Artificial Intelligence and Law **1:** 245–274, 1993

Model for knowledge and legal expert systems

ANJA OSKAMP
Computer/Law Institute, Vrije Universiteit Amsterdam, De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands

(Received 6 November 1991; accepted 16 July 1993)

Key words: legal knowledge sources, metaknowledge, levels of knowledge, knowledge representation

Abstract. This paper presents a four layer model for working with legal knowledge in expert systems. It distinguishes five sources of knowledge. Four contain basic legal knowledge found in published and unpublished sources. The fifth consists of legal metaknowledge. In the model the four basic legal knowledge sources are placed at the lowest level. The metaknowledge is placed at levels above the other four knowledge sources. The assumption is that the knowledge is represented only once. The use of metaknowledge at various levels should make it possible to use the appropriate knowledge for the problem presented to the system. The knowledge has to be represented as closely to the original format as possible for this purpose. Suitable representation formalisms for the various types of knowledge in the five knowledge sources are discussed. It is not possible to indicate a 'best' representation formalism for each knowledge source.

1. Introduction

1.1. POINTS OF DEPARTURE

Many projects on the use of expert systems in law tend to focus on one very specific legal task and on developing an expert system for that particular task. In this way it is possible to confine the research to a limited amount of knowledge as well as to a restricted use of this knowledge. This paper looks at the subject from a different point of view. It starts from the knowledge itself. This leads to the following questions that are looked into in this paper. What kinds of knowledge could, or should, a legal expert system use? What does this knowledge look like? How can such knowledge be implemented in and be adequately used by an expert system, taking into account the nature of the knowledge? These questions are dealt with in a general way. They are not related to either a specific domain or a specific task. The examples in this paper for the most part do not refer to legal policy making. Rather, the examples emphasize the task of giving legal advice.

This paper is predicated on the belief that as much as possible expert systems should embody the same knowledge as a human expert would use to perform the same task. But there exist certain ineluctable limits to the amount of knowledge that an expert system can contain because human experts possess vast amounts of common sense knowledge; and their expertise traverses many levels of knowledge and covers a wide range of topics.

1.2. THE NEED TO MODEL KNOWLEDGE FOR LEGAL EXPERT SYSTEMS

It is clear to every lawyer (as well as to many non-lawyers) that applying legal rules to the facts of a specific cases is not a simple process. Statutory rules, for instance, are meant to refer to many different situations and, therefore, are put in general terms. Applying these rules to a specific case requires the interpretation of both the legal rule itself and the terms used in it. Using only statutory rules is not sufficient for law application. Other types of legal knowledge are needed, as well as common sense knowledge. Therefore, as Sergot rightly suggests [Sergot 1991, p. 11], expert systems performing legal tasks need different kinds of knowledge. Without further means of interpretation, a legal expert system may only be able to give a schematic overview of the legal rulès in the domain chosen. In certain cases it could mechanically follow the applicable rules and by mere deduction come to a conclusion. This might solve easy cases when the circumstances of the case strictly match the conditions of the rules.[1] Even then, most choices and interpretations will have be to made by the user of the system. She will have to decide whether the facts match the conditions of the rules. These systems by setting forth a narrow framework to be filled in by the user often provide a rather clear overview of the rules within a certain domain. But they only offer the user very restricted paths within which those rules can be applied.

A few legal expert systems have been constructed in this way and are already commercially available. Susskind explains that the paucity of such legal expert systems

> ... is explicable partly in terms of the nature of the legal knowledge that has so far been input to the systems; the coverage of legal domains has invariably been shallow with emphasis solely on statute law. Yet it is clear to all who have engaged in legal problem-solving that statutes often provide no more than a starting point for legal research, and recourse to case law and the commentaries of legal scholars must generally be made [Susskind 1987, p. 74].

By adding other types of knowledge expert systems will improve their value by being able to perform more complicated tasks. As Bench-Capon [1989, p. 41] observes, the systems should not operate merely on a syntactic level, as the deductive systems do, but also take into account the semantic level.

