiteit Limburg
Vakgroep Informatica
P.O. Box 616
6200 MD Maastricht
The Netherlands

Summary

In this paper it is argued that extensions of traditional propositional or predicate logic are not adequate for representing legal rules. This tenet is illustrated by examples from deontic logic. As an alternative, the (informal) outset of a system of reason based reasoning is presented. By means of examples, evidence is presented that reason based reasoning avoids the problems connected to the formalization of rules and that it fits in with actual legal reasoning.

1. Introduction

The research in the area of Legal Knowledge Based Systems (LKBSs) goes hand in hand with theoretical investigations concerning types of logic needed for these systems. Examples of research on non-monotonous, epistemic, and deontic logics are [Gärdenfors 89]; [Prakken 89]; [Alchourrón & Bulygin 89]; [Castaneda 89]; [Bench-Capon 89]; [Alchourrón & Martino 89].

Much research that strives for a formalized theory tries to build upon 'traditional' propositional or predicate logic (PL).¹ In this paper firstly I will try to show why this approach is wrong where rules are concerned, by pointing out some classical problems attached to deontic logic, and by giving theoretical reasons why the whole 'traditional' approach leads to problems if applied to rules. Deontic logic figures in this paper as an extension of PL. The problems discussed are not caused by the deontic character of the logic, but because of the attempt to represent rules by means of (an extension of) PL.

Secondly, as an alternative, I will present the beginnings of a theory of reason based (legal) reasoning (RBR). My expectation is that LKBSs using this type of reasoning will fit in more with traditional legal reasoning than most present day systems. Moreover, systems using RBR will avoid the problems connected to the representation of rules and have a better capability for handling special legal reasoning techniques.

2. Some problems of deontic logic

When people reason about 'may' and 'ought', they reason in a fashion which is not very different from their normal reasoning. Even more, their reasoning is usually mixed, that is, people do not separate their alethic (factual) and their deontic (normative) reasoning. Therefore, a system of deontic logic (DL) must also allow for all the arguments which are valid in alethic logic. In other words, an adequate system of DL should include an adequate system of alethic logic.

Reasoning along these lines leads to the attempt to create a DL which is an extension of PL. Such an extension is usually achieved by adding deontic operators

or predicates to the syntactical apparatus of PL and devising a number of definitions, axioms and rules of derivation (and often informal rules of interpretation) which, if added to PL, hopefully lead to intuitively acceptable results. The last goal is seldomly accomplished. This is not the place to enumerate all the paradoxes and counter intuitive results of different systems of DL. Instead I will mention a few difficulties, which are at the heart of the issue I want to discuss. The difficulties will be demonstrated by means of some example sentences:

- | | |
|-------|--|
| A. | It is forbidden to steal |
| B. | It is not forbidden to steal |
| I. | $F(\text{Stealing})$ |
| II. | $F(x)(\text{Stealing}(x))$ |
| III. | $(x)(\text{Stealing}(x) \rightarrow Fx)$ |
| IV. | $(\exists x)(\text{Stealing}(x) \ \& \ \sim Fx)$ |
| V. | $(x)(\text{Stealing}(x) \rightarrow \sim Fx)$ |
| VI. | $(x)(\text{Stealing}(x) \rightarrow \text{Burglary}(x))$ |
| VII. | $(x)(\text{Burglary}(x) \rightarrow Fx)$ |
| VIII. | $(\exists x)\text{Stealing}(x)$ |
| IX. | $N(x)(\text{Stealing}(x) \rightarrow Fx)$ |

Let us assume that F is a deontic operator or predicate (depending on the system of DL) and that it stands for 'Forbidden'. We will look at a number of different formalizations of A. One formalization is I, where Stealing is the action *type* stealing. Clearly F is a *predicate* which works on action types. But if F works on types, how can we express that a certain action token is forbidden?²

We can also try to formalize A by making use of a *universal quantifier*. One style of formalization is II, 'For all x (we quantify over actions) it is forbidden that they are of the type Stealing'. F functions as an *operator* working on *whole sentences*.³ But how can we, following on this interpretation, say of individual actions that they are forbidden? If F is an operator and not a predicate, it does not work on individual constants, nor on definite descriptions.

Another style of formalization is III, 'For each x holds that if x is of the type Stealing, x is forbidden'. Now F is a *predicate* working on action *tokens*.

This approach leads to problems in the case of B, the negation of A. The negation of III is IV, saying that there is at least one case of stealing which is not forbidden.

As a formalization of B, IV is unacceptable for a number of reasons. Firstly, a more intuitive formalization would be V, saying that 'No act of stealing is forbidden'. But, on its turn, this is too strong. It may be the case that all actual cases of stealing are also cases of burglary, a situation formalized by VI. And it may also be the case that burglary is forbidden, although stealing in itself is allowed. Then VII would also be true. From VI and VII we can derive III, which is only consistent with V on the counter intuitive assumption that there are no cases of Stealing.

But the idea that IV would be a correct formalization of B is also in itself strange. Because from IV we can derive VIII implying that from the fact that stealing is allowed, it follows that there is at least one case of stealing.

This problem can again be handled by assuming that the quantifier does not have actual but possible actions as its range.⁴ But, firstly, given that interpretation A should normally be formalized as IX. The quantification over possible actions (or actions in possible worlds) is expressed by means of a modal operator.⁵ Secondly, even this interpretation leads us to the strange conclusion that actions which are allowed are also possible. Because VIII should then be interpreted as saying that there is a possible action of stealing.

3. The cause of the problems

What makes it so difficult to formalize a simple rule as the one that forbids stealing? In my opinion, the basic mistake is that rules are treated as assertive sentences. They are taken as saying something about the world, while they do not say anything about the world at all.⁶

I do not want to make the traditional distinction between *is* and *ought* here.⁷ My point is also applicable to rules of language, and rules of derivation. All rules should be strictly distinguished from universal statements. Rules cannot be negated. They cannot be used for reasoning backwards (Modus Tollens). And rules cannot contradict each other, although they may be incoherent.

Instead, rules are some kinds of instruments. They are used to make facts into reasons. Rules of language make characteristics of an object reasons for classifying that object in a specific manner. Deontic rules make facts into reasons for behaviour. They do not describe that certain things ought or ought not to be done. They *make it the case* that things ought or ought not to be done. *Rules are constitutive* [Hage 87].

Let me illustrate this by means of an example. In a small country there is a rule of law, stating that all burglars ought to be punished. There is no law forbidding or threatening sanctions for theft. It happens, however, that all thieves are also burglars. In that country all thieves ought to be punished, not because they are thieves, but because they are burglars. Therefore, the *statement* 'All thieves ought to be punished' is true in that country.

But the *rule* with the same formulation is not valid. If the rule is formalized in the same manner as the corresponding statement, which tends to be the case in systems of DL which build upon PL, problems result.

In my opinion, this type of problem disappears if we work with a logic of reasons. In such a logic, rules do not figure explicitly in the reasoning. Their influence is only traceable through the types of reasons which are adduced. This takes away the temptation to treat them like statements. Rules (or better: warrants; cf. section 4.6) which are instruments, belong to quite a different logical category than reasons which are facts, and should be treated differently in the process of reasoning.⁸

Apart from the fact that RBR avoids the problems attached to the representation of rules, there is an important independent reason for adopting a form of RBR, namely that people usually reason by presenting arguments for conclusions instead of inferring conclusions from a set of premisses that entail them. RBR fits in better with everyday and especially legal reasoning than PL [Toulmin 58].

In the next section, I will provide the prolegomena for such a logic of reasons. By means of examples I will show how the constructs of this logic reflect particularities of legal discourse. As a consequence, the prolegomena can also be read as *a conceptual analysis of legal discourse*.

4. Prolegomena for a logic of reasons

4.1. Facts

Reasons are (groups of) facts with impact on other facts. 'Facts' should be understood to mean the denotatum of anything which can grammatically (and maybe semantically) correctly and truly be stated after 'It is a fact that ...' or 'It is the case that ...' [Strawson 71, p. 196; Hage 87, p. 13]. In this sense of 'fact', the following sentences, if true, express facts:

- It is raining.
- If John has killed Mary, he is a murderer.
- It is forbidden to steal.⁹
- Clarissa will certainly come.
- This elephant cannot fly.
- Six is necessarily two times three.

It should be noted that this definition of facts is more comprehensive than some positivist definitions which would exclude at least the third, and possibly all except the first example.

4.2. Reasons

If a fact, or a combination of facts, F1 pleads for or against some other fact F2, F1 is a reason for or against F2. For instance, the facts that something has a seat for one person and more than one leg together are a reason for the fact that this object is a chair. Or the fact that John has killed Mary is a reason for the facts that John did something forbidden and that John is punishable.

Reasons for facts can often ipso facto be considered reasons for (mental) behaviour. The facts that something has a seat for one person and more than one leg also are a reason for *considering* this object a chair. And the fact that John has killed Mary is a reason to *assume* that John did something forbidden and that John is punishable. Furthermore, the fact that John's projected act would kill Mary is a reason for *abstaining* from that act. The results of a number of experiments can be a reason for *adopting* a law of nature and statistical data can constitute a reason for *using* a rule of inference.

4.3. The weight of reasons

Reasons are often assumed to have some *weight* or *force*. They plead for or against other facts with a certain force [Raz 75, p. 25 ff.; Dworkin 78, p. 24 ff.] The exact nature of that weight or force is unclear, for although people speak of weighing reasons for and against, the precise nature of this weighing is never elucidated [Hage 87, p. 218 ff.]¹⁰

Despite these doubts, it seems a good starting point for a logic of reasons to assume that reasons have a certain weight which can be expressed numerically. The weighing of reasons can then be represented by means of some arithmetical operation upon these weights.¹¹

4.4. Categories of reasons

On the ground of their conclusions¹² reasons can be subdivided into a number of categories. Facts belong to either one of the following four categories:

- *unqualified facts*; (e.g. John is a murderer);
- *anancastic facts*; (e.g. John cannot murder);
- *deontic facts*; (e.g. John must not murder);
- *epistemic facts*; (e.g. John has probably not murdered).

These different types of conclusions ask for different types of reasons. Unqualified facts ask for conceptual reasons, facts which make it the case that John is a murderer, or, more strictly, facts which make the concept 'murderer' applicable to John, such as the fact that John has killed Mary after premeditation.

Anancastic conclusions ask for facts which make the conclusion (un)necessary or (im)possible. For instance the fact that John is paralyzed makes it impossible for him to murder.

Deontic conclusions ask for reasons which make that something is or is not ordered, permitted or forbidden. The fact that there is a legal prohibition of murdering is a reason why John must not murder.

Epistemic facts, finally, ask for reasons which make the conclusion (un)certain or (im)probable. For instance, the fact that John has an alibi makes it improbable that he has murdered Mary.

4.5. Points of view

Often reasons and their conclusion are qualified by a point of view. For instance, the fact that committing euthanasia is a case of willfully depriving a person from his life is a reason why (in general) euthanasia is *legally* forbidden (section 287, Code of Dutch Penal Law). Both the reason and its conclusion are qualified as legal.

Points of view are orthogonal to categories of reasons. For instance, it is legally forbidden to murder (deontic). The fact that John has willfully committed euthanasia on Mary makes him legally a murderer (conceptual). The municipal council cannot legally ratify international treaties (anancastic). And the fact that there is a notarial act testifying to it makes it legally certain that a conveyance has been entered into (epistemic).

The fact that reasons belong to a point of view entails that those reasons are only valid within that point of view. Legal reasons as such have no force outside the law. Take for instance the case of euthanasia. And it is the other way round too: Extra-legal reasons have no value within the law. (This does of course not exclude the possibility that the law incorporates extra-legal reasons and makes them in that way also legal reasons. Cf. also section 4.12.)

4.6. Warrants

Reasons can be universalized. If F1 is a reason for F2, facts like F1 are reasons for facts like F2 [Hare 63, chapter II]. Beyond the use of reasons there is a general rule or principle. After Toulmin [Toulmin 58, p. 97 ff.], I will call such rules 'warrants'. Using the warrant from F1 to F2 and considering facts like F1 to be reasons for facts like F2 are by definition one and the same thing.¹³

Just like reasons, warrants belong to categories and may belong to points of view. So there are deontic warrants (rules of behaviour), epistemic warrants (rules of inference), conceptual warrants (rules of language) and anancastic warrants (rules of logic, laws of nature). This entails that the notion of a warrant is more comprehensive than that of a rule of inference. Rules of inference (just like the notion of uncertainty) belong to one specific category, namely the epistemic [Hage 88].

Deontic rules can belong to the law, to morals, to etiquette or be prudential. Rules of inference can be commonplace, but also statistical. Rules of language (conceptual rules) can belong to every day natural language, but also to some specialized professional language like that of nuclear physics or the law. Rules belonging to different points of view can supplement each other, but they may also lead to conflicting results in special cases. Biologically, a whale is not a fish, although it is one according to everyday use of language.

4.7. The scope of warrants

Instruments should be used in some places and not be used in other places. This is also the case with the instruments called 'warrants'. For instance, the logical warrant allowing substitution of coreferential identifiers is not applicable in intensional contexts [Linsky 71 and 77]. The rule of inference that if Ellen's car is not before her home at noon, Ellen may be assumed to be at her work, does not hold during Ellen's holiday. And the rule that a debtor should comply with his contractual obligations does not hold in the case of force majeure. Warrants have a *scope* determining when they can and when they cannot be used.

One may be tempted to consider the scope boundaries of warrants as negative conditions. For some (logical) purposes this may do, that is, lead to acceptable results. Nevertheless, there is an important difference. The conditions of a warrant, the facts which are made reasons by a warrant, are directly relevant for the conclusion. The scope boundaries by themselves are not relevant for the conclusion. They are only relevant insofar as they are important for the validity of the warrant use.

Thus, the fact that Ellen is not on holiday is *not* part of the reason why we may assume that she is on her work. It is only a background condition which makes the use of the rule of inference valid. And the fact that there is no force majeure is *not* part of the reason why the debtor must pay hundred guilders. It is his contract which is the reason behind his obligation. The absence of force majeure is only a background condition which makes the reason a valid one.

4.8. The backing of warrants

Using warrants is a form of behaviour for which (deontic) reasons can be given. In the terminology of Toulmin, providing such reasons is giving the warrants a *backing* [58, p. 103 ff.] The *types* of reasons one should give to provide warrants with a backing depends on the *domain* (or point of view; cf. section 4.5) of the warrant. Rules of inference can be backed with statistical data. Rules of law can be backed by references to sources of law. Physical laws are backed by empirical data (if we may believe positivists).

It should be noted that the reasons used for backing a warrant can also be generalized. The results are again warrants. These warrants will have a meta-character with respect to the warrants which were originally backed. Thus, the relation between empirical data and laws of nature rests on methodological warrants of physics. And the relation between sources of law and rules of law rests on rules of recognition [Hart 61]. In the case of the law, this type of warrants contains *legal meta-knowledge* [Hage 90].

4.9. Meta-reasons

There are also reasons whose workings concern other reasons. These may be called *meta-reasons*. There are meta-reasons which *influence the weight* of other reasons. For instance, a thief must give back what he has stolen. But the reason why he must do so increases in weight if the victims of the theft are left in miserable circumstances. By itself these miserable circumstances need not constitute a reason to give something. But since there is an obligation to give, the reasons behind that obligation become more weighty because of the misery.

There are also meta-reasons which make other reasons invalid. That is, because of these meta-reasons, some facts which would otherwise have been reasons are not valid reasons anymore. For instance, force majeure deprives a contract of its force.

Or, the fact that a rule of law says that guardianship over minors should be entrusted to a parent according to the interests of the minor deprives the interests of the parents of their (legal) force. Reasons which deprive other facts from their reason-giving force are called *exclusionary reasons* after Raz [75].

4.10. Types of exclusionary reasons

The two examples of exclusionary reasons belong to two different categories. The first example, about the force majeure, has to do with the *scope of the warrant*. The fact that a case is outside the scope of a warrant makes reasons based on that warrant invalid. Reasons of this type are *strictly exclusionary*. They have no impact on the conclusion of the reasoning otherwise than by excluding some other facts as reasons.

The second example has to do with the separation of points of view. The legal point of view is characterized by a number of rules (legal warrants) determining what facts are relevant (reasons) for the law. The rules may be based on pre-legal reasons, but once the rules have been formulated, the force of these pre-legal reasons is lost. Only what is identified as a reason by a legal rule has force in law. The applicability of such a rule excludes the force of all other reasons. Thus, the fact that it is in the interest of the child that it should be entrusted to its father, excludes the relevance of the facts that the father and the mother both love the child very much and would like to be its guardian. Of course that love and those wants can be factors determining what is in the interest of the child. But they do not count as independent reasons, which can be weighed along with or against the interest of the child.

4.11. Exclusionary reasons and mandatory rules

This last type of exclusionary reason is not strictly exclusionary. The fact that it is in the interest of the child that it should be entrusted to its father has also by itself the force of a reason. This type of reason is extra forceful. It not only provides a reason for a conclusion itself, but it also excludes other facts as reasons. It puts its concurrents out of action and will therefore usually win the battle of reasons.