These observations are far from new. They apply to most research projects on legal expert systems. However, often the decision as to which knowledge to implement in a specific expert system is taken on an ad hoc basis. The contents of a knowledge base are determined according to the domain and the task the system is to perform. Although these should be the main criteria for choosing the knowledge, they may also severely limit the applicability of the system. Conversely, a model giving more general guidelines will make it possible to make the choices consciously. This can improve the range of applicability of the system. It also offers the possibility to reuse parts of the knowledge bases. By this I mean that a careful ordering and representation of the knowledge could make it possible to use part of the knowledge base for expert systems that perform different tasks within the same domain. This paper presents such a model.

[1] But note [Gardner 1989, p. 19].

1.3. THE BASIS OF THE MODEL

Several ideas lay behind this model. The first is that a legal expert system should have almost the same knowledge at its disposal as a human lawyer. The second is that the representation of the knowledge should be as isomorphic as possible. By identifying the knowledge with the source one preserves valuable information about the knowledge [Haft *et al.* 1987, p. 79; Biagioli *et al.* 1987, p. 243]. The third is that the knowledge can be used for different kinds of situations, in different sequences and in different combinations. This is necessary to apply the law according to the circumstances, and 'take into account the purposes behind the rules' [Berman & Hafner 1987, p. 6].

There exist more general requirements for robust legal knowledge bases. The user must be able to retreive the knowledge quickly, effectively and accurately. The system has to operate in a transparent, clearly structured and flexible way. Finally, it should be easy to maintain. The model proposed in this paper offers the possibility to fulfil these requirements.

There are two cornerstones of this model. First, it is based on the use of different forms of legal knowledge. This knowledge is implemented in the knowledge base according to its original source. There are five different sources to be distinguished. This is described in Section 2. The second cornerstone is that the knowledge base is built in four layers, each with a specific kind of knowledge. This is described in Section 4. Building on these cornerstones, Section 5 describes the format in which the knowledge is to be inserted in the system.

The model proposed in this article has not yet been implemented. Rather, I have composed a theoretical model to enable the building of flexible and maintainable legal knowledge bases, containing as much knowledge as possible. Should one have to take into account the possibility of practical application, this would imply making concessions. For this reason I have restricted myself to indicating the ideas on which the model is based. I am aware of the fact that the practical realization of these ideas will be far from easy. For instance, problems of a technical nature might occur as the model proposed will ask for powerful computer systems because the amount of knowledge contained in such systems may result in unacceptable response times when the system manipulates the knowledge.

This model only gives guidelines. Many problems still have to be solved in order to make a system based on this model function in an optimal way. For example, problems referring to the possibility of enhancing isomorphism [Prakken & Schrickx 1991] need to be addressed. Therefore, this model cannot be seen as more than a point of departure and an indication for further research. Such research is currently being carried out within the Prolexs-project at the Computer/Law Institute of the Free University in Amsterdam. This paper is based on a research project that was part of the Prolexs project,[2] but it is not a description of the Prolexs system. For a description of the Prolexs system see [Walker &

[2] This research project resulted in my thesis, which was finished in the summer of 1989. It was published, in Dutch, by Kluwer Deventer [Oskamp 1990]. This paper is a revision of a part of this thesis. It is updated, especially with respect to relevant literature and to some developments in the field. However, it remains a reflection of ideas that were mainly formed in the period 1986–1989. Therefore, some parts may be slightly outdated. I tried to avoid this as much as that was possible without affecting the line of thought which lies underneath the model presented in this paper. For the points were it does occur I ask the reader to consider it as an example of the rapid progress in the field.

van den Berg 1988; Walker *et al.* 1989; Walker *et al.* 1991]. This paper describes a theory that goes further than the currently implemented Prolexs system, which was based on an earlier version of this theory [Oskamp, 1989]. Nevertheless, I was able to extract many examples from the Prolexs project and it will be referred to several times in this paper.

2. Legal knowledge for legal expert systems

2.1. THE LEGAL KNOWLEDGE SOURCES

This section distinguishes among the various forms of knowledge that are used for applying the law to specific facts and sets forth the proper ordering of the knowledge. Finally, it constitutes the basis of the different layers of the model described in Section 4.

The knowledge necessary to perform legal tasks comes initially from traditional legal sources. These written sources include: legislation, case law and the various forms of legal literature. Whenever these sources exist in a specific domain, using them is a *conditio sine qua non* for the performance of many legal tasks. Hence, they are indispensable for an expert system that will have to perform such a tasks. The application of these legal sources calls upon legal expertise as well as common sense knowledge. This knowledge can not be found in traditional sources. Rather, it resides with experts in the domain. Expert knowledge can be divided into two groups: factual expert knowledge and expert metaknowledge.

This leads us to the observation that five separate knowledge sources can be distinguished. Three of them, i.e. legislation, case law and knowledge based on legal literature, guidelines, etc., find their origin in widely accepted written materials. The fourth, expert knowledge, consists of experiential knowledge that is mostly factual knowledge that is relevant to a specific domain. The fifth source, legal metaknowledge, consists of knowledge that makes it possible to optimize the use of the knowledge coming from the other sources. Hereafter I will indicate these sources as the five knowledge sources of a legal expert system.

2.2. THE KNOWLEDGE SOURCES: LEGISLATION, CASE LAW AND LEGAL LITERATURE

The knowledge source 'legislation' contains all legislation relevant in the domain. In Dutch law this includes the constitution and other statutes, but also lower laws, such as treaties, decrees and by-laws from county councils and municipal councils.

The knowledge source 'case law' contains the judgements of a variety of courts. In this knowledge source it has to be possible to refer to a part of a specific case and to use it as an argument. All knowledge not based on one specific identifiable case will either be part of the knowledge source called 'legal literature' (if it is published) or part of the knowledge source called 'expert knowledge'. One form of expert knowledge is produced by deducing an 'average' of a series of cases rather a compiling rules found in individual cases [Susskind 1987, p. 98]. Such secondary sources are often published materials that contribute to a better understanding and more appropriate application of legislation.

Examples of legal literature include the written history of statutes, such as the Explanatory Memorandum; a Memorandum in Reply and other forms of direct explanation to legislation, such as guidelines to and from regional and local authorities. Further examples of the legal literature are text books and papers in legal journals.

2.3. EXPERT KNOWLEDGE AND LEGAL METAKNOWLEDGE

The fourth knowledge source is ‘expert knowledge’ and the fifth ‘legal metaknowledge’. They are both expert knowledge, although they each represent a different form of expert knowledge. Susskind distinguishes two forms of ‘experiential knowledge’. First, there is

> ‘juristic knowledge’, ‘the informal, judgemental, experimental and often procedural knowledge’ and in the second place ‘non-juristic procedural knowledge’, by which he means ‘knowledge about how to go about the administration of the law’ [Susskind 1987, p. 57]

However, Susskind’s categories do not correspond to my classification. The larger part of Susskind’s groups are better classified as ‘metaknowledge’. In this knowledge source I place all forms of knowledge that give some information on the handling other legal knowledge. It consists of rules of thumb and strategies; the overview that an expert has of the domain; and the weight one should attach to the knowledge from a specific source. It is absolutely necessary that legal expert systems embody metaknowledge [cf. Susskind 1987, p. 106]. Without this knowledge one can not adequately apply the ‘rough’ knowledge from the legal sources.

Metaknowledge which controls the application of knowledge from the various legal sources turns systems into expert systems.

> The most important goal in expert system work is to attain the high level of performance that a human expert achieves in some task. Acting like an expert means producing high quality results in minimal times, usually by taking advantage of tricks of the trade and high-level inference patterns (hunches) that come from years of experience at a given task [Hayes-Roth *et al.* 1983, p. 43].

I incorporate into the legal expert systems forms of expert knowledge which Susskind does not consider. This knowledge is similar to the knowledge from the first three knowledge sources. In a way it is a ‘group of remainders’ – everything that does not fit into the other three categories and does not qualify as metaknowledge falls under ‘expert knowledge’. Therefore, this eclectic knowledge source consists of various and often very distinctive forms of knowledge.