Reasons of this type derive their force from mandatory rules [Raz 75, p. 73 ff.] Mandatory rules identify in their conditions of applicability which facts count as reasons for their conclusion. These facts are exclusionary reasons. They are not only reasons for the conclusion of the rule, but they also exclude other facts as bearing upon that conclusion.

4.12. The scope of exclusionary reasons

Exclusionary reasons have a scope. This scope should not be confused with the scope of warrants. The scope of a warrant determines within which boundaries a warrant can be validly applied. The scope of exclusionary reasons indicates *what types of facts are excluded as reasons* by the exclusionary reason [Raz 79, p. 22]. For instance, while drafting a legal rule, the lawgiver has taken a number of pre-legal reasons for and against this rule into consideration. These reasons are more or less incorporated in the rule. When the rule is applicable, these facts should not be taken into consideration independently while making legal decisions. They are excluded as *legal* reasons. But it may be the case that the lawgiver oversaw some relevant pre-legal reasons while drafting the law. At least in some views about legal decision making, this is why they should not be excluded as reasons by the applicability of the legal rule.¹⁴ In that case the pre-legal reasons can also be legal reasons.¹⁵ They fall outside

the scope of the exclusionary reason identified by the mandatory rule.

4.13. *Default reasoning and background*

An important difference between traditional PL and RBR is that PL is monotonous and that RBR is inherently non-monotonous. There are at least three reasons why RBR is non-monotonous.

Firstly, reasons on their own never give decisive support to their conclusion. They only contribute with their weight to a conclusion and can in principle always be outweighed.

Secondly, reasons can always lose their force because of some exclusionary reason which excludes them.

And thirdly, the weight of the reasons involved in some argument can be changed because of the presence of some weight changing meta-reason.

Of course it is always possible to pretend that only the facts actually adduced in some argument are relevant reasons. On basis of this logical counterpart of the closed world assumption it is possible to develop a monotonous logic of reasons. But this is an unnatural way of going about in a logic of reasons. The whole idea of working with reasons, which only indicate the direction of a conclusion and not the destination, resists such an approach.

Instead an approach based on *default reasoning under the presupposition of backgrounds* should be adopted. Such an approach works with conclusions which are relativized to the reasons on which they are based.¹⁶

Furthermore a background containing a number of warrants is assumed. This background guarantees the force of a number of reasons. But the background is only a presupposition. It can always be the object of discussion.

If the background is drawn into the discussion, the argumentation is extended with an extra layer. For instance: the discussion moves from the legal status of a specific case to the validity of the rules used in deciding that case. Or the discussion moves from the validity of legal rules to the meta-legal criteria for identifying legal rules. In both cases something that was originally presupposed and belonged to the background (the validity of some legal rule, or of some rule of recognition) becomes the topic of argumentation. In such cases the nature of the discussion changes from plain legal fact to legal-dogmatic and possibly even to legal-theoretical. These changes are reflected in the arguments which are allowed and the warrants which are valid.

5. Conclusion

In the preceding sections I have presented what may be called the prolegomena for a logic of reasons. Of course the results should be formalized in order to be applicable in a LKBS. The enterprise of formalization falls outside the scope of this paper, but I hope to have given sufficient information to get at least an impression which direction such a formalization might take.

6. Notes

1. A formalization of what I mean with 'traditional predicate logic' can be found in e.g. [Hughes and Cresswell 68, p. 133 ff.]
2. Von Wright tried to solve this problem by distinguishing between operators, working on action types and predicates, working on action tokens [81]. This has

the disadvantage that the word 'forbidden' has a different status in rules than in judgments about individual actions.

3. This type of formalization is used in [Hintikka 70].
4. This approach is taken by [Brouwer 90].
5. Under 'anancastic qualification' I understand qualification by operators which are usually called 'modal', such as 'necessarily' and 'possibly' [Von Wright 71]. Strictly considered, 'modal' is a more comprehensive category to which anancastic, deontic and epistemic qualification all belong. It is a common technique to express subjunctive conditionals by means of modal operators. [Haack 75, p. 179-181, Lewis 68 and Stalnaker 68].
6. Cf. what Toulmin has to say about the ambiguity of the maior premiss is the syllogism [58, p. 107-113].
7. Literature on this distinction can be found in [Brecht 59] and [Hudson 69].
8. These topics are explored extensively in [Hage 87].
9. This sentence is ambiguous. It can be the formulation of a rule, in which case it does not express a fact. It can also describe the normative consequences of the rule, in which case it does state a fact. [Kelsen 60, p. 57 ff.; Hage 87, p. 158 ff.]
10. It is, however, possible to give an interpretation of weighing reasons in terms of the interactions of nodes in an artificial neural network.
11. This approach seems to work in the Leidraad LKBS. [De Wildt and Quast 89]. In the Prolexs project a similar technique is used under the heading of 'abstraction finder' [Walker and Van den Berg 88]. The HYPO project also uses a similar technique [Rissland and Ashley 87].
12. I will call the facts for or against which reasons plead 'conclusions' of these reasons.
13. Cf. [Hage 87, pp. 160 ff.], where the word 'nexus' is used for what are called warrants here.
14. Cf. the case of Maring vs. Assuradeuren discussed in [Hage 90].
15. The warrant which incorporates these pre-legal reasons into the law might be a principle in the sense of [Dworkin 78].
16. In this respect my approach is similar to that of Doyle [79].

7. References

Alchourrón, C.E. and Bulygin, E. (1989). Limits of Logic and Legal Reasoning, *Pre-proc. of the Third International Conference on Logica, Informatica, Diritto*, (A.A. Martino ed.), Florence, vol. 2, pp. 1-20.

Alchourrón, C.E. and Martino, A.A. (1989). A Sketch of Logic without Truth, *Proc. of the Second International Conference on Artificial Intelligence and Law*, ACM, New York, pp. 165-179.

Bench-Capon, T.J.M. (1989). Deep Models, Normative Reasoning and Legal Expert Systems, *Proc. of the Second International Conference on Artificial Intelligence and Law*, ACM, New York, pp. 37-45.

Brecht, A. (1959). *Political Theory: The Foundations of Twentieth-Century Political Thought*, Princeton University Press, Princeton.

Brouwer, P.W. (1990). *Samenhang in het recht*, Wolters-Noordhoff, Groningen 1990.

Castaneda, H.N., (1989). The Content of Legal Speech Acts and Deontic Logic, *Pre-proc. of the Third International Conference on Logica, Informatica, Diritto*, (A.A. Martino ed.), Florence, vol. 2 pp. 203-234.

Doyle, J. (1979). A Truth Maintenance System, *Artificial Intelligence*, vol. 12, pp. 231-272.

Dworkin, R. (1978). *Taking Rights Seriously*, 2nd ed., Duckworth, London.

Gardenfors, P. (1989). The Dynamic of Normative Systems, *Pre-proc. of the Third International Conference on Logica, Informatica, Diritto*, (A.A. Martino ed.), Florence, vol. 1, pp. 293-300.

Haack, S. (1975). *Philosophy of Logics*, Cambridge University Press, Cambridge e.a.

Hage, J.C. (1987). *Feiten en betekenis*, Leiden.

Hage, J.C. (1988). Rechtszekerheid, *Proc. NAIC '88*, (M. van Someren en G. Schreiber eds.), Amsterdam, pp. 214-223.

Hage, J.C. (1990). Meta-kennis voor juridische kennissystemen, *Proc. NAIC '90*, (H.J. van den Herik and N.J.I. Mars eds.), Kerkrade, pp. 299-308.

Hare, R.M. (1963). *Freedom and Reason*, Oxford University Press, Oxford e.a.

Hart, H.L.A. (1961). *The Concept of Law*, Oxford University Press, Oxford e.a.

Hintikka, J. (1970). Some Main Problems of Deontic Logic, *Deontic Logic: Introductory and Systematic Readings*, (R. Hilpinen ed.), D. Reidel, Dordrecht e.a., pp. 59-104.

Hudson, W.D. (ed.) (1969). *The Is-Ought Question*, London.

Hughes, G.E. and Cresswell, M.J. (1968). *An Introduction to Modal Logic*, Methuen, London.

Kelsen, H. (1960). *Reine Rechtslehre*, Franz Deuticke, Wien.

Lewis, D. (1968). Counterpart Theory and Quantified Modal Logic, *Journal of Philosophy*, vol 65, pp. 113-126.

Linsky, L. (ed.) (1971). *Reference and Modality*, Oxford University Press, Oxford e.a.

Linsky, L. (1977). *Names and Descriptions*, University of Chicago Press, Chacago, London.

Prakken, E. (1989). Incomplete and Inconsistent Knowledge in Legal Expert Systems, *Pre-proc. of the Third International Conference on Logica, Informatica, Diritto*, (A.A. Martino ed.), Florence, vol. 2 pp. 761-780.

Raz, J. (1975). *Practical Reason and Norms*, Hutchinson, London.

Raz, J. (1979). *The Authority of Law*, Oxford University Press, Oxford e.a.

Rissland E.L. and Ashley, K.D. (1987). A Case Based System for Trade Secrets Law, *Proc. of the First International Conference on Artificial Intelligence and Law*, ACM, New York, pp. 60-66.

Stalnaker, R.C. (1968). A Theory of Conditionals, *Studies in Logical Theory*, (N. Rescher ed.), Blackwell, Oxford.

Strawson, P.F. (1971). *Logico Linguistic Papers*, Methuen, London.

Toulmin, S.E. (1958). *The Uses of Argument*, Cambridge University Press, London, New York.

Walker R.F. and Berg, P.H. Van den (1988). Prolexs, an Object Oriented Legal Expert System, *Five articles on A.I. and Legal Expert Systems*, (Herrestad, H. and Maesel, D.S. eds.), Complex 13/88, Universitetsforlaget, Oslo.

Wildt, J.H. de and Quast, J.A. (1989). The Concept of 'Commensurate Work' in a Legal Knowledge Based System, *Proc. Expert Systems in Law*, Bologna.

Wright, G.H. Von (1971). A New System of Deontic Logic, *Deontic Logic: Introductory and Systematic Readings*, (R. Hilpinen ed.), D. Reidel, Dordrecht e.a., pp. 105-120.

Wright, G.H. Von (1981). On the Logic of Norms and Actions, *New Studies in Deontic Logic*, (R. Hilpinen ed.), D. Reidel, Dordrecht e.a., pp. 3-36.

INVOLVEMENT, PHASES AND SATELLITES AIMS FOR RESEARCH AND DEVELOPMENT OF LEGAL KNOWLEDGE BASED SYSTEMS

A.W. Koers, D. Kracht, M. Smith, J.M. Smits, M.C.M. Weusten
Project on the Use of Computers in Law
Faculty of Law, University of Utrecht
Boothstraat 6
3512 BW Utrecht
The Netherlands

Summary

In this paper we will state the nature of our commitment to AI and law in terms of the theoretical and practical aims of our research. Focus here will be on the practical side: our view of how knowledge based systems in law should be developed. The first section ends with a list of concepts c.q. assumptions that have figured as the basis for our further work. In the next section, the methodology that we use will be roughly outlined. In this description three of the tools we developed for building knowledge bases systems in law will be mentioned briefly. The application of the methodology and tools in a concrete project is the subject of section 4: the development of a prototype system called LASYR will be discussed in some detail. Finally, the insights gained with the development of LASYR will be confronted with the original assumptions.

As regards terminology: we speak of an Advisory System for Legal Questions (abbreviated as ASLQ). Firstly, we want to stress that a system should not, in our opinion, have more than an advisory function. Secondly, we want to avoid the connotations of the term "expert system".¹

1. Motivation of the commitment

To put it very crudely: we have no commitment as such to AI and law. Rather, our commitment is to the use of computers in law. We take the law as a starting point and ask ourselves how lawyers can benefit from the new technologies and insights. Clearly, this involves more than knowledge based systems.

The most obvious use that lawyers can make of computers is, of course, for administrative purposes. Administrative applications are, however, not the most interesting topic from a scientific point of view. But issues like legal documentation, document generation, and also knowledge based systems are surely of interest. So for us interest in knowledge based systems in law is no more than a derived interest.² Nevertheless most of our research concerns these systems - so we are involved.

Turning now to the aims of our research, a distinction can be made between theoretical aims on the one hand and practical aims on the other.

A theoretical question is, for example, to gain a better insight in the nature of legal reasoning by looking at it from a new perspective. With the possibilities of implementing some forms of deductive reasoning, the question surfaces as to what extent legal reasoning can be reduced to deductive reasoning [Smith 89].

Another theoretical question concerns legal knowledge representation: what does legal knowledge look like, and how can it best be represented [Koers 90], [Weusten 89]. Which of the general techniques for knowledge representation can be used in the legal domain, and which do not seem appropriate? And, more importantly, why?

Under the heading of "practical aims" the construction of operative systems can be mentioned, or rather, in our case, the development of tools to facilitate the building and maintenance of such systems. Of course, one cannot develop tools without at the same time testing them. So, in the process of research a number of prototype ASLQs has been built.

As was already stressed above, we feel that any attempt to apply results and insights gained in other areas to the legal domain should be founded upon the characteristics and peculiarities of the law. The law should not be tailored to fit general theories and techniques of computer science and technology. It should rather be the other way round.

This is not to say that it is unacceptable to make some necessary presumptions and reductions in relation to the law and the work of lawyers. The concepts that have formed the basis for our research can be divided into two groups. The first consists of concepts that relate to the methodology and tools used to build knowledge based systems in law:

1. Legal knowledge should not be primarily acquired from expert opinion, but directly from the sources of law such as legislation and case law. Therefore, knowledge acquisition and knowledge elicitation is from written material only.³
2. ASLQs must be capable of accommodating frequent changes in their knowledge base, as the law frequently changes.⁴
3. In law there is a fiction of certainty; unlike most other domains, there is no room for margins of chance.⁵
4. Special explanation facilities are needed so as to accommodate the fact that lawyers argue cases; sometimes a list of relevant arguments is more useful than the conclusion itself.⁶

The second group is made up of concepts that relate to a specific application:

5. The domain of an ASLQ must be very clearly specified and the purpose that the system is designed for should be stated explicitly.⁷
6. Common sense reasoning is not needed to generate useful conclusions (since the user decides on the competence of the system and the function of the system is restricted to that of giving advice).⁸

2. Methodology and tools

A number of methodologies is available for the development of data processing systems. All these methodologies have in common that the development of a system is divided into several phases so that the total process becomes more manageable. Usually, the phases are more or less independent from each other. For each phase the steps to be taken and the results to be achieved are indicated. The results of each phase are passed on to the next phase (e.g. [Buchanan 83]). The methodology that we propose to use in building ASLQs consists of the following four phases:

1. Definition and analysis of legal knowledge.
 - a. First of all, the borders of the domain must be specified as clearly as possible, both in terms of breadth and depth. After that, the relevant sources (such as statutory provisions, case-law and maybe literature) can be collected.
 - b. The next step is to structure the knowledge and to identify its composing elements, i.e. subunits of related knowledge within the domain.
 - c. Finally, the appropriate knowledge representation formalisms for each subunit must be drawn up. The input program called ANALIES is helpful here: using ANALIES, it is easy to fill the knowledge base with knowledge.
2. Programming.

A basic idea underlying our methodology is that programming should be done automatically on the basis of the knowledge representation drawn up in the first phase. Using the program called GENER_DB a ready-for-use ASLQ can be generated. The ASLQ consists of two main parts: the text files (with questions and conclusions, help and statutory provisions and case law) and the Prolog-environment (the Prolog-program itself and facts concerning the statutory provisions and case law).
3. Testing and validation.

Testing involves checking the technical performance of the program. Validation refers to checking the accuracy of the program in relation to the legal domain in question. Validation should, therefore, confirm whether or not all legally valid possibilities are present (no less and no more), as well as whether or not these possibilities are invoked by the program in a manner that is correct from a legal point of view.

Needless to say, validation of ASLQs is a crucial factor in the acceptance of such systems by the legal profession.

A very useful tool in this phase is the program called PATHANAL. Through parsing, PATHANAL can provide all the possible paths through the program, in other words, the ASLQ's decision-graph. In this way looping is sometimes detected: from predicate A another predicate is called and from that predicate (possibly via other predicates) predicate A is called again. These situations have to be analyzed very carefully. They may reflect the actual legal situation, but more probably they indicate that some mistake was made in the phase of Definition and analysis.

4. Maintenance.

Maintenance must be taken into account right from the start. In developing ASLQs we use simple features such as keeping text-files outside the programs, and more complicated features such as using a generator-program instead of programming an ASLQ directly. The latter feature makes it possible to regard (and treat) an ASLQ as a "disposable" program. On the basis of an updated knowledge representation, a completely new ASLQ can be generated easily.

All these phases are interrelated as can be seen in Figure 1. The tools used are also indicated.