Its contents can vary from domain to domain, and even from task to task within the same domain. Although this expert knowledge is often of priceless value, it is often not reduced to writing and ‘lives’ in the head of the expert. This ‘expert knowledge’ does not even have to be pure legal knowledge. More general knowledge or common sense knowledge, which contributes essentially to legal knowledge and by this to the quality of the output of the expert system, is classified as ‘expert knowledge’. An example is knowledge about the meaning of ‘ordinary’ language, when it is used in the same way in the legal domain. [Cf. Bing 1987, pp. 6–10; Gardner 1987, pp. 43–44] Another example is knowledge based on an understanding of the common law which cannot be readily extracted from the published opinions. This kind of expert knowledge can also be based on the expert’s memory of judgements in similar cases that are often not published nor generally known.

Such knowledge can be rather specific. It can even be restricted to a specific region or to a precise court. One could even think of knowledge about habits, peculiarities and predilections of specific judges that may influence the result in a litigated case. Of course, this knowledge is the most specific knowledge that an expert system can contain. It will vary when the task changes or even when the group of users changes. A large part of this knowledge can hardly be used in general expert systems. It will be restricted to systems 'tailor-made' for specific clients.

Finally, a very specific form of expert knowledge is the knowledge that makes it possible to adjust the outputs of the system for diverse uses. This form of expert knowledge also depends very much on the task of the system, on the domain and on the users. An example: an expert system gives legal advice in the domain of landlord/tenant law. Considering the facts of the case it could come to the conclusion that the client is allowed to refuse to pay the increase of rent demanded by the landlord. The system could merely pass this outcome to the user. Another possibility would involve composing a standard letter using all relevant details of the case at hand. The client could then put the letter into an envelope, put a stamp on it and bring it to the nearest mailbox[3] [Oskamp 1986, p. 702]. Such an option asks for a careful inventory of the possible forms that the output can take and a careful choice of the most suitable form of output in specific case.

2.4. DISCUSSION

In building a knowledge base, an obvious way of ordering the knowledge is to reference the original knowledge source. This ordering provides information on value and position of the original source in the hierarchy of the knowledge sources. [Cf. Haft *et al.* 1987, p. 202; Biagioli *et al.* 1987, p. 243] The value of arguments based on a specific source can also be determined. Information on the variety of sources on which a specific argument is based will not be lost. In addition, incorporating various sources of knowledge also adds to the flexibility of the system. For instance: in many cases the interpretation of legislation with the help of case law is important. The facts of the present case will dictate which prior judgements are most relevant. In other cases the use of legal literature or expert knowledge to interpret legislation will be more to the point.

Another example: published case law says A, but the expert knows that one specific judge in 9 out of 10 cases says B in such circumstances.[4] When the expert knows that this specific judge will deal with the case, he will take into account this judge's preferences. The knowledge source 'expert knowledge' of a specific expert system, for instance developed for a legal clinic in a specific jurisdiction, could contain such knowledge. However, such knowledge should be readily traceable in order to maintain it since these headstrong judges are often overruled.

[3] In such a case it may be inevitable that the person consulting the expert system is not the client, but an intermediary. This legally trained intermediary would have to take care that possible policy questions and the weighing of interests are done in a correct way. He would also be able to give further explanation.

[4] This is allowed in the Dutch legal system, but the judge can be overruled.

Last, but certainly not least, it is absolutely necessary that the knowledge base of an expert system be ordered efficiently because these knowledge bases can become extremely large. For that reason one should try to encode all knowledge only *once*, even where specific knowledge will often be applied in very different situations with different interpretations. Encoding the knowledge only once also facilitates maintenance because the knowledge only needs to be modified once.

3. Examples of models for legal knowledge bases

An expert system knowledge base should be constructed to reason by combining and inferencing with several knowledge units. To ensure tractibility one should avoid designs that require the searching of the entire knowledge base. Therefore, I propose models for structuring the knowledge base to obtain the economies achieved through a single entry into the knowledge base while still being able to efficiently retrieve the knowledge. After discussing some of advantages and disadvantages of these models I will, in Section 4, present my own model, which integrates various elements of these models.

3.1. MODEL 1: CLUSTERING THE KNOWLEDGE

The knowledge in a legal expert system can be gathered in what one could call problem-oriented clusters. These are clusters of knowledge directed towards solving one specific kind of problem. An expert system itself is already a cluster of knowledge on a specific domain. However, it can be further divided into clusters containing knowledge referring to specific subdomains. This method of clustering has as its principal advantage that when the system has decided which cluster of knowledge is the most effective to solve the problem put to the system, it will only have to use the knowledge from this cluster. In this way the quantity of knowledge with which the system has to reason is limited.

This kind of system is goal oriented. The goal will define the necessary knowledge, and the system will usually backward chain. This method is very suitable for systems that have to solve a limited number of rather specific and well described problems. Before building a cluster system one should make sure that the problems presented can be solved with a rather limited quantity of knowledge.

A disadvantage of this method is that it limits the possibilities. Knowledge that is not directly related to the problems to be dealt with in the cluster will not usually be part of that particular cluster even though this knowledge may be contained within other clusters that may be part of the expert system. Thus the system will not be able to find a solution when the facts do not exactly match the conditions of the rules [Susskind 1987, p. 151], even though the necessary knowledge can be found elsewhere in the system. Changing clusters will then be necessary. This requires a flexible system which can become rather difficult to develop and maintain.

Another disadvantage is that knowledge located in one cluster but which is needed frequently in other clusters, will usually have to be encoded more than once. We are talking, in a way, about a collection of many little expert systems, which are each capable of solving one specific problem. As mentioned previously, such an approach causes maintenance problems because the knowledge must be entered more than once. In addition to maintenance problems, the cluster system may lack flexibility because of the problems of transporting certain knowledge from one small expert system to another.

3.2. MODEL 2: A DECISION TREE

An expert system using a tree structure follows predetermined paths through the knowledge base. These paths are related to specific problems. The system puts questions to the user, to which this user can give a limited number of answers, such as 'yes', 'no', 'do not know', 'maybe'. The system leads the user along specific and fixed paths corresponding to the answers to the questions. There are many variations to this system because it is possible to combine it with networks or to work with menus. A number of operational legal expert systems are based on a variation of such decision trees [McCarty 1987, p. 197; Van Noortwijk 1990, pp. 36–40]. These trees, like in the systems of Schlobohm [Schlobohm & Waterman 1987, pp. 18–28] and Susskind [Capper & Susskind 1988] often become quite elaborate.

An advantage of this method is that the system can handle the knowledge rather easily because the path has been traced out beforehand. In this way large knowledge bases can be assembled expeditiously and traversed rapidly. Non-essential knowledge will not be used. Another advantage is that the system is rather transparent because it is possible to indicate the paths beforehand and even publish them. This enables people other than the developers to evaluate and maintain these systems.

Finally one could claim that such an approach enhances flexibility [Susskind 1989, p. 3], particularly for maintenance purposes because whenever a change occurs, one only has to change the rule and, perhaps, the path in question. I disagree with this claim. Flexibility is directly related to the way in which the alterations influence the structure of the path or paths. This tree structure, however, is far from flexible because the paths are fixed beforehand which means that most alterations will have many consequences for the whole path. Moreover, the process of indicating the paths is often rather labour-intensive.

Such an inflexible structure can negatively impact on the decision process because the variety of problems that can be solved will be fixed at the time of developing or maintaining the system. Every question that falls outside this structure cannot be dealt with by the system. Nor can the system respond appropriately to facts that do not exactly match the conditions of the rules. A further disadvantage is that the system can only use knowledge that falls within the path. The latter is a severe restriction because often one needs to use additional knowledge that lies outside the restricted paths in order to make a balanced appraisal. Therefore, these systems usually are quite shallow [Cf. Bench-Capon, 1989, pp. 38–39]. On the other hand, these systems can be very useful when dealing with repetitive cases. Schlobohm indicates that his estate planning system can be used for 70% of the estate planning cases. Thus

> the expert could spend his or her time more creatively by designing estate plans for the minority of clients who could not be adequately accommodated by the expert system [Schlobohm & Waterman 1987, p. 19].

A further potential disadvantage of this type of system is that the user must often answer every question. This often results in a tedious undertaking because the user must answer many questions that are not relevant to the case at hand.