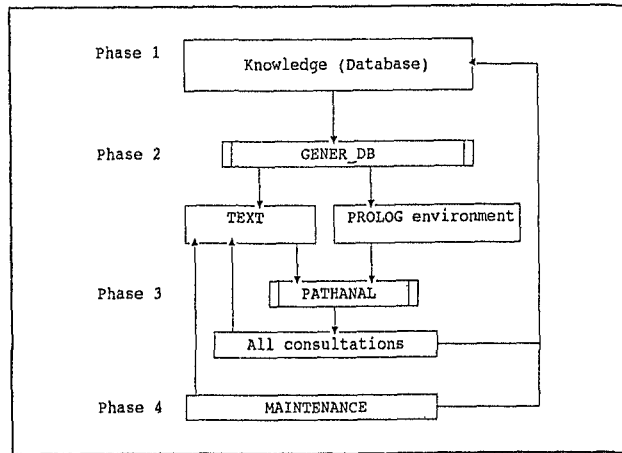


Figure 1: Overall view

In the next section attention will turn to one of our prototype ASLQs, called LASYS.

3. LASYS

3.1. Domain

The Radio Regulations of the International Telecommunication Union (ITU) establish a procedure designed to prevent interference between two earth stations transmitting to a satellite at the same time on the same frequency. In other words, the aim of the procedure is to achieve co-ordination between the owners of earth stations and satellites. The procedure is very complex, both from a legal and from a technical point of view. A special feature of this domain is the need to be able to switch easily between different languages.

Figure 2 illustrates the position of the domain within the context of the International Telecommunication Convention.

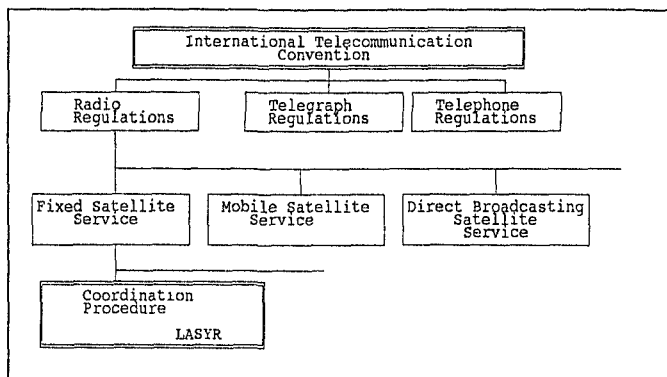


Figure 2: Domain Lasys

The prototype ASLQ called LASYR [Smits 89] provides an answer to the question of whether or not co-ordination with other telecommunication administrators is required if an administration wishes to start using a frequency in a telecommunication satellite network. For this purpose, LASYR has a knowledge base with the relevant parts of the Radio Regulations (which are treaty provisions, not statutory law).

3.2. The development of LASYR

A first version of LASYR was demonstrated in september 1988 to the Center of Telecommunication Development (CTD) of the ITU. The Indian member of the CTD then proposed to establish a joint effort to develop the prototype further into a program capable of handling all legal issues in the field of co-ordination of Fixed Satellite Services. In December 1988 an agreement between the Netherlands Postal and Telecommunication Foundation NEPOSTEL, the TCIL (Telecommunication Consultants India Ltd.) and the Project on the Use of Computers in Law was reached to:

1. Transfer programming knowledge from the Netherlands to India;
2. Develop LASYR with the help of Indian radio- and computer specialists;
3. Test and validate LASYR in the Netherlands;
4. Translate and input all ITU languages in India (French, Spanish, Arabic, Russian and Chinese); and
5. Submit the results to the CTD for further distribution to the ITU member countries.

It turned out that it took very little effort to explain our methodology to the Indian radio expert, who could then start working on the Definition and analysis phase. With some help and co-operation the phases of Definition and analysis and of Programming were completed by three Indian specialists working on the project in six weeks.

In the phase of testing, a loop in the law itself was detected. Validation was done by radio-experts from the Ministry of Transport and Communications in the Netherlands, who did not find any misrepresentation in the knowledge of the domain.

After this validation phase the texts of the questions and the help facilities were translated into Spanish.

During development the system was demonstrated several times to officials of the ITU in Geneva. At a demonstration for the organ of the ITU responsible for the domain, i.e., the International Frequency Registration Board (IFRB), an interesting aspect of our methodology came to the light. The officials agreed that the work on the knowledge representation had resulted in a correct representation insofar as knowledge could be derived from the "paper" sources of international law in this field. However, the Radio Regulations are the result of more than 30 years of negotiations at the international level. Therefore, they contain a great deal of intentionally ambiguous wording due to political compromises.

The outcome of these demonstrations was that expert officials of the IFRB should make a new knowledge representation of the Radio Regulations on the basis of their own interpretation in order to transform LASYR into a system that can be used in

practice. However, constructing a new knowledge representation will involve financial commitments from the ITU. It remains to be seen whether the ITU will be able to allocate money for this.

4. Discussion

Coming back now to our original assumptions and concepts (section 2) the following observations can be made on the basis of our experiences so far with LASYR:

1. In the case of LASYR, the actual work in the phase of Definition and analysis was done by the experts themselves - so indeed, the knowledge acquisition and knowledge elicitation were from written material. However, it was also clear that the definition and analysis could not be done by any lawyer just on the basis of the sources of law. An interesting aspect of this domain is that under the ITU Convention the authority to determine interpretations for use in practice rests exclusively with the IFRB.

This situation is in clear contrast to our first assumption where we explicitly state that sources of law suffice. So we should reconsider this concept, especially if international law is concerned.

2. There is no experience yet with a change in the regulations (or for that matter a different interpretation) that required a new LASYR to be generated.
3. If there is one domain in which the fiction of certainty holds, it is LASYR's. The final conclusion of LASYR is a "yes" or "no" (with the corresponding arguments and/or the possible actions to be taken) to the question of whether or not co-ordination with other Telecommunication Administrators is required.
4. The special explanation facilities, such as help and treaty provisions, are implemented.
5. Within the ITU there is general agreement on the purpose and the limitations of the co-ordination procedure as contained in LASYR.
6. As LASYR is concerned with a very specific and a very technical domain, our assumption that the system does not need common sense in order to reach a legally valid conclusion was confirmed. It should be noted that common sense is relative to the domain in question - for example, facts concerning radio-frequencies may not belong to the collection of common sense in general, yet they may be regarded upon as common sense for the potential future users of the system.

The final conclusion of the LASYR project is that the methodology and tools we developed indeed considerably facilitate the construction of ASLQs. Moreover, it turned out to be relatively easy to train other persons in using our methodology and tools.

However, we still feel that in order for knowledge based systems to be really of use in practice, integration with other facilities, such as the generation of the appropriate documents, is essential (e.g. [Bench-Capon 90]). Additionally, it would be very useful

in practice if the structuring of the domain knowledge of knowledge based systems could be used for information retrieval purposes as well.

Also, the theoretical questions formulated in the beginning of this paper are barely touched upon, and it is to these more abstract issues that our present and future research will be directed. After all, developing prototypes such as LASYS is for us no purpose in itself.

5. Notes

1. A complete overview of our research, both of its theoretical foundations and its practical results, is given in "Knowledge Based Systems in Law, In Search for Methodologies and Tools" [Koers 89].
2. We believe that knowledge based systems in law, more than any other application of computer science and technology to law, will have a profound impact upon the work of professional groups such as lawyers and upon those who depend in any way upon their work. Knowledge based systems in law have the potential to alter dramatically the way in which law is administered, as well as the way in which lawyers work: suffice it to note that technically these systems are capable of making decisions that until now have been the exclusive domain of human decision-makers.
3. This concept is motivated by the following considerations:
 - That law is well-documented and that this documentation is highly structured;
 - That the documentation meets high standards of completeness, reliability and authority; and
 - Most importantly: that law as such is found primarily, if not exclusively, in legislative documents and/or case law and not in the heads of experts (which does not imply that experts do not know a great deal about law).
4. If legal knowledge is to be derived primarily from legislation and/or case law, a knowledge based system in law should be capable of accommodating quickly and easily every change in legislation and/or case law that affects the domain with which the system is concerned.
5. We take the view that law differs fundamentally from other domains of knowledge in that law postulates a fiction of certainty in relation to legal decisions. This can be illustrated best by referring to the decisions made by judges and courts of law.

The judge/expert differs from other experts in that, as a general rule of law, he or she is not allowed to decide not to decide, for example by saying that there is not enough knowledge to reach a conclusion. Similarly unacceptable is the decision of a court of law to the effect that the law is for 40 percent clear and for 60 percent unclear. Under the legal principle of non liquet such avenues of decisions are not available to a judge or a court. At most, a judge may decide that he or she will not hear a case.
6. The work of lawyers is characterized by a perennial conflict of arguments. Every lawyer is aware of the fact that more often than not his or her solution to a legal problem, as well as the grounds for that solution, will be confronted by other

solutions and the scrutiny of other lawyers. This implies that a lawyer may be more interested in the arguments than in the conclusion.

7. It is, of course, true that in law, too, everything is linked to everything else. Nevertheless, we consider it possible to identify domains of legal knowledge that are both sufficiently small to be handled by a knowledge based system and sufficiently autonomous to serve as the only reference for solving problems arising within that domain. Indeed we feel that the long history and tradition of legal doctrine is most helpful here as it has produced detailed divisions in the overall domain of legal knowledge that are accepted and used by all lawyers.
8. It is impossible to envisage a lawyer dealing with a legal problem without using common sense, *i.e.* the knowledge which all humans share in relation to everyday situations. However, the fact that lawyers, like most people, cannot work without common sense does not in itself imply that knowledge based systems in law can only be functionally valid if they too have common sense. The real question is whether or not on the basis of legal knowledge only, *i.e.* without common sense, these systems can generate answers to legal questions that are as valid from a strictly legal perspective as the answers given by lawyers who also invoke only legal knowledge.
We take the view that this question can be answered positively if legal knowledge is understood to cover not only knowledge derived from legislation and case law, but also certain facts of a non-legal nature that are specifically required for solving legal problems in a particular domain.

6. References

- [Bench-Capon 90] Bench-Capon, T.J.M., & Dunne, P.E.S. (1990). An approach to the Integration of Legal Support Systems. In A.M. Tjoa & R. Wagner (Eds.), *Database and Expert Systems Applications* (pp. 105-111). Wien, New York: Springer-Verlag.
- [Buchanan 83] Buchanan, B.G., Barstow, D., Bechtal, R., Bennett, J., Clancey, W., Kulikowski, C., Mitchell, T., & Waterman, D.A. (1983). Constructing an Expert System. In F. Hayes-Roth, D.A. Waterman & D.B. Lenat (Eds.), *Building Expert Systems* (pp. 127-167). Reading (Mass.): Addison-Wesley.
- [Koers 89] Koers, A.W., Kracht, D., Smith, M., Smits, J.M. & Weusten, M.C.M. (1989). *Knowledge Based Systems in Law. In search of Methodologies and Tools*. Deventer: Kluwer.
- [Koers 90] Koers, A.W., Kracht, D., Smith, M., Smits, J.M. & Weusten, M.C.M. (1990). Criteria for the Classification of Legal Knowledge Systems. Advisory Systems on Legal Questions. In D. Kracht, C.N.J. de Vey Mestdagh & J.S. Svensson (Eds.), *Legal Knowledge Based Systems* (pp. 23-35). Lelystad: Koninklijke Vermande BV.
- [Smith 89] Smith, M. (1989). The Formalization of Legal Reasoning. In A.A. Martino (Ed.), *Pre-proceedings of the III International Conference on Logica, Informatica, Diritto* (Vol. 1) (pp. 693-704). Florence: CNR, Istituto per la documentazione giuridica.

- [Smits 89] Smits, J.M., & Kracht, D. (1989). Experiences with a Methodology for developing Advisory Systems for Legal Questions. In A.A. Martino (Ed.), *Pre-proceedings of the III International Conference on Logica, Informatica, Diritto* (Vol. 1) (pp. 705-721). Florence: CNR, Istituto per la documentazione giuridica.
- [Weusten 89] Weusten, M.C.M. (1989). Maintenance of Knowledge in Advisory Systems on Legal Questions. In A.A. Martino (Ed.), *Pre-proceedings of the III International Conference on Logica, Informatica, Diritto* (Vol. 1) (pp. 853-862). Florence: CNR, Istituto per la documentazione giuridica.

THE JURICAS-SYSTEM IN PRACTICE: DECISIONS IN A SOCIAL SECURITY ENVIRONMENT

C. van Noortwijk, P.A.W. Piepers and J.G.L. van der Wees
Workshop for Computer Science and Law
Erasmus University Rotterdam
P.O. Box 1738
3000 DR Rotterdam
The Netherlands

Summary

After bringing six ready-made legal computer advice systems on the market, the Workshop for Computer Science and Law is now selling its 'empty' JURICAS-shell as well. The advantage is that experts in organisations can build their own computer advice system, which makes it possible to build a system that fits all specific needs. One project was a system for a social security service in the Netherlands. This social security service developed a JURICAS-system on the basis of the Dutch Social Security Act. The intention was not to make a system that gives advice in all cases, but to build a system that supports the user in making routine decisions. The results of the social security system are positive. Some attention has to be paid, however, to the protection of the data within the organization. Implementation of the system at this social security service is expected to take place early in 1991.

1. Legal knowledge based systems

When the development of legal knowledge based systems is discussed, it is implicitly assumed that legal knowledge is available. In this context knowledge is usually comprehended as empirical knowledge. Yet, this is exactly the shortcoming in the field of law; there is little empirical knowledge about law, and sometimes no empirical knowledge at all is available on a specific legal subject.¹ The legal science which attempts to increase the empirical knowledge about law is jurimetrics. Jurimetrics is defined as 'the empirical science which concerns itself with the study of syntax, semantics and pragmatics (and the relationships between them) of demands and authorizations issuing from state organizations'.² Research is called empirical and scientific when it is aimed at increasing knowledge of the world of experience. Knowledge itself has to be gained systematically and has to be falsifiable.³ Empirical knowledge is formed on basis of observations (measurements) which are obtained by a systematic perception of objects in the world of experience.

Since this kind of research is rare in the field of law, the development of a legal knowledge based system will not be practicable in the short term. Only if the term 'legal knowledge based system' is interpreted as just a legal matter system will it be possible to build a computer system with legal contents. Such a system, however, does not say anything about the world of experience, for that empirical knowledge would be necessary, but restricts itself to the opinion of the author (or authors) on the legal subject. The term legal computer advice system is better suited for such a system and will be used henceforth.

In recent years the Workshop for Computer Science and Law has gained a lot of experience with the development of legal computer advice systems with the aid of

the self-developed JURICAS-shell (JURICAS stands for JURIdical Computer Advice System). In practice the JURICAS-shell is used in two ways.⁴ Both ways will be looked at below.

1.1. Ready-made JURICAS-systems

Before starting with the development of a ready-made JURICAS-system, a subject for the system has to be defined. Sometimes, because of the lack of empirical knowledge in the field of law mentioned above, it is difficult to give an unambiguous solution for the legal problem on the basis of legislation, case law and literature. To overcome this difficulty the author of the JURICAS-system can give a solution which is correct in his opinion, and give reasons for his solution to the user of the system. An alternative is that the system asks the user to make a choice from possible solutions. This latter option is often used in ready-made JURICAS-systems.⁵ The fact that the user is asked to make a choice, does not mean that the system is of no value or not practicable, because without the system the user may not have considered these alternatives at all.

During the development of a ready-made system the builder must keep an eye on the target group, so the level of the system is fitted to the user. Since it is also expected that such a system handles special cases and exceptions, the analysis of the legal subject is even more complicated and time-consuming.

Till now five ready-made JURICAS-systems have been brought onto the market and the sixth will follow soon:

- Remanding in custody
- Dismissal law
- Inheritance law
- Sentencing according to Hulsman
- Military service
- Travel documents (soon).

The use of these systems has created the wish on the part of some users to write such systems themselves. Because of this and other reasons the Workshop has decided, in consultation with the publisher Royal Vermande, to bring the 'empty' JURICAS-shell onto the market accompanied by a tutorial course on how to make your own legal computer advice system. During the course the student is taught how to make a system that fits all the needs of his own working environment.

1.2. Self-developed JURICAS-systems

After some JURICAS-courses it became clear that the ones who develop a computer advice system for their own use do not wish to build a system that thinks of everything and is based on legislation, case law and literature, but one that supports them in making more or less routine legal decisions. These users make a lot of similar decisions every day, which all need to look at the same aspects and require a printed decision. Although a legal basis is present and often case law too, the authors base their work on practical knowledge. They reason that the system does not have to follow all legal rules exactly, as in practice all the legal rules are not strictly applied. The aim is to reach a sound decision. Unlike the ready-made JURICAS-systems the author is therefore in a better position to implement a solution in the computer advice system, even for a situation where it is difficult to

give an unambiguous solution on basis of legislation, case law or literature. This is partly due to the fact that the system only has to reach correct decisions within a particular organisation and is not valid for the whole legal field.

The goal of these authors is not to build a system that gives a solution for all cases, but a system that only handles the frequent cases. If a special, rare case is presented to system which has not been implemented, the system reports this and the user has to make his own decision. The author works economically; if an advice system can be written that, for example, solves 90 percent of the cases, and the system can be written in an amount of time of which the costs can easily be recovered, it is desirable to write the system. The time investment for the other 10 percent would be too much to recover the costs and should not take place. An example of such a self-developed system is the project for a social security service that will be described in sections 2 and 3.

1.3. Reasons for the development of legal computer advice systems

The development of legal computer advice systems is not the first goal of the Workshop for Computer Science and Law. It is a derivation of the research done by the Workshop, in which jurimetrics plays a prominent role. As said before, jurimetrics is defined as 'the empirical science which concerns itself with the study of syntax, semantics and pragmatics (and the relationships between them) of demands and authorizations issuing from state organizations'. By studying the syntax, semantical and pragmatical aspects of law a contribution can be made to increasing the body of empirical knowledge about the law. If more empirical knowledge is available about the law, it will be easier for a legal computer advice system to reach an unambiguous solution with regard to cases which had previously seemed ambiguous.

Although the development of legal advice systems itself does not contribute to increasing empirical knowledge directly, these systems are interesting because they provide information by answering users' questions. The systems have an obvious author who anticipates users' questions and who directs his texts to these questions.

2. A Computer-Advice system for Social Security Law

In this section an example is given of a legal computer-advice system which is, at present, being tested out in practice. The system helps with the processing of requests for social security. It was developed by a social service⁶ on its own account, with the use of the JURICAS-software which in turn was developed by the Workshop for Computer Science and Law. The resulting advice-system will probably be put into use early in 1991. Before that, a large number of user tests will take place which have to show if the system works in practice.

What is special about this new computer-advice system is not so much the subject that it is about (the Tessec-system of the Technical University of Twente deals with Social Security Law too)⁷, but that it was put together by the experts themselves, namely by employees of the social service where the system will be used. A big advantage of this method is, of course, that no interaction with a 'knowledge engineer' is needed. A possible source of misunderstanding during the construction of the system is avoided in this way, while the involvement of the employees of the organisation with the project is of course larger; those who are building the system now, will have to work with it themselves later. Maintaining the system - adding new

legislation, but also tuning the system to users' demands - will be much easier when this can be done internally and by the organisation's own employees. It is also expected that the system will be accepted more easily by the users.

To make it possible for a domain expert to build his (or her) own computer-advice system, it is important that certain conditions are met. In the first place, the system has to be very user friendly, not only to the end user, but also to the author. In JURICAS, this is achieved by providing the author with a number of special tools, all of which help in constructing, modifying and testing the prototype system. In the second place, it is very important that authors of a system can rely on the fact that support will always be available when they run into trouble. And in the third place, especially with legal computer-advice systems, it is of importance that the way a system is set up more or less resembles the way lawyers usually work. As they are used to stating the grounds for the choices they make, the system has to give them the opportunity to do so. When they want certain background information to be at their disposal, the system has to provide for this.

Advice-systems that are built using the JURICAS software have an important advantage. Their starting point is the implementation of rules and regulations, not merely the statute law that underlies it. Therefore JURICAS systems can be tailored very well to the needs of the organisation by which they will be used. Even the most detailed implementing orders can be included, to produce a really complete system.

2.1. Nature of the project

The project at the social service came about in the following way. In 1988, a publication on advice systems in law led people in the service to wonder if such a system could be of use to them too. The expectation was that uncomplicated cases could be processed more quickly, so more time would be available for more complicated matters. The system would also possibly reduce the number of errors. These objectives will be discussed later in this section.

An inventory of the market for knowledge systems eventually led to the decision to use the JURICAS-system. Important considerations which played a role in selecting JURICAS were, amongst others, the fact that not only an 'empty' JURICAS-shell, but also ready-made systems could be obtained immediately (so that the achievements of the software could be estimated well) and that a training course for the JURICAS- language was available for those who wanted to build their own systems. An employee of the social service followed this authors training course and he and a colleague, therefore, were exempted from their normal duties to build a first demo system. After this system was approved, the building of a fully fledged advice system for the social service was started. In August 1990, this system was so far advanced that it could be tested in practice.

2.2. Starting points

The authors of this advice system were of the opinion that it was not very sensible to build a system which can always solve all cases. In daily practice, cases that demand an expert's attention always occur. But most of the cases (often as many as 90%) can be handled in a routine way. For most social services in Holland this is the case: most of the decisions that are taken (for example by welfare officers) follow directly from factual conditions. When a large part of someone's work consists of routine work, this can of course lead to problems: not only is this felt to be unpleasant and

undesirable by many people, but it is also possible that cases that look simple and standard at first sight are dealt with too fast, while the one point that makes this case different is overlooked. In other words: in a flood of routine cases, the one case that requires special attention is recognized less easily.

The fact that there is routine work also has a positive aspect. An important point about routine cases is, of course, that they have many things in common. And it is this characteristic that makes it possible to use an automatic system (for instance a computer-advice system) when dealing with cases like these. One important point has to be considered here: how can the small number of special cases be discerned from the large mass of routine cases? A computer-advice system can provide a solution; by posing specific questions it can investigate how special a case is. As much time has been saved on the routine cases, the user now has sufficient opportunity to use his knowledge and experience for this special case.

2.3. Purpose of the system

The purpose of the computer-advice system that was developed by the social service is particularly to improve the service towards clients. The average time taken to deal with a request for a social security payment will become shorter. As the system takes over part of the routine work, employees have more time to talk to clients. And in addition to this, the quality of the decisions taken will improve as, inter alia, the system works as a 'checklist', and so prevents certain facts being overlooked or that certain information has not been obtained when talking to clients. Calculation is a thing that computers do well, so the checking of ages, terms and so on is also a point where the system can save labour.

The programme that is presently being tested has the following functions:

- input of client data, during which no more and no less data are asked than is necessary for taking the decision
- checking of the internal consistency of the input data, making calculations and drawing conclusions
- reporting of the conclusions of the system, on the screen or on the printer.

3. An assessment of the (expected) accomplishments of the system

At the moment, future users are testing the social security system in order to give an assessment of the accomplishments of the system. Apart from handling a case as usual, they are also using the JURICAS-system to decide whether someone will receive a benefit. The reactions of the users will make it clear as to whether the system will meet expectations. Of course the experiences of the future users are of vital importance for the future of the system; will it be implemented or not? Up till now, we have received a lot of positive reactions and it seems that the system serves its purpose.

3.1. The (expected) accomplishments

With the help of the JURICAS-system, which is fully adjusted to their wishes, people working for the social security service will be able to make decisions quickly. The process will take less time than it used to do. One reason for this is that while the request is being dealt with, a report is also being built up by the JURICAS-system.

The one who has to make the decision as to whether the client will receive a benefit or not, will get the printed report and, after checking it, only has to sign it.

It will also be more likely that similar decisions will be made in similar cases. This is an advantage all JURICAS-systems have and this social security system will certainly not be an exception. One reason for that in particular is that all the criteria considered to be relevant (by the author) will come up for examination. No matter who is at the computer, he or she must give a judgement based on the same criteria and must answer the same questions. This will, of course, promote equal treatment or at least equal treatment at the social security service where the system has been built and implemented. Other social security services will possibly have a different policy on certain points, which could lead to other decisions. So the other services cannot always use the system as it is at present. With JURICAS it will not be very difficult to adjust the system to their wishes.

The report is built up while the system is being used. Name, address and so on must be typed in, and are stored in a certain file by the system. When the work is done there will be a report file with all the information that is needed. The user is able to adjust this file with the help of a text-processing programme or to print the report immediately, if everything seems to be satisfactory. This report will be checked and signed by a decision-maker; next case!

Clients of the social security service will be satisfied that similar cases will be treated in a similar manner, especially when it is to their advantage. When this is not the case, it is possible that the use of a computer advice system will still lead to a greater acceptance of the decisions; if the system is able to sum up the exact reasons why the social benefit has been denied, the decision will probably be accepted by the client. Complaints by clients at the counter of the social security service office that they do not receive a particular benefit whereas a neighbour in precisely the same situation does, will probably become less frequent.

On the other hand, it is also possible that clients of the social security service will not feel very comfortable because of the fact that there is a computer on the desk of the civil servant. They might get the feeling that their very important request for a benefit will be treated, and maybe rejected, by a machine. Of course that is not true; it is not the system which decides. The responsibility for the advice given always remains with the user of the system. The client however does not know this. The decision-maker may be able to eliminate the client's suspicion by informing him of the fact that the system is intended as a support and is there to ensure that the employee of the social security service does not forget anything.

An existing problem was the fact that it took a lot of time to collect relevant data. Often the client had gone home again before the data were found, so another contact had to be made. This will be improved with the help of the JURICAS-system. The system collects all necessary data to finish the case.

Although a lot of regulations are indisputable, decision-makers often had to consult the employees working in the field, because they did not apply these regulations as officially they should have. This also cost a lot of time. With JURICAS less mistakes are made and a case is finished much faster.

3.2. The limitations of the system

The responsibility for the advice given remains with the decision-maker. This also implies that the user of the system must have a certain level of knowledge to be able to use the system in the right and proper way. In other words, it is certainly not the case that the development and use of the JURICAS-system will lead to the dismissal of decision-makers in the social security service.

Until now, the system has fulfilled expectations, although JURICAS is still not perfect. This is why the Workshop for Computer science and Law is now, for example, working on a new feature which gives the user of the system the ability to save a completed case, so that it becomes possible to look at it again later, and also to make an archive.

Another feature being developed is the possibility for the JURICAS-system to communicate with other systems. For example, name and address, already known but stored elsewhere, have to be inserted in the JURICAS-system once again before the case can be dealt with; not the best possible situation. It is also very important that the data inserted while working with JURICAS can be written back to, for example, administration files. It should not be necessary to start another program to update the file of a client who, with the help of JURICAS, just received a benefit. A connection with other systems should give the opportunity to get known data, work with them in the JURICAS-system and write them (together with inserted data) back to these systems. Of course the JURICAS-developers are aware of the fact that this is a very important feature and has to be implemented in JURICAS.

3.3. Responsibility and security

When people begin to work with the benefit-system within their organisation, it is certainly not impossible that new problems will arise. In particular, it must be established who is going to be responsible for the reliability and validity of the data in the system, because it is hard to identify the person who entered the data. Of course it is possible to make a hard copy of the data in the system which can be signed, but then the social security needs (another) archive for these hard copies. That does not seem the right approach if automation is the aim of the office. Another possibility to solve this problem might be the development of a feature which makes it possible to connect certain data in the JURICAS-system with a particular person, probably the one who put the data in the system. This, of course, is a security problem which probably can be solved by technical means and by some slight changes in the organisation.

4. Notes

1. See [De Mulder 89] and [Noortwijk 90].
2. See page 240 of [De Mulder 84].
3. See page 40 and further of [Popper 65].
4. See [Noortwijk 90] for the way knowledge is represented in JURICAS.
5. An example is given in [Kerkmeester 88].
6. Details on participants of this project (Social service, authors of the system, etc.) will be published in 1991, when testing of the system will be completed.
7. See page 28 and further of [Nieuwenhuis 89].

5. References

- [Kerkmeester 88] Kerkmeester, H.O. and C.J. van de Velde (1988), De ontwikkeling van juridische computer-adviessystemen (The development of legal computer advice systems), *Computerrecht*, 1988/3.
- [De Mulder 84] De Mulder, R.V. (1984) *Een model voor juridische informatica (A model for the application of computer science to law)*. Vermande, Lelystad.
- [De Mulder 89] De Mulder, R.V., C. van Noortwijk and C.J. van de Velde (1989), De JURICAS-programma's besproken (The JURICAS-systems reviewed), *Computerrecht*, 1989/3.
- [Nieuwenhuis 89] Nieuwenhuis, M.A. (1989), *TESSEC: een expertsysteem voor de bijstandswet (TESSEC: an expert system for the Dutch social security act)*, Kluwer, Deventer.
- [Noortwijk 90] Noortwijk, C. van (1990), Criteria in the JURICAS-project, in: D. Kracht, C.N.J. de Vey Mestdag and J.S. Svensson (eds.), *Legal Knowledge Based Systems*, Vermande, Lelystad.
- [Popper 65] Popper, K.R. (1965), *The logic of scientific discovery*, New York.

COMPUTER - AIDED LEGISLATIVE DESIGN: WORTH WHILE THE EFFORT?

W.J.M. Voermans
Faculty of Law, Tilburg University
Hogeschoollaan 225
5000 LE Tilburg
The Netherlands

Summary

Until now little research has been conducted into the possibilities of computer-applications within the legislative process. This contribution explores the legislative process in search for clues for the possible development and application of an Advisory System on Legislative Questions. Does the ASLQ-concept apply to the problems which face legislators in the drafting phase of the legislative process? By way of a provisional conclusion this contribution tries to sketch an outline of the possible features of an Advisory System on Legislative Questions.

1. Introduction

Though research into and development of computer-applications are booming in many different areas of law, the prospects for the applicability of sophisticated computer-techniques and -programmes in the legislative area have traditionally been grim. The very nature of legislation itself would, it was supposed, oppose most forms of computerization. Not only was legislation considered to be (the result of) a creative process -determined by an infinite number of variables-, the fact that the process of the enactment itself holds fundamental guarantees for legal security, equality before the law and democratic decision-making would leave little room for the use of computers [Hustinx 85]. Furthermore, the absence of routine- or stereotype-decisions -which we encounter in public administration and in some of the activities of the judiciary- would render the domain unfit for computer-applications. This pessimism was equally adopted by those who set out to research the applicability of expert-systems in law [Oskamp 90].

However, does this conclusion with regard to the legislative process, apply to all the different phases of this process? Have researchers not taken a too global view of the phenomenon of legislation, and do legislation and the legislative process not merit the attention of legal computer-science indeed?

In this contribution I will try to approach this problem from two different angles, i.e. the legislative process itself and the developments in legal computer-science. Based on these considerations I will try to draw a provisional conclusion in the last paragraph.

2. Features of the legislative process

2.1. Applicability-research

Like many other law-related activities the legislative process can be characterized as a decision process. In most European countries formal, or statutory laws -to which form of written law this contribution will be confined- are generally prepared by the

governmental administration and enacted in agreement with Parliament. This process holds a great number of constitutionally determined stages of decision. It is this constitutionally embedded process which warrants the fundamental values of legal security, equality for the law, and the democratic calibre of statutes.

Research into the possible implementation of some sort of computer-aid in the legislative process¹ has mostly restricted itself to the perception of the process as a whole [Hotz 84 and Stoyles 89]. And, indeed, when perceived as a whole, indivisible entity, the process leaves little room for the overall use of computers. On the other hand research into the applicability of computer-science within the separate stages of the legislative process is lagging behind. Some aspects within the process have been explored with regard to possible legal computer-science applications, [Ees 85] but little or no attention has been paid to the separate stages of the legislative process.

2.2. The drafting phase

It is for this reason that I would like to focus attention on one particular stage at the beginning of the actual legislative process: the drafting phase. Although it is hard to distinguish actual (chronological) stages -e.g. a separate policy and drafting c.q. editorial phase- in the departmental process of preparation of legislation [Eijkern 77], one can theoretically discern a phase oriented at putting down in writing the structure and text of a future bill. This drafting phase -which encompasses more problems than those of a mere semantic and grammatical nature²- can tentatively³ be described as the phase which is aimed at the drawing-up of a bill (proposal for a statute) taking into account all of the relevant preconditions. These relevant preconditions consist of the policy choices which have already been made, the knowledge concerning the subject-matter of the regulation, the relevant interests, the factual and legal conditions,⁴ and (when available) legislative-technique directives,⁵ as well as (scientific) knowledge about legislation itself. These preconditions or requirements must be met in the interest of a swift, and competent enactment.

2.3. Requirements in the drafting phase

Perceived like this the drafting phase emerges as a decision process in itself, preconditioned by the afore-mentioned requirements. Lawyers (or other professionals) tasked with the initial drafting of a bill -whom I will call "legislators"- will have to make well-considered choices regarding the phrasing, the construction, and the content of a regulation. Depending on the room these conditions allow, legislators will have more or less drafting discretion, as the case may be. In order to be able to make well-considered choices depending on varying conditions, a lot of skill, expertise and experience is required. Some of this expertise and experience is available in writing. For example in the Netherlands (but also in Germany, Austria, Belgium etc.), legislative-technique directives⁶ have been drafted, consisting of an extensive list of requirements which all bills must meet. These directives mainly hold standard-formulations, directions as to which phrasing and constructions should, or should not be used, provisions concerning transitional law and more in general stipulations relating to the construction and content of statutes (i.e. sanctions, legal protection etc.). Moreover, other directives, even more relating to the content of legislation, have been issued by the Dutch government.⁷ Besides governmental directives other (scientific) knowledge about legislation is available in writing also. Some of this written knowledge and most of the governmental directives constitute requirements which all bills must meet, regardless of the subject-matter. We can therefore define them as homogeneous requirements. Homogeneous requirements are the requirements which are, for example, equally important in the drafting of

proposed alterations of statutes, as in the drawing up of environmental statutes or rent acts. In the Netherlands requirements of this nature are mainly to be found in the above-mentioned legislative-technique directives. When discussing homogeneous requirements I will confine myself to these legislative-technique directives. The span of this paper does not allow to examine all existing categories of homogeneous requirements in depth.

On the other hand drafting discretion is restricted by heterogeneous requirements. Requirements of this sort depend to a high degree on the subject-matter of regulation. For example: when drafting an environmental bill which is aimed at reducing air-pollution caused by exhaust fumes, legislators need to know something about the toxicity of exhaust fumes, the relation between exhaust fumes and air-pollution etc. Knowledge of this kind is relatively unique to the subject-matter of regulation and can hardly be considered to be relevant in the drafting of a rent bill.

It stands to reason that if any attempt is made to develop some sort of computer-aid in order to assist legislators in their professional activities, the focus must be on the homogeneous requirements. If we can classify these requirements in an appropriate manner, it will, in a mirror image-way, tell us something about legislative reasoning (which must be directed at meeting these requirements) and may or may not produce clues for the possible representation and subsequent computerization of this type of reasoning.

3. Do legislators need computer-assistance?

What kind of assistance can a computerized system offer in the drafting phase? Do existing computer-systems possess features that can help legislators to make better bills in an easier way?

In order to be able to answer these questions we need to know what are the major problems of legislative drafting.

As we all know, language and writing are creative processes, as is, in essence, the drafting of a bill. However, creativity generates diversity and although creativity is indispensable when drafting a bill, diversity in formulation holds the risk of disharmony and through this inconsistency. Disharmony of and inconsistencies in the formulation of statutes diminish the credibility and legitimacy of legislation in their turn. In a nutshell these are some of the main reasons for harmonization and coordination in legislation. Computers can be of help in this way, as I hope to demonstrate in paragraph 5. When computer-programmes employ a uniform approach to the drafting-problem, use the same directions, model-phrases etc., harmonization can be brought about in a natural way. I will not discuss every aspect of the harmonization that can be achieved by using the computer as an aid in the drafting process, but will content myself with noting that a uniform approach can have harmonizing effects in itself.

Combined with the advantages of time-saving and the easier accessibility of relevant information computer-programmes can prove their worth in the drafting process [Stoyles 89].

4. Computer-assistance: if possible, what kind of system?

4.1. *The drafting of legislation as a suitable aim for ASLQ-research*

How can legislators benefit from the developments in computer-science? Directed towards the problem at hand, we can try to translate this problem into a more specific question: does legislation, or does some stage in the legislative process, offer a suitable aim and domain for research and development of legal knowledge-based systems?

In the foregoing I have made some preliminary remarks about the different stages of the legislative process itself, and the type of knowledge which is fundamental when legislators want to bring about a "relatively appropriate" bill [Hotz 84]. In this respect legislative reasoning differs from legal reasoning because legislative reasoning can never result in a bill that can profess to be totally (legally) correct. The result of the drafting phase, the bill, is a document which is only one possible answer to a legislative question, open for debate. At best a bill can claim to meet the homogeneous and most of the heterogeneous requirements, but meeting these requirements is essential in view of the aims of the drafting phase.

This conclusion automatically rules out the possible application of any goal-driven knowledge-based (KB) system⁸ [Koers 89]. Furthermore, no assistance for legislators is to be expected from systems that pretend to give authoritative or correct -infallible- results, like expert-systems. First of all the diversity of the knowledge involved (especially the heterogeneous requirements) and the impossibility to comprise all of this knowledge into a single system, as well as the impossibility to solve legislative problems in the drafting phase conclusively, leave no room for the development of systems that profess to have disposal of expert knowledge ('expertise') over a specific domain and a mechanism to conclusively solve complex problems in the field in an intelligent way [Oskamp 90]. But does this conclusion towards the non-applicability of the expert-system-concept automatically rule out research into the possible application of other species of KB-systems?

In their study "Knowledge Based Systems in Law" Koers c.s. describe Advisory Systems on Legal Questions (ASLQs) as the systems which are designed to be advisory in nature and assist in the solving of legal problems [Koers 89]. ASLQs accomplish this by confronting the facts of the case as derived from the answers given by the user with a representation of knowledge on the legal domain in question. Consequently, an ASLQ generates its conclusion through an interaction between the user and its own knowledge about the law. In my opinion the key-concepts held in this definition are -apart from the advisory nature of such a system - knowledge representation and problem-solving through interaction. Confronted with the problem of computer-assistance for legislators, the concept of ASLQs can produce clues for research although the setting of the legislative process, i.e. the drafting phase, is quite different from the one the ASLQ-concept regards. Where ASLQs address legal problems, the legislative drafting phase faces a legislative problem, which may imply legal problems, but on the whole cannot be identified with them. To put it briefly: legal problems mainly concern application of the law, while the problem of legislative drafting consists in the making of a text which may, in term, function as statute law. We can, however, short-circuit this terminological dilemma by way of rephrasing the legal problem-solving aspect in the ASLQ-concept. When we understand the solving of a legal problem as the bringing about of a decision, related to a dispute which we can qualify as legal (because it concerns the contestation of rights and obligations, or more formally, because it concerns a law-related issue), which meets legal (statutory) requirements to the highest degree, the ASLQ-concept can well bear meaning and apply to the problems

which legislators face in the drafting phase. The difference between the solution of legal problems and the solution of legislative problems in the drafting phase would then be reduced to the degree in which requirements are met. Legal problem-solving will have to meet all relevant legal standards in order to produce legally correct results, whereas legislative problem-solving in the drafting phase will have to meet all relevant homogeneous and relevant heterogeneous requirements in order to be able to make a relatively appropriate bill.

4.2. The legislative drafting phase as a suitable domain for possible development of an ASLQ

Does all of this procure sufficient cause for the assertion that the legislative drafting phase offers a suitable aim for research and possible development of an Advisory System on Legislative Questions?

Aside from a terminological similarity in activities involved, we will have to pause at the question whether the legislative activities in the drafting phase can be considered a suitable domain for research and development of ASLQs. When we want to answer this question, the criteria which Koers c.s. use to determine the suitability of domains of legal knowledge when trying to construct prototype ASLQs, may be useful [Koers 89]. These criteria are derived both from the concept of ASLQs, as well as concepts with regard to law and lawyers. In this paper I will not discuss these concepts in depth; I will content myself with adopting these concepts in a legislative setting. The limited scope of this paper does not allow an extensive research into the similarities and differences of concepts with regard to law and lawyers compared to concepts with regard to (the drafting of) bills and legislators. I refer to the paragraphs 2, 3 and 4a in which I have discussed briefly the concepts with regard to (the drafting of) bills, legislators and ASLQ's.

When selecting a suitable domain of legislative knowledge to serve as an object for the development of a(n) (prototype) ASLQ I consider the most important criteria in this context to be:

- a) the need for a domain to be not too large and thus become unmanageable, nor too small as to offer no challenge;
- b) the knowledge in a domain should be fairly complicated in structure;
- c) it must be possible to isolate the domain, i.e. the domain must consist of a set of autonomous questions and problems which set the domain apart from other related domains;
- d) there should be some variation in the profile of users.

The criteria mentioned here originate from the features of the ASLQ-concept. These specific types of legal knowledge based systems originally were designed to generate legally valid conclusions on the basis of interactive support in legal decision-making for lawyers and non-lawyers alike, by way of legal knowledge representation and specific case-related input from the user [Koers 90]. In paragraph 4a I argued that the concept can be translated to the legislative-drafting environment. An advisory system in a legislative context never can claim to produce legally conclusive results. A relatively appropriate result will have to suffice. A result like this can be brought about by way of meeting the homogeneous and most of the heterogeneous requirements to the extent of a relative optimum. Notwithstanding this difference there exists a distinct similarity between the law and the drafting-process. Therefore ASLQ-criteria, based upon the features of the ASLQ-concept, can be used to determine whether a phase in the legislative process offers a suitable environment for the development of prototype ASLQs.

When we confront the legislative process with the afore-mentioned criteria, the drafting phase can be considered to meet them. This phase, which I described in paragraph 2b & 2c, is neither too large, nor too simple-structured. On top of that the legislative drafting phase is relatively autonomous and a projected advisory system will certainly not only serve the group of professional legislative draughtsmen, but policymakers and other interested persons, like members of Parliament etc., as well.

Two of the criteria which Koers c.s. discern in their search for suitable domains for ASLQ-development are not comprised in the afore-mentioned list. The requirement that the knowledge must be found both in legislation and case law, and the requirement that the law must be fairly specific are missing. It stands to reason that these criteria do not apply in a process which is aimed at enacting a statute. The drawing-up of statute law-to-be is not solely governed by legislation and case law-knowledge, although other statutes and law in general precondition it substantially. The major part of the (legal) knowledge needed in the drafting phase (in the Netherlands) has been laid down in legislative-technique directives. These directives, which will be substantially revised in the near future, do not merely procure practical formulation hints, but also deal with more complicated, substantial matters. Furthermore, these directives have become quite numerous, which makes it more difficult to observe them. Not observing them means a valuable waste of time in the governmental preparation of a bill as well as possible difficulties in the process of enactment of the bill. In this aspect they can be considered the law of law-making. I think that all that has been stated above gives enough grounds for the conclusion that the drafting phase of the legislative process indeed meets the criteria which are being used to determine whether a specific legal domain offers a suitable aim for research and development of (prototype) ASLQs.

5. Features of an Advisory System in a legislative context

5.1. An Advisory System on Legislative Questions

As I see it, a possible Advisory System on Legislative Questions must have a hybrid character [Vandenb 85] in order to be able to assist legislators in the drafting phase in an optimal way. Besides wordprocessing features, such a system needs an extensive dataprocessing-capacity. Preferably the dataprocessing element possess an internal and external database. In the internal database uniform acts, model clauses and bills, preambles, headings, abstracts of statutes, (governmental) legislative directives, legislative hints, editorial notes, and other relevant data, varying from a legislative thesaurus to full text reports, literature etc., can be made available, while menu-guided and uniform full-text-retrieval of external legislative databanks (containing statutes) must be made possible. As for the wordprocessing features one could imagine facilities for styling the text of a bill, a legislative spelling-check, downloading-facilities for information retrieved from internal as well as external databanks, etc. [Stoyles 89] With features like this, though, a mere extensive electronic library is brought about. The crux of an Advisory System is, as we have noted above, advising through interactive knowledge representation. How can knowledge about legislation be brought into a system without impeding or slowing down the drafting-activities (which basically still should be performed by the legislators themselves)?

Here legislators might well benefit from an interactive-wordprocessor, which as a rule will interfere at a minimum with the actual phrasing of the text of a bill. The

system should merely prestructure the drafting-approach by way of levels which are linked to an internal database.

These levels, in their turn, can be linked to questions or points of attention in the internal databank, like notes, legislative-technique directives and other items that merit special attention within a particular level. Some of the most important points of attention must be "hard" -meaning that they automatically arise and actually do interfere-. Other less fundamental items that deserve attention (the major part), can be "soft" and should merely be pointed out to the user as hints.

The "hard" points of attention should of course for the major part consist of legislative-technique directives. In these directives a lot of legislative-drafting knowledge is comprised. They also procure the bulk of the homogeneous requirements that have been discussed in paragraph 2. Although the legislative-technique directives contain in themselves a lot of legislative-drafting knowledge, they do not do so exclusively. Other knowledge, like constitutional knowledge about (statute) law and law-making, can also be considered to be fundamental. Consequently they must be transformed into "hard" points of attention within the proposed system.

A system like the one proposed should not merely retrieve the written knowledge from documents held in a database. It should also process knowledge by way of advice or guidance. This aspect can be provided by a model-law-structure. Based upon an analysis of statute laws such a model-law-structure can be drawn up. This model-law-structure consists of elements all statute laws as a rule must possess. The model-law-structure should not only convey these elements but should rank them in a consecutive sequence. Brought into the proposed system the elements of the model-law-structure constitute the different levels. In their turn these levels are linked to questions and points of attention in the databanks. The passage through the different level-stages must be a "hard" point of attention in itself. When a user fails to deal with a level-stage or a "hard" attention-point within a level-stage it should be made visible for other legislators or other participants in the legislative process.

Because the knowledge of legislators operating in the drafting phase partly consists of knowledge of requirements and knowing how to meet them in order to comply with the aims of the drafting phase, the level structure and interrelative question-structure constitute knowledge representation themselves.

A system like this can also be considered to be purely advisory in nature even though it uses "hard" points of attention. "Hard" points of attention do not inhibit the user to progress in the process of drawing up a bill per se. When a users fails, for whatever reason, to deal with a "hard" point, he/she is free to do so, without interference of the system. The only consequence is that this non-compliance is made visible in the final result. "Soft" points of attention do of course not inhibit users by there very nature.

5.2. Limitations

Initially an Advisory System on Legislative Questions would merely offer a modest tool for legislators, but potentially it can have a big impact. It only offers help and advice in drafting-matters that are somehow linked to the form, shape, and sometimes to the contents of bills. Within a particular level it can offer more substantial and detailed advice. However, drafting logical errors will still be possible, although some of the levels will of course be interlinked in order to make impossible choices that logically rule out one another. To deal with these limitations a constant-evaluation-mechanism must be brought into the system itself, enabling the users to comment on the advice they were given, or problems they have encountered when

using the system. At the same time the results gained from the system should be subject to scrupulous analysis from a legislative-expert group.⁹ In the same way as the legislative-technique directives were drawn up, a list of known mistakes and errors can be created and subsequently be brought into the system as "hard" points of attention. A dynamic error-catalogue can, combined with a user-comment-catalogue, guarantee a constant-update. A constant evaluation mechanism will even allow the system to grow into more than a mere drafting-aid.

There is still another reason why an Advisory System on Legislative Questions always will be modest in its assistance. An ASLQ of this type tries to help and guide users during the actual drafting process itself. Because it allows the users to reach a result in more than one way, the conclusive detection of logical errors is very difficult. The system proposed here is not primarily meant to be an evaluation-system in itself. It is not unthinkable that in the future a legislative-evaluation-system can be linked to the ASLQ proposed. An Advisory System on Legislative can even create new possibilities for the development of such a system.

6. Conclusion

The care for the quality of legislation is always an important item on the political agenda. For those who know the size of the body of legislation and the pace with which this corpus expands, the need for a harmonized and systematic approach to legislative problems will be evident. Advisory systems may help legislators in doing this. Advisory systems may be practical tools for those who are tasked with the drafting of bills and those who will be in the future. Systems like the one proposed can not only be used for the afore-mentioned purposes, but can also give rise to a more in-depth understanding of the activities performed by legislators in the drafting process and can eventually tell us more about legislation itself.

In his article "The Unfulfilled Promise", Stoyles supposes that the lack of computer-applications in legislative settings is caused by the fear of legislators that the legislative process might be depersonalized by the introduction of computers [Stoyles 89]. This fear originates from the presumption that computers in this setting are able to replace the activities of legislators. I hope I have succeeded in proving that there can be no question of such a replacement and that the basis of the drafting activity will always be the work of creative professionals. But the work may well benefit in a number of aspects from legislative advisory systems. Thus, research in this field may well be worth the effort.

7. Notes

1. In this contribution I will abstain from discussing legislation as a separate concept. I will understand legislation as the objective and final product of the legislative process.
2. E.g. problems related to the structure and the system of statute law, but also issues concerning transitional provisions etc.
3. Only a thorough analysis of the actual foregoing can lead to a more accurate definition. This explanatory contribution, however, does not allow to give an in depth analysis.
4. E.g. constitutional requirements, EEC-regulations etc.
5. In the Netherlands legislative-technique directives are being used during the departmental preparation of bills. These directives have been enacted by decree

of the Prime Minister in accordance with the council of ministers. Aanwijzingen voor de wetgevingstechniek, vastgesteld bij besluit van de minister-president, handelende in overeenstemming met het gevoelen van de ministerraad, van 14 februari 1984, Stcrt. 13 maart 1984.

6. Aanwijzingen voor de wetgevingstechniek, see supra note 5.
7. E.g. Aanwijzingen inzake externe adviesorganen, Stcrt. 6 april 1987, nr. 67, Aanwijzingen inzake terughoudendheid met regelgeving, Stcrt. 27 november 1984, nr.232, Aanwijzingen inzake openbaarheid van bestuur, Stcrt. 9 januari 1980, nr. 6 en 6 juli 1981, nr. 125, etc.
8. Like A.W. Koers, D. Kracht, M. Smith, J.M. Smits and M.C.M. van Weusten in their, *Knowledge Based Systems in Law*, Deventer/Boston 1989, p. 37, I regard knowledge-based systems as systems which solve problems by applying a symbolic representation of human expertise, instead of employing more algorithmic or statistical methods. This definition is derived from P. Jackson, *Introduction to expert systems*, Workingham 1986, p. 1. [Koers 89].
9. In the Netherlands the Department of Justice just recently established a special division to, among others, safeguard and check the quality of bills and statute laws. Divisions like this can be considered to be equipped to analyze the results of the system proposed. Stafafdeling Algemeen Wetgevingsbeleid established by ministerial decree of October 10th 1989 (226\089).

8. References

- [Ees 88] Ees, J. van, Voortgangsinformatie-systeem bij wetgeving, in: *RegelMaat*, mei 1988, pp. 71 etc.
- [Eijkern 77] Eijkern, W.J. van, De macht van de zwoegers in het vooronder, in: *De jurist-ambtenaar* (J.M. Kan-bundel), 1977, pp. 39-47.
- [Hotz 84] Hotz, R., Strukturierung des Vorverfahrens der Gesetzgebung - Erste Schritte zu einem allfälligen Einsatz von Computern bei der Schweizerischen Gesetzgebung, in: Theo Öhlinger (Hrsg.), *Gesetzgebung und Computer*, München 1984.
- [Hustinx 85] Hustinx, P.J., Aspecten van wetgeving in verband met automatisering, in: E.M.H. Hirsch Ballin en J.A. Kamphuis, *Trias automatica*, Deventer 1985, pp. 41-49.
- [Koers 89] Koers, A.W., Kracht, D., Smith, M., Smits, J.M., and Weusten, M.C.M. van, *Knowledge Based Systems in Law*, Deventer/Boston 1989.
- [Koers 90] Koers, A.W., Kracht, D., Smith, J.M., Weusten, M.C.M., Criteria for the classification of legal knowledge systems Advisory Systems on Legal Questions, in: Kracht, D., De Vey Mestdagh, C.N.J., Svensson, J.S., (eds.), *Legal knowledge based systems* (An overview of criteria for validation and practical use), Lelystad 1990, pp. 23-36.
- [Oskamp 90] Oskamp, A., *Het ontwikkelen van juridische expertsystemen*, Deventer 1990.

- [Stoyles 89] Stoyles, R.L., The Unfulfilled Promise: Use of Computers by and for Legislatures, *Computer Law Journal* 1989, Vol. IX, no.1, pp. 73-103.
- [Susskind 87] Susskind, R.E., *Expert Systems in Law*, Oxford 1987.
- [Vandenberghe 85] Vandenberghe, G.P.V., *Software-Orakels?*, Deventer 1985.

COMBINING ANALOGICAL AND DEDUCTIVE REASONING IN LEGAL KNOWLEDGE BASE SYSTEMS - IKBALS II

George Vossos, John Zeleznikow, Tharam Dillon

Database Research Laboratory
Applied Computing Research Institute
La Trobe University
Bundoora Victoria Australia, 3083
Phone: (61) 3 479 2598

Summary

Current legal knowledge based systems have generally tried to model legislation alone, emphasizing statutes and their interpretations rather than realizing the importance that precedence plays in the construction of sound legal argument. To develop truly intelligent litigation support systems, we propose a legal knowledge based architecture which permits both statutes and precedent to be accessed through the use of analogical and deductive reasoning. We discuss these two forms of legal reasoning and their corresponding implementations using the IKBALS II prototype. IKBALS II is an object oriented litigation support system allowing the lawyer to develop supporting arguments for his or her case using current statutes and precedence. The legislation used is the Victorian (Australia) Accident Compensation Act, 1985.

1. Introduction

Legal reasoning is an intellectual process by which lawyers and judges use cases and rules to solve legal problems. Legal practitioners primarily combine two forms of reasoning when dealing with litigation: reasoning by analogy and reasoning by deduction. This paper describes these two forms of legal reasoning in the context of building Legal Knowledge Base Systems (LKBS) or Expert systems (ES). The aim is to build an automated litigation support system to allow lawyers to intelligently retrieve and analyze the sources of the law, namely statute and case law, so as to successfully argue their clients case in court.

1.1. Motivation for commitment to LKBS

The use of the computer to assist lawyers, judges, and the courts themselves, is not new. Important applications of computers in the legal profession include time recording, billing and document processing systems. Countries like Australia have installed computerized networks in some of their courts (Magistrate Courts) to help judges with the day to day administration of justice. Other countries have introduced computerized systems which allow judges to receive assistance with sentencing by retrieving a list of recent sentences handed down in some previous similar cases.

The major use of such computers in law currently however, has been that of *legal information retrieval* (also referred to as *indexing* of legal text). These systems are usually on-line databases containing text in the form of legislation, precedent cases and reports. [Bing 86] describes the state of legal retrieval systems.

While such computerized tools can be helpful to lawyers for the retrieval of legal material, many believe that they are not very useful in litigation support [Martino & Socci Natali 86]. That is, the use of boolean queries is not sufficient to express concisely, or express at all, the query of the lawyer. As a result, searches deliver an excess of irrelevant documents or fail to retrieve the bulk of those relevant texts that are in fact stored within the database. [Gordon 88] reports on the difficulties of applying boolean query languages to legal retrieval.

There has therefore been a gradual appreciation by many in the field that it is now necessary to attempt to develop computer systems in law that can be said to embody *knowledge*, that is, exhibit intelligent behaviour. Such systems could help lawyers with the generation of plausible arguments for their case.

With the development of analogical and case based reasoning in law, [Ashley & Rissland 88] and [Riesbeck & Schank 89], software tools are only now becoming available to make it worthwhile to consider developing automated legal reasoning systems to help with litigation support. The books of [Riesbeck and Schank 89] and [Kolodner 88] describe numerous legal case based reasoning systems. JUDGE [Bain 86], works in the domain of criminal sentencing attempting to model a judge who is determining sentences for people convicted of crimes; HYPO [Ashley & Rissland 88]; [Ashley 90], does case based reasoning in the area of patent law generating plausible arguments for the prosecution or the defence; and PERSUADER [Sycara 87], proposes resolutions for dispute situations.

The authors of this paper support using case based reasoning techniques like the one proposed by [Ashley & Rissland 88] in attempts to model the highly qualitative legal process. Furthermore, they believe that it is essential that any intelligent legal reasoning system has the ability to couple the reasoning methods employed by lawyers; namely combining deductive reasoning (rule based systems) with analogical reasoning (case based reasoning) systems. Such facilities are necessary in order to allow legal practitioners to intelligently access and analyze statutes and past precedent cases.

1.2. Current LKBS

Until recently, most attempts at building LKBS have relied on using the standard deductive ES architecture, consisting of a knowledge base, an inference engine and a user interface. They are similar to those systems used for modelling intelligent behaviour in highly quantitative domains such as medicine, finance, manufacturing and oil exploration. These legal ES have relied on "if-then" production rules [Waterman & Peterson 86], [Schlobohm & Waterman 87], or logic [Sergot et al. 86], as the basis of knowledge representation. Although modest systems have been built using the standard ES architecture [Suskind 87], such systems have failed to gain the use and support of legal practitioners. A major reason for this is the highly qualitative nature of law. From the point of view of building LKBS, one must be prepared to supplement deductive reasoning with other forms of reasoning, namely analogical, temporal, and fuzzy reasoning in order to effectively deal with the "open-textured" nature of the law. These reasoning processes allow the lawyer to:

- i) locate relevant norms from a piece of legislation (deductive reasoning);
- ii) retrieve relevant precedent cases (analogical reasoning);
- iii) allow amendments to be easily incorporated in the knowledge base: including the time when such amendments were passed (temporal based knowledge);
- iv) give probabilities that a certain line of legal reasoning will be successful in court (fuzzy reasoning)

Hence, not only are most legal information retrieval systems of minimal utility to lawyers, but so are most LKBS because of the mere fact that their architecture is not flexible enough to allow the type of hybrid reasoning needed to aid lawyers with litigation.

1.3 Incentives for Developing IKBALS II

Legal practitioners are in general not solely interested in the likely outcome of a court case. They are more concerned with providing support to argue the case that their client wishes to put forward, although admittedly they do encourage their client to pursue a path that has a reasonable chance of success. Hence what is really needed is a litigation support system which will allow the lawyer to navigate through the vast amount of legal sources available, permitting him to find the relevant rules and precedent cases to successfully argue his client's case.

Instead of having the LKBS drive the consultation, the system needs to be able to advise the lawyer on the probable outcome of pursuing each possible course of action (hence the need for fuzzy reasoning). The system also needs to take into account the fact that statutes regularly change and hence it will be necessary to develop knowledge bases that have a temporal aspect.

Such a system will be able to:

- (a) identify the relevant legal norms;
- (b) identify the relevant precedent cases for the particular dispute at hand, in a reliable efficient manner, from a very large number of possible cases;
- (c) having identified the relevant precedent cases, compare it with other cases in the system, and in particular, distinguish the current case from others whose conclusions run to the contrary;
- (d) suggest arguments which can be made in favour of the current case and the facts and precedents which can be cited to support them.

IKBALS II is an attempt at providing such facilities to lawyers.

2. Using Cases and Rules

A legal rule is an abstract or general statement of what the law permits or requires of classes of people in classes of circumstances. A case on the other hand represents a short story of an incident in which the state acted or may act to settle a particular dispute [Burton 85]. Expression of the law take the form of both rules and their interpretation and cases (particular instances of rules).

To persuade a court of what it should do in a current case, a lawyer points out what courts have done in other, similar cases. The practice of comparing and contrasting cases is seen to be more advantageous than using rules, as cases supply particularities that general rules leave untreated [Levi 48].

3. Forms of Legal Reasoning

In Australia Parliament drafts the laws and the courts interpret the laws with the assistance of the common law when appropriate. The courts also interpret the law where parliament has left a gap, e.g., legislation re murder under the Crimes Act is relatively short, but the precedent base and literature about it is voluminous. Hence lawyers when considering a particular dispute normally try to find all the relevant cases to that particular dispute. They then extract the applicable principles or interpretations from these previous precedent cases in order to form a judgement as to whether to proceed with the dispute. If so, they then develop an argument in support of their position in the dispute and anticipate counter arguments and lures of arguments supporting the other party. This is how lawyers reason analogically.

Some means of organizing the overwhelming mass of legal material a lawyer must process when reasoning about a case is essential. Generally speaking, it is difficult to express such knowledge simply as production rules. It appears therefore that in modelling legal reasoning, one requires the doctrines of the law (e.g. Habeas Corpus, Mens Rea, negligence); the particular statutes; and the legal arguments and particular facts contained in precedents.

Whilst there might be some chance that rule based or logic based systems could capture some of the elements of the statutes, it is highly unlikely that they will be able to capture the remaining requirements. This has led to our interest in IKBALS as a means of dealing with the subtlety and complexity inherent in legal reasoning problems by experimenting with analogical reasoning.

3.1. Reasoning by analogy in LKBS

In order to incorporate analogical reasoning into a LKBS, the process by which the system will operate must be understood. A working analogical reasoner would be expected to (a) retrieve similar cases, (b) choose the best case(s), and (c) use the solution in the current problem in some way. What makes the task very difficult is that the underlying architecture must be based on a conceptual model of legal analysis. A conceptual model will "understand" concepts of a particular area of law. In contrast, most legal information retrieval systems only aim at retrieving legal sources based on boolean key word matching.

One strategy for IKBALS II, uses a conceptual clustering algorithm to classify the different precedent cases into clusters, given the data about these cases [Tyree et al. 87]. The central principle used is that the cases which are close to one another should be included in the same cluster. The efficiency of this method will crucially depend on:

- i) The set of attributes selected;
- ii) The chosen measure of closeness;
- iii) The algorithm of forming clusters out of sets of cases, given the data on their distances.

3.2. Reasoning by Deduction in a LKBS

Most current LKBS reason by deduction. These systems use a production system formalism, where rules can be seen as a series of antecedents which are linked to a consequent. If all the antecedents are "true" then it logically follows that the consequent is "true". These production rules are in a form where the conclusion represents some

legal concept and the condition represents all the important facts which are required in order for the legal rule to be applicable.

There are essentially three steps in legal reasoning by deduction:

- i) Identify the legal rules that plausibly may govern the case at hand. This is often referred to as formulating the major premise;
- ii) Formulate a minor premise in the language of the major premise. The problem here is that the facts in any case can be described in a variety of terms;
- iii) Combine the second and third steps to package the facts as a minor premise and to use the premises to yield a conclusion through deductive reasoning.

4. Combining the Two Forms of Legal Reasoning - IKBALS II

Because of our desire to develop a real life prototype dealing with the complex issues of combining analogical and deductive legal reasoning, the authors of this paper have used the IKBALS prototype [Vossos et al. 90a] and [Vossos et al. 90b] to extend its present hybrid object-oriented/rule based architecture to handle case based reasoning; IKBALS II. IKBALS II deals with applications for compensation under the Accident Compensation Act 1985 (WorkCare). In particular, the system advises injured workers on the likelihood of a successful application for compensation, be it for weekly payments, lump sum entitlements, or medical and like expenses. In the following sections, we illustrate the principles mentioned above by discussing the deductive module and the analogical reasoning module of IKBALS II, respectively.

4.1. Modelling of The Act

It is possible to delineate clear stages in dealing with claims under WorkCare. These are:

- Stage 1 Elements giving rise to a WorkCare entitlement
- Worker
 - Injury
 - Whether Injury was in the course of employment
- Stage 2 Is the claim under the jurisdiction of the Act
- Stage 3 Statutory Entitlement
- Weekly payments for total incapacity
 - Compensation for partial incapacity
 - Lump sum compensation for injury to limbs
 - Lump sum compensation for industrial deafness
 - Lump sum compensation for dependents of a deceased worker
 - Compensation for medical and like expenses
- Stage 4 Advising the client
- Interim financial relief
 - Make-up pay
 - Social security entitlement
 - Common Law claim
 - Legal costs

- Stage 5
- Weekly payments disputed - the Arbitration Process
 - Options where genuine dispute found
 - Genuine dispute not found

In order to exploit the reasoning structure used by the expert¹, IKBALS used a hybrid object oriented/rule based approach employing a lattice of classes and objects. These lattices consist of classes connected in a fashion which supports direct inheritance and multiple inheritance. They were then combined with rule and meta-rule sets, making it possible to reason deductively with the legislation. The inference strategy used by IKBALS is goal-directed forward chaining; a combination of forward and backward chaining. Furthermore, the encapsulation property of the object oriented paradigm permits one to cope with the problem of maintenance of the knowledge base that arises from statute and regulation changes. For an outline of the benefits of automating legal reasoning using an object oriented knowledge based approach, as well as a transcript of a sample consultation, refer to [Vossos et al. 90b].

To summarize, the hybrid object oriented/ rule based representation:

- (a) captures the natural knowledge structure of the problem,
- (b) makes for easier maintainability, and
- (c) makes for efficient execution through limiting the attention of search to within a particular object's rule set.

4.2. Modelling of Precedent

Efforts to build serious LKBS will be of no use unless they are able to reason with precedent. In order to reason with case law, each case must be represented in such a way as to facilitate the identification of that case, i.e., "indexing". More precisely, what is needed is an intelligent retrieval process which will identify similar "target" cases to a particular current "source" case. Our technique extends the object oriented/rule based architecture described above to incorporate analogical reasoning.

The importance of knowledge representation cannot be underestimated. In order to implement case based reasoning, it is important to structure the key elements of the case at hand in the knowledge base so as to facilitate identification and retrieval. Our technique involves the use of object/class structures with intersecting decision lattices.

Faster indexing of similar cases is facilitated by classifying resolved cases under as many case types as possible in order to maximize the range of applicable precedents. In particular, resolved cases are indexed by pta's. Each pta represents a way of arguing about a case. Points To Argue are the subset of facts of a case that the court deemed significant in determining the case's outcome; that is, the subset of facts that were relied upon in identifying the significant legal principle in question.

Currently, problems are input in a form, based on what the knowledge engineers have deemed 'important' attributes of the case. Note, that the choice of these attributes is quite crucial to the success of the system. When the current legal problem requiring a solution is input, a rule based engine determines which ptas apply to the current case. These ptas are then used to retrieve those cases that are indexed under the same ptas in the system. The pta's prerequisites determine what features to look for in a case. Since a case can be indexed under several different ptas, after a number of candidate resolved cases have been retrieved, the most similar case(s) is chosen. For each of the

most relevant precedents, IKBALS then proceeds to justify that the outcome of the current case should, or shouldn't be, the same as the precedent's outcome by drawing the analogy between them focussing on their important similarities and differences. This approach is similar to the one proposed by [Ashley 88].

5. Conclusion

In this paper, we have shown that it is important that LKBS act as litigation support systems rather than the more conventional judgement systems that generally only aim at interpreting statutes using a simple production rule representation. To offer genuine litigation support, a LKBS must allow arguing with precedents. We have hence shown how to incorporate the type of analogical reasoning employed by lawyers when reasoning with past cases by introducing case based reasoning into our object oriented prototype IKBALS. IKBALS is a LKBS that is supporting deductive and analogical reasoning. We then proceeded to discuss in detail how the two forms of legal reasoning could be combined by extending IKBALS's hybrid object oriented/rule based architecture. We also discussed in detail how past cases are indexed by using points to argue (ptas).

6. Notes

1. The IKBALS project uses as its domain expert Graeme Taylor. For the purpose of this project, Taylor helped in developing the knowledge structure of IKBALS. The model was based on what he considered to be the legislation's purpose.

7. References

- [Ashley 88] Ashley, K. D. (1988), "Arguing by Analogy in Law: A Case Based Model", *Analogical Reasoning*, Kluwer Academic Publishers, pp. 205-224.
- [Ashley 90] Ashley, K. D. (1990), *Modelling Legal Argument-Reasoning with Cases and Hypotheticals*, Bradford/MIT Press.
- [Ashley & Rissland 88] Ashley, K.D., and Rissland, E.L. (1988), "A Case-Based Approach to Modelling Expertise", in: *IEEE Expert*, Fall, pp. 70-77.
- [Bain 86] Bain, W. M. (1986), *Case Based Reasoning: A Computer Model of Subjective Assessment*, Ph.D. thesis, Yale University.
- [Bing 86] Bing, J. (1986), "The Text Retrieval System as a Conversion Partner", *Yearbook of Law, Computers and Technology*, Volume Two.
- [Burton 85] Burton, S. J. (1985), *An Introduction to Law and Legal Reasoning*, Little, Brown and Company.

- [Gardner 87] Gardner, A. (1987), *An Artificial Intelligence Approach to Legal Reasoning*, Bradford/MIT Press.
- [Gordon 88] Gordon, M. (1988), "Probabilistic and Genetic Algorithms for Document Retrieval", *Communications of the ACM*, October 1988, Volume 31, Number 10. pp. 1208-1218.
- [Kolodner 88] Kolodner, J. (1988), *Proceedings of a Workshop on Case Based Reasoning*, May 1988, Clearwater Beach, Florida.
- [Levi 48] Levi, E.H. (1948), *An Introduction to Legal Reasoning*, The University of Chicago Press.
- [Martino & Socci Natali 86] Martino, A. A., Socci Natali, F. (1986), *Automated Analysis of Legal Texts- Logic, Informatics, Law*, North Holland.
- [Riesbeck & Schank 89] Riesbeck, C. K. and Schank, R. C. (1989), *Inside Case Based Reasoning*, Lawrence Erlbaum Associates Publishers.
- [Schlobohm & Waterman 87] Schlobohm, D.A. and Waterman, D.A. (1987), "Explanation for an Expert System that Performs Estate Planning", *Proceedings First International Conference Artificial Intelligence and Law*, ACM New York, NY.
- [Sergot et al. 86] Sergot, M.J., Sadri, F., Kowalski, R.A., Kriwaczek, F., Hammond, P. and Cory, H.T. (1986), "The British Nationality Act as a Logic Program" *Communications ACM*, Volume 29(5), pp. 370-386.
- [Susskind 87] Susskind, R.E. (1987) *Expert Systems in Law*, (A Jurisprudential Inquiry), Clarendon Press, Oxford.
- [Sycara 87] Sycara, E. P., (1987), "Resolving Adversarial Conflicts: An Approach to Integrating Case Based and Analytic Methods", *Technical Report GIT-ICS-87/26*, Georgia Institute of Technology, School of Information and Computer Science, Atlanta, GA.
- [Tyree et al. 87] Tyree, A. L., Greenleaf, G. and Mowbray, A., (1987), "Legal reasoning: the problem of precedent", *Proc. Conf. AJAI*, Sydney, November 1987, pp. 419-432.
- [Vossos et al. 90a] Vossos, G., Dillon, T., Zeleznikow, J., Taylor, G. (1990), "An Object Oriented System for Legal Reasoning - IKBALS", *Proceedings of the Tenth International Workshop on Expert Systems and their Applications*, pp. 741-754, Avignon, France, 1990

- [Vossos et al. 90b] Vossos, G., Dillon, T., Zeleznikow, J., Taylor, G. (1990), "The Use of Object Oriented Principles to Develop Intelligent Legal Reasoning Systems", to appear in *Australian Computer Journal*.
- [Waterman & Peterson 86] Waterman, D.A., Paul, J. and Peterson, M. (1986), "Expert Systems for Legal Decision Making", in: *Applications of Expert Systems*, The Rand Corporation.

THE EVOLUTION OF RESEARCH AIMS

Jaap de Wildt, Aernout Schmidt, Jeannette Quast,
Jaap van den Herik, Robert van Kralingen, Pepijn Visser
and Wim de Vries
PO Box 9521
2300 RA Leiden
The Netherlands

Summary

In this paper we investigate the evolution of aims for application-related legal-AI research at the Leiden University. The initial goals are reconsidered and their evolution is viewed with respect to the aims of other research projects. It is concluded that after a period of rather holistic research approaches with attempts to create complete legal knowledge-based systems, now more specific research goals have arisen. The potential conceptualization of these research goals coincides with a better understanding of the domain structure. This, in turn, may subsequently lead to a fruitful cooperation among researchers.

1. Introduction

The research on legal knowledge-based systems (LKBSs) has two mainstream sets of goals. There is a practical set of goals and a theoretical one. The practical set of goals aims at the creation of computer programs which can handle the legal knowledge adequately. These programs are based on formal representations of legal knowledge. The type of research involved shows how unruly legal knowledge usually is and how difficult it is to formalize statute law and case law.

In contrast to these practical goals, we have the theoretical goals which emphasize the characterization of legal knowledge. This type of research attempts to reveal the nature of legal knowledge and legal meta knowledge. As one might have expected, practical and theoretical goals are interrelated: practical goals raise theoretical questions and *vice versa*.

An almost ironic instance of such a theoretical question posed by a practical goal occurred in the Netherlands in 1986. Shortly after a revision of the social-security legislation had passed Parliament, a public request was issued by an MP (Member of Parliament) urging the development of a computer program able to explain the meaning of the new legislation. Since the generic nature of the new legislation left much to be resolved by case law, no one ventured to produce such a program. However, this request raised the fundamental question whether the new legislation should be made explicable by computer scientists instead of legislative authorities.

The MP's request was inspired by the success of a 1986 computer program for the Student-Funding act. The program, constructed shortly after a change to the act had been made, was meant to explain the rules, which were unintelligible for students as well as for administrators. Moreover, since 1986, the Student-Funding Act has been adapted several times and remarkably the administration has consistently failed to implement these adaptations in time.

Another case of failure occurred in 1989 when the Auditor's Office produced a report on the construction of information systems for the Dutch police. It turned out

that only very few of the systems initiated during the last 15 years had reached a (sometimes partially) operational stage.

Allowing for the successes accomplished so far, we remark that the software crisis in the 1970s has become a major obstacle for many law-dependent applications. Unfortunately, this handicap still exists and in this time of object-oriented modelling one may wonder why.

It is sometimes argued that the failures to produce the law-dependent applications mentioned above are incidental and the result of a vicious combination of mismanagement and incompetence. However, we believe that such a reasoning does not make a point. It is rather improbable, assuming that anyone involved in these projects is ambitious and competent. Therefore our hypothesis reads that the current state of the art in information-system-design methodology is not (yet) fit to model the many law-dependent information procedures.

If this hypothesis holds, the expectation of producing a substantial legal knowledge-based system in a single research project by a few researchers is too optimistic. Nevertheless, we admit, this is exactly the type of optimism which guided our research programs in 1985. Considering this analysis of our initial aims for LKBS research, we propose a new approach of research.

In the next section, we first examine our initial aims in relation to the aims of other researchers, and show how our aims have evolved (Section 2). Then we support our current view on the aims of the legal-AI research with the description of relevant projects (Section 3). Finally, we present our conclusions in Section 4.

2. Goals in legal knowledge-based-system research

In this section we investigate the practical and the theoretical aims of some LKBS-research projects. First, we have attempted to identify the aims and to classify them into practical and theoretical aims (Section 2.1). Subsequently, three projects from Leiden are treated accordingly (Section 2.2). Then we compare our research aims with those of the other projects (Section 2.3). Finally, we conclude with some tentatively formulated remarks expressing some optimism (Section 2.4).

2.1. Related projects

We have selected the Dutch research projects which have resulted in actual systems, such as JURICAS, TESSEC and PROLEXS. In addition to the Dutch projects we have examined a few well-known Anglo-American systems, such as HYPO, the British Nationality Act and the Latent Damage System. We believe that these six projects provide a representative context for our analysis.

- The practical aim of the JURICAS project has been the development of a shell for legal-advice systems (De Mulder *et al.*, 1989). The system's database primarily contains a decision tree supplied with relevant texts, such as parts of the legislation, abstracts of regulations, or descriptions of cases. The database is filled by a domain expert. The system's users should have a legal education. The JURICAS project has been launched with the prominent question whether it is currently (say in the 1980s) possible to build a general shell for legal knowledge-based systems. The *present* JURICAS researchers believe this not to be possible due to a lack of knowledge about legal decision-making processes in general. Their current (possibly intermediate) theoretical aim is to obtain empirical know-

ledge on legal-decision making. They prefer to use jurimetrical methods.

- M.A. Nieuwenhuis' (1989) practical aim was to build a legal expert system (TESSEC) that improves the quality of decisions of the Social Welfare administration. One of the additional practical aims was to develop a knowledge-representation language which adequately represents statute law with respect to Social Welfare legislation. As a result, the TESSEC project also raised some theoretical questions related to the consistency of regulations and the distinction between well- and ill-structured knowledge.
- The aim of the PROLEXS project is twofold: (1) to develop a shell for building legal expert systems and (2) to develop a methodology for building a legal expert system with the help of this shell (Oskamp *et al.*, 1989). The first aim can be considered practical, the second theoretical.
- HYPO was intended to be a system partially performing case-based reasoning in Anglo-American common law (Rissland and Ashley, 1987). A theoretical aim was to develop a theory that provides computational definitions with respect to the impact of precedents in a legal argument (Ashley, 1989).
- The aim of THE BRITISH NATIONALITY ACT EXPERIMENT (Sergot *et al.*, 1986) was to test the suitability of PROLOG logic when expressing and applying legislation. The experiment highlighted two problems: (1) the negation as failure (anything which is not known is assumed to be false) and (2) the counter-factual conditions. It was argued that in legal reasoning negation as failure is inappropriate, because it is unreasonable to make a Closed World Assumption. Moreover, direct use of PROLOG logic is not adequate for dealing with counter-factual conditions.
- Finally, we mention Susskind and Capper's research, which resulted in THE LATENT DAMAGE SYSTEM. Susskind's theoretical aim can be described as the investigation whether theoretical or practical obstacles to the development of rule-based expert systems exist in the nature of law or in the processes of legal reasoning (Susskind, 1986). In cooperation with Capper his practical aim was to build a rule-based expert system, which could enter the market (Capper and Susskind, 1988).

In summary, the listed projects aim at the creation of computer programs in which legal knowledge is represented. The programs should perform some kind of legal reasoning. Several theoretical questions on the constitution of legal reasoning result from these aims.

Although we here have described six projects only, many more have been considered (cf. Van den Berg *et al.*, 1988). We found that most research projects started around 1985 aimed at "complete" solutions for implementing legal expertise in knowledge-based systems. The projects often resulted in the specification of less general aims for future research.

2.2. Leiden

At the Leiden Faculty of law, legal-AI research is performed by the Department of Law and Computer Science (since 1985) and by the Research Group on Expert Systems and Social Security (JUS, since 1986). Both groups closely work together. Their history is short and confluent; the aims of their initial projects are easy to

identify (cf. Schmidt *et al.*, 1988). We confine ourselves to the relevant aims that resulted in prototypes of knowledge-based systems and to the questions raised in this research.

In Leiden, three different prototype systems have been developed since 1985: PEM, LEIDRAAD and CAMO. As these systems have been described extensively elsewhere (Schmidt, 1987; Quast and De Wildt, 1988; Visser *et al.*, 1990, respectively), here we provide an outline only.

- PEM is a knowledge-based system profitably used as a vehicle in Schmidt's Ph.D. thesis. Its practical aims were (1) to show that legal-policy changes expressible in terms of criterion values can be adequately implemented by legal-expert users and (2) to show that simple heuristics based on these criterion values may support the legal-expert user to keep his case base consistent (or to pinpoint inconsistencies in empirically-constructed case bases).

PEM's theoretical aims were to investigate to what extent professional-legal knowledge on probation can be formalized efficaciously. A detailed analysis showed that the formalization of inarticulate knowledge cannot be performed satisfactorily. Moreover, as a typical area of inarticulate legal knowledge, the knowledge required to reduce or enlarge the current scope for policy making was identified. This type of knowledge is assumed to be important when criterion values are changed or when legal rules are reformulated.

The main question raised by the PEM research project was: which means are to be provided to the legal-expert user in order to handle unsteady ruling, meanwhile keeping the case base consistent?

- LEIDRAAD is a prototype knowledge-based system developed by Quast and De Wildt in 1988. LEIDRAAD's practical aims are to show that a knowledge-based system can advise decision makers effectively on complex social-security cases.

LEIDRAAD's first theoretical aim was to investigate to what extent case law can be formalized. The researchers mainly concentrate on modelling legal knowledge in (truth-conditional) predicates and weighing functions. The second theoretical aim has been the method of how to adapt the weighing functions. Furthermore, maintainability was a necessary condition for the development of LEIDRAAD. This condition yielded a third theoretical aim: the definition of a specification language understandable by a legal expert, especially when implementing changes in the knowledge base.

Three main questions raised by LEIDRAAD are: (1) can the distinction between predicates and functions be related to legal theory? (2) can the specification language be ameliorated in such a way that maintenance can be performed by the legal-expert user? and (3) will the quality of the knowledge base improve by adding an incrementally-growing, consistent case base? (cf. De Wildt and Quast, 1989).

- CAMO is a knowledge-based system developed by Visser *et al.* (1990). CAMO's practical aims stress the use of a KBS for legal experts to store and retrieve relevant case law.

CAMO's theoretical aims are (1) to investigate whether a combination of hierarchical structures of conflicts and characteristics support an appropriate modelling of case law and (2) to examine whether a simple heuristic will keep the case base consistent.

The experiments showed that various users tend to create different, but possibly equipollent models. From these experiments two questions resulted. The first question was: how to identify the equivalence of the different models? The se-

cond question raised originated from the analogy between case models and the structure of domain knowledge. This question reads: what is the relation between hierarchical models (primarily generated to store cases in a consistent way) and the system's domain knowledge?

Our research has treated the question in which situations numerical (or statistical) methods are appropriate for the representation of legal expertise (Schmidt and Van den Herik, 1989). Previously, the PEM study had shown that whenever legal knowledge is not static, statistical methods are inadequate to describe legal processes. The study claimed that statistical models are only appropriate for otherwise inarticulate *evaluative* legal reasoning (Schmidt, 1987). Approximately at the same time, the LEIDRAAD study proved that numerical methods are rather suitable for the description of decision processes, in which judges weigh arguments pro and contra (De Wildt and De Jong, 1990). However, for an adequate statistical analysis of legal decisions there are some restrictions. First, the facts relevant for a decision, must be identifiable. Second, a significant number of decisions is necessary to gain reliable results. And third, the decisions must be consistent over a longer period.

In the Leiden research program the projects are characterized by open norms, vagueness and open texture. They focus on the maintainability of knowledge bases. The PEM system enabled the user to adjust the knowledge base after a change in policy. The CAMO system offered a maintainance heuristic when legal policy (of a court) had changed. The LEIDRAAD project concentrated on the realization of a knowledge base in which a legal concept was defined by sub-concepts and sub-concepts by sub-sub-concepts, etc. Such a conceptual tree is transparent and supports the classification in social-security law. Moreover, it supports maintenance. Although such a structure is not capable to represent all legal knowledge, it has been a basis for the implementation of truth-maintenance heuristics in the PEMBA system.

As the other projects mentioned above, the Leiden research program aims at the accomplishment of computer programs in which legal knowledge is represented and used. Two of the earlier Leiden research projects (PEM and LEIDRAAD) aimed at providing "complete" solutions, i.e., they considered the implementation of legal expertise in knowledge-based systems. Finally, this resulted in the specification of less general research aims. Understanding the cause of this failure, the CAMO project aimed at a specific research question from its start.

2.3. *Evaluation*

In this section we evaluate the results of the aims described above. We distinguish between statute-based and case-based systems.

Although it is possible to build legal knowledge-based systems based on statute law, these systems will at least be controversial in their treatment of negation, counterfactuals, vague norms as well as open texture. The controversy is due to differences of opinion within the professional legal forum (cf. Leith, 1986). The discussion concentrates on well-structured knowledge as opposed to ill-structured knowledge. For vague norms and open texture, M.A. Nieuwenhuis (1989) developed a special-purpose knowledge-representation language. Since we do not know anything about the structure of intuitive legal knowledge we refrain from classifying it into the well-structured or ill-structured knowledge class. Finally we remark that the work of Sergot *et al.* (1986) emphasizes the possibilities of PROLOG logic for the representation of statute law. Their results are confirmed by the Leiden research.

In case law, analogous problems on the structure of legal knowledge arise. The HYPO system can be considered as a pure case-law system. Rissland *et al.* (1987) demonstrated that representing cases with frames is an adequate technique for storing cases. Furthermore, Ashley (1989) developed a computational theory about most persuasive and most salient precedents, which is especially relevant for developing case-law systems in Anglo-American law.

Although in Anglo-American law precedents play obviously a more important role than in continental law, this difference is not significant enough to allow Dutch research to ignore case-based reasoning. As pointed out by Bing (1990), there is a high degree of similarity between (Anglo-American) case-based reasoning and (continental) rule-based reasoning. Consequently, the Leiden research program aims at combining both types of legal reasoning.

As PEM, LEIDRAAD and CAMO, the PROLEXS project is inspired by research related to both statute-law systems and case-law systems. Various types of knowledge are linked with different kinds of representation formalisms. In the prototype PROLEXS system the choice of a representation formalism is dependent on the characteristics of the knowledge source. For instance, statutes are represented by rules (M.A. Nieuwenhuis, 1989 and Sergot *et al.*, 1986) and case law is represented by frames (cf HYPO).

In our view, however, the problems with case-based systems do not relate to the pro's and cons of representational formalisms, but are a result of the accepted existence of *inarticulate, intuitive* legal professional knowledge: most legal experts are able, for instance, to assess the relevance of matching case elements to legitimate the use of analogous reasoning and to pinpoint inconsistencies among cases. However, an articulate and objective description of this skill does not seem to be available.

If there is any consensus at all amongst legal experts regarding this type of skill, it concerns the impossibility to describe it formally and objectively (J.H. Nieuwenhuis, 1976; Mac Cormick 1978). As a result case-based systems, including the Leiden projects, use heuristics as *hypotheses* for the description of legal skills regarding analogy and inconsistency among cases. In other words: we believe that case-based systems can be built as statute-based systems can. However, the case-based systems will be controversial in their treatment of cases. In the research years to come we must address the *why* and the *how* of the similarity of cases as well as the inconsistency of a collection of cases.

2.4. Optimism

In 1987, however, Susskind concluded that there is no element in the nature of law or in the process of legal reasoning that constitutes a theoretical or practical obstacle to the development of rule-based expert systems of limited scope: "In the short term it will be possible to build systems that will solve clear cases." He continued "In the long term, assuming that human legal experts are willing to make their expertise available to knowledge engineers, expert systems will be built to solve deductive cases, they will have heuristic knowledge in their knowledge bases, will reason with uncertainty, and will draw probabilistically phrased conclusions." (Susskind, 1987). The conclusion of De Mulder *et al.* (1989) that there is need for empirical (jurimetrical) research seems to be compatible with the last two statements. As most members of the legal-AI research community, Susskind and De Mulder are optimistically inclined towards the ultimate solvability of the problems mentioned earlier.

In the problems discussed above, we have faced again a few well-known issues of legal AI as well as legal philosophy. The issues have proved to be obstinate ones from the philosophical point of view. (Legal-philosophical disputes on negation, counterfactuals, vague norms, open texture, objective evaluation and intuition have a long history and a promising future.) From the legal-AI perspective, these problems would be solved by providing valid formal models of the notions and processes involved. We have seen that the legal-AI community is optimistic with regard to the attainability of their aims. To supply our own opinion: some of the authors of this paper are as optimistic as the ones cited, others are less optimistic. Nevertheless sound optimism carries good research and this optimism implies that we either expect to be able to solve (some of) the problems mentioned by successful research, or that we can assume that (some of) the problems are merely theoretical and have no practical significance.

3. Structures in the legal-AI research domain

Since we abandoned the idea of solving all problems for various legal knowledge domains in a single research project, we need additional structure in the research domain allowing (a) the identification of existing legal AI knowledge and (b) the identification and incorporation of fresh contributions. To obtain this structure we have applied three commonly-used research approaches. The first one is to indicate stages on the road from legal expertise to legal knowledge-based computer applications (Section 3.1), the second is how to discriminate different types of legal knowledge and legal reasoning (Section 3.2) and the third is how to categorize domains of legal knowledge with respect to legal content (Section 3.3). We also provide a summary description of these approaches which enables us to compare the aims of our current research projects to these approaches.

3.1. Milestones in expert-system development

The development of expert systems in general provides a link between a source system (consisting of a client in dialogue with a legal expert) and a target system (consisting of a user in dialogue with an expert system). Several transformations are to be dealt with: (1) from source systems to *conceptual models* (by knowledge acquisition), (2) from conceptual models to *formal models* (by formalization), (3) from formal models to *technical models* (by implementation) and (4) from technical models to the *target system*.

The identification and incorporation of fresh contributions to AI knowledge will obviously benefit from the use of standard modelling languages. However, due to methodological differences, a multitude of modelling languages is available. We have categorized a few of them in a crude, illustrative fashion. Modelling languages for conceptual models may be provided by knowledge-acquisition tools such as SHELLEY (Anjewierden, 1990) and MOLE (Eshelman, 1988). More formal modelling languages may be provided by tools such as DDL (Schreiber *et al.*, 1989) or DESIRE (Kowalczyk and Treur, 1990), while design languages may for instance be provided by object-oriented tools, such as SMALLTALK (Cox, 1986).

We here present our current research goals in such a way that its intended contributions to the available AI knowledge are clear. Therefore, we have made a selection from the available modelling languages and methodologies.

For *conceptual models* our fundament will be adopted from the KADS methodology (Breuker *et al.*, 1987 and De Greef, 1988). The current project (by Van Kralingen and Visser) on legal knowledge representation aims at (1) the validation of available knowledge sources, meta classes and interpretation models and, if necessary, at (2) the contribution of new ones. (This project is sponsored by NWO, the Netherlands Organization for Scientific Research). The same approach is valid for our new projects on the modelling of the legal experts' knowledge on interaction with a client (De Vries), on the modelling of the legal experts' heuristic knowledge (Quast), and for our new joint research project with IBM (called CHIPS) on the modelling of legal tuition (Van den Herik, Schmidt, Van der Klaauw and Van Ginkel). The selection of the KADS methodology has not been made for all stages of the research projects mentioned above. As it stands, we expect some major problems with the strategic level and with the model of cooperation. However, the domain-, inference- and task levels seem to present a solid foundation and the level of acquaintanceship with KADS in the AI community guarantees at least a collective effort to remove deficiencies.

For *formal models* we intend to use the DESIRE framework as our initial standard. The current NWO project aims at the construction and validation of generic task models for legal applications, probably using existing generic task models (e.g., SIX). The same approach is intended for our new projects on the modelling of the legal experts' knowledge on interaction with a client, on the modelling of the legal experts' heuristic knowledge, and on legal tuition. In addition, the PEMBA system (a spinoff from the PEM project) has been set up for providing a LKBS shell and a technical modelling environment. PEMBA is now redesigned to support the implementation of formal models specified in DESIRE.

The selection of DESIRE is based on its well-defined syntax and semantics, on the analysis of a few experiments and on the awareness of its possibilities as an extension of the KADS knowledge-acquisition methodology as shown by the DESIRE initiators (Kowalczyk and Treur, 1990).

Since a contribution to the domain of technical AI knowledge is not one of our research aims, we will not elaborate on this aspect. The PEMBA project (Schmidt, Van Ginkel and computer-science students) simply aims at providing the *technical language* and implementation tools for the validation and use of formal models developed in the research projects mentioned.

3.2. *Types of legal knowledge and legal reasoning*

A second approach that aims at the structuring of the legal-AI research domain is through a generic typology of legal knowledge. We have selected the typology of accepted legal argumentation, as presented by J.H. Nieuwenhuis (1976). Nieuwenhuis' typology distinguishes (1) logical inferences, (2) specification for making distinctions (refinement), (3) abstraction for making analogies, (4) "a contrario" arguments and (5) comparing (hypothetical) cases. He has shown that these types of argumentation are acceptable in Dutch legal theory, especially with regard to the persistent philosophical problems mentioned earlier. Nieuwenhuis argues that these types of argumentation are related to the heuristic procedures used by legal experts when formulating their solutions to legal problems. However, neither Nieuwenhuis, nor any other legal scientist has provided explicit descriptions of the heuristic procedures implied by types (2) - (5), although some attempts are worth to be mentioned (Crombag, De Wijkersloot and Cohen, 1977).

Many legal AI researchers hope to provide more explicit and still valid descriptions

of these heuristic processes. As a matter of fact, these contributions aim at enhancing proper legal theory, and not so much at legal AI knowledge. Consequently, it is hardly to be expected that contributions from the legal-AI community will be wholeheartedly welcomed since they will tend to suggest simplicity, uniformity and clarity on issues that perhaps had better not be clarified. An important part of legal expertise regards the reservation of absolute free competence of changing opinions. The use of freedom cannot be modelled without contradiction. We seem to reach the area in which legal expertise is meant to be subjective. As a matter of fact we must reconsider the aim of contributing with clear and objective models to this area of exclusively theoretical and little practical value.

As a result, there is a limit to the objective contributions legal AI can make to the issues distinguished on the approaches discussed. Research regarding these restrictions has already been performed (e.g. HYPO, PEM, CAMO). However, in the vein of Lawlor's (1979) "personal stare decisis" and as suggested by, e.g., Schöpping and Fernhout (1989), legal AI might venture to contribute further practical solutions in the form of *personal* LKBSs.

In summary, we believe that the categorization in types of legal knowledge and legal reasoning may be helpful in defining specific research projects. In Leiden, the current research projects on representation and on legal heuristics aim at contributing to the complete typology.

3.3. Categorization of domains

The third structuring approach uses well-known categories of legal content. At the moment, most legal AI research relies heavily on this technique. Parts of social security (JURICAS, TESSEC, LEIDRAAD, CAMO) are popular in this respect, and the current Leiden research projects are all committed to subdomains of social-security law. Important aspects of research on a great deal of LKBSs containing legal knowledge of different and restricted legal domains, treat the finding and validation of generic models (conceptual as well as formal models). Once again, a prerequisite for this type of research would be the use of standard modelling languages in earlier research.

4. Conclusions

We have shown that the aims of the legal-AI research in Leiden, as well as those of the legal-AI community in general have shifted from the idle hope of constructing "complete" solutions to more modest contributions. This shift has already supplied (and promises also to supply) the availability of techniques, meanwhile it distinguishes legal AI knowledge into several types which are easy to handle. As a promising technique, we have high expectations of the use of well-founded and well-documented conceptual- and formal-modelling languages. Although several of these languages exist, no one was specifically designed to model legal expertise. Consequently, the emergence of deficiencies is a risk that must be taken into account while selecting these languages. The Leiden research team prefers this risk to the continuation of cooperation based on Babilonical speech acts and has selected KADS and DESIRE as conceptual- and formal-modelling languages to present their future research results.

5. References

- Anjewierden, A. (1990). (ed.). *Shelley User's Guide - Version 1.0* ESPRIT project 1098 Technical Report, University of Amsterdam.
- Ashley, K.D. (1989). Toward a Computational Theory of Arguing with Precedents: Accomodating Multiple Interpretations of Cases. *Proceedings of the Second International Conference on Artificial Intelligence and Law*. ACM Press, New York.
- Berg, P. van den, A.F.M. Grütters, O.W.M. Kamstra and T. van Willigenburg (1988). *RI-Paradigmata - Toekomst van Nederlands onderzoek in de rechtsinformatica*. Koninklijke Vermande B.V., Lelystad.
- Bing, J. (1990). *Classification of Legal Knowledge-Based Systems and a Discussion of Similar Cases*. Norwegian Research Centre for Computers and Law, Oslo University.
- Breuker, J., B. Wielinga, M. van Someren, R. de Hoog, G. Schreiber, P. de Greef, B. Bredeweg, J. Wielemaker and J-P. Billault (1987). *Model-Driven Knowledge Acquisition: Interpretation Models*. University of Amsterdam, Memo 87, VF Project Knowledge Acquisition in Formal Domains.
- Capper, Ph. and R.E. Susskind (1988). *Latent Damage Law - The Expert System*. Butterworths, London.
- Cox, B.J. (1986). *Object-Oriented Programming. An Evolutionary Approach*. Addison Wesley Publishing Company, Reading, Massachusetts.
- Crombag, H.F.M., J.L. De Wijkersloot and M.J. Cohen (1977). *Een theorie over rechterlijke beslissingen*. Groningen.
- Eshelman, L. (1988). MOLE: A Knowledge-Acquisition tool for Cover-and-Differentiate Systems. In: Marcus, S (ed.) *Automating Knowledge Acquisition for Expert Systems*. Kluwer Academic Publishers, Dordrecht.
- Greef, P. de, J. Breuker and T. de Jong (1988) *Modality; An analysis of functions, user control and communication in knowledge based systems*. University of Amsterdam, A4-PR-004.
- Kowalczyk, W. and J. Treur (1990). On the use of a formalized generic task model in knowledge acquisition. In: B. Wielinga e.a. (eds), *Current trends in knowledge acquisition*, IOS, Amsterdam - Washington - Tokyo.
- Lawlor, R.C. (1979). Computer Analysis of Judicial Decisions. Niblett, B. (ed.) *Computer Science and Law*. Cambridge University Press.
- Leith, P. (1986). Fundamental Errors in Legal Logic Programming. *The Computer Journal*. Vol. 29, No. 6, 1986.
- MacCormick, Neil (197). *Legal reasoning and legal theory*. Clarendon Press, Oxford.
- Mulder, R.V. de, C. van Noortwijk and H.O. Kerkmeester (1989). Knowledge Systems and Law - The JURICAS Project. In: Martino, A.A. (ed.), *Pre-proceedings of the III International Conference on Logica, Informatica, Diritto*, Florence, Vol. I.

Nieuwenhuis, J.H. (1976). Legitimatie en heuristiek van het rechterlijk oordeel. *Rechtsgeleerd Magazijn Themis* 1976, 494-515.

Nieuwenhuis, M.A. (1989). *TESSEC: een expertssysteem voor de Algemene Bijstandswet*. Kluwer, Deventer.

Oskamp, A., R.F. Walker, J.A. Schrickx and P.H. van den Berg (1989). PROLEXS: Divide and Rule, a Legal Application. *Proceedings of the Second International Conference on Artificial Intelligence and Law*. ACM Press, New York.

Oskamp, A. (1990). *Het ontwikkelen van expertsystemen*. Deventer: Kluwer.

Quast, J.A. en J.H. de Wildt (1988). De ontwikkeling van een juridisch kennissysteem. *Proceedings NAIC-88*, Amsterdam.

Rissland, E.L. and K.D. Ashley (1987). A Case-Based System for Trade Secrets Law. *Proceedings of the First International Conference on Artificial Intelligence and Law*. ACM Press, New York.

Schmidt, A.H.J. (1987). *Pallas ex machina: Informele systemen in verband met het recht*. Koninklijke Vermande B.V., Lelystad.

Schmidt, A.H.J., J.C. Hage, J.A. Quast and H.J. van den Herik (1988). Onderzoek op het gebied van de rechtsinformatica te Leiden. In: Berg, P. van den, A.F.M. Grutters, O.W.M. Kamstra and T. van Willigenburg (1988). *RI-Paradigmata - Toekomst van Nederlands onderzoek in de rechtsinformatica*. Koninklijke Vermande B.V., Lelystad.

Schmidt, A.H.J. en H.J. van den Herik (1989), Kennissystemen als instrument voor rechtswetenschappelijk onderzoek. *Proceedings AIT*, 1989

Schöpping, J.M.A.L. and F.J. Fernhout (1989). Onderhoud van juridische programma's; Een empirisch onderzoek naar de haalbaarheid van decentrale software. *Toogdagbundel Juridische Informatie-systemen*, Netherlands Association for Computers and Law.

Schreiber, G., B. Wielinga, P. Hesketh and A. Lewis (1989). *A KADS Design Description Language*. Esprit P1098 Deliverable B7. University of Amsterdam & STC Technology Ltd.

Sergot, M.J., F. Sadri, R.A. Kowalski, F. Kriwaczek, P. Hammond and H.T. Cory (1986). The British Nationality Act as a logic program. *Communications of the ACM*, Vol. 29, No. 5.

Susskind, R.E. (1986). *Expert Systems in Law, A Jurisprudential Inquiry*. Ph.D Thesis University of Oxford Trinity Term.

Susskind, R.E. (1987). Expert Systems in Law, Out of the Research Laboratory and into the Marketplace. *Proceedings of the 1st International Conference on Artificial Intelligence and Law*. ACM Press, New York.

Visser, P.R.S., H.J. van den Herik, A.H.J. Schmidt en J.H. de Wildt (1990). Het modelleren van casus. *Proceedings NAIC'90*, Stichting Informatica Congressen.

Wildt, J.H. de and Ph. R. de Jong (1990). *Een jurimetrische benadering van het begrip passende arbeid*. JUS-Rapport.

Wildt, J.H. de and J.A. Quast. (1989). The concept of 'commensurate work' in a legal knowledge-based system. *Preproceedings of the Expert Systems in Law Conference*, Bologna, Italy.

PREVIOUS PUBLICATIONS BY JURIX:

RI-Paradigmata

Toekomst van Nederlands onderzoek
in de rechtsinformatica

Paradigms in legal informatics

*Exploring the future of Dutch research programmes
in legal informatics.*

Editors:

P. van den Berg, A.F.M. Grütters,
O.W.M. Kamstra, T. van Willigenburg

Legal knowledge based systems

An overview of criteria for validation
and practical use

*Exploring the quality of applications resulting
from research programs in the Netherlands*

Editors:

D. Kracht, C.N.J. de Vey Mestdagh,
J.S. Svensson