3.3. MODEL 3: STRUCTURING RELATED TO THE DOMAIN

Gardner's model is of a different nature. Her model orders the knowledge strictly within a domain and a task. Its points of departure are the various distinguisable phases in the process of contracting [Gardner 1987]. The big advantage of her model is, in my opinion, that it distinguishes among various levels of legal reasoning. It starts with the establishment of the legal nature of the problem and the identification of the legally relevant relations. Then her model goes via the solution of a problem by the sole application of legal rules to completing the concepts that cannot be established by the rules. The model reflects the process of legal reasoning, but it does so in a very narrow part of a rather specific domain. Gardner's approach also raises some unanswered questions. First, will her method work in other domains? Second, can one transfer her approach which is based on the common law legal system to civil law systems? Finally, one wonders whether her system, which provides an analysis of specific problems and includes an indication of the questions that it cannot solve, can be used for building expert systems outside the laboratory environment in which it was developed.

The factual problem addressed by Gardner's model is rather artificial because the cases have been constructed with a view to a rather elementary form of legal analysis. I question whether the system will be able to make such an analysis in real world settings involving rather atypical situations [cf. Goldman *et al.* 1987, pp. 220–221]. These questions become important when we are talking about systems that are to be used in practice. The range of the knowledge bases of such systems will be much larger than the one Gardner used. Moreover, the nature of the real world problems will demand that the system performs many forms of analysis not contemplated by Gardner's system.

4. A New Model for the Use of Legal Knowledge

4.1. FOUR LAYERS

The model I propose in this section consists of four layers, each containing a different kind of knowledge. The basic idea is that all knowledge will be represented in a knowledge base as objectively as possible (but see Section 5.6.), and preferably only once. The lowest layer will contain all relevant knowledge except metaknowledge; the second, third and fourth layers will contain various types of metaknowledge. These last three levels will combine, interpret and manipulate the knowledge in the first layer.

4.2. THE FIRST LAYER

The first layer contains all relevant knowledge from legislation, case law, legal literature and expert knowledge. In this layer knowledge will be identifiable with the original knowledge source to which it belongs. That can be rather easily accomplished by arranging the knowledge according to the knowledge source. This identification of the source is important because the origin establishes the place in the hierarchy among the various knowledge sources. For that reason the content of the knowledge sources is strictly defined. The knowledge source 'legislation', for instance, only holds the literal text of the statutes. It does not hold the explanation of or the concepts that inhere within these statutory rules unless these are included in the statute itself. In this way the statutory rules are strictly distinguished from their interpretation.

Separating the text of a rule from its various interpretations will improve the flexibility of the system. The rule and its concepts can thus be interpreted more easily in view of the circumstances of the case. More important, however, is that separating rule and interpretation underlines the special position of the rules, as well as their reference function: legislation is the point of departure for every lawyer. This point of view also implies that the knowledge at this level has to be represented as objectively as possible. This could mean the use of a number of different representation techniques. I will deal with this in Section 5.

Grouping the knowledge according to the knowledge source is only a first ordering. Within each knowledge source we can identify different groups of knowledge with different values. Refinement in the knowledge source itself is possible and often necessary. This is especially important for the legal knowledge sources 'legal literature' and 'expert knowledge'. They contain many different forms of knowledge, with many different values and a different status. But also for legislation and case law further arrangement may be necessary by reflecting the hierarchy of courts or the hierarchy within legislation. This arrangement, refined or not, is the only form of metaknowledge at this level.

The hierarchy of the knowledge sources is partly defined by the distinction between the primary sources of law (legislation and case law) and the secondary sources of law (legal literature and expert knowledge). The primary sources contain the law itself, while the secondary sources contain law derivations: personal interpretations of the law. The primary sources of law are clearly higher up in the hierarchy among the knowledge sources. Furthermore, legislation generally prevails over case law. An exception to this is a judgement *contra legem.* Such a judgement could be based on a change in social circumstances, which, in the view of the judge, makes it no longer appropriate to apply the statute to the facts of the case. The judge then contradicts that part of legislation. Case law *contra legem* will be a rare occurrence, but it has important consequences for legal expert systems because it affects the hierarchy based on the principle 'legislation as a point of departure'. When a judgement *contra legem* is inserted in the knowledge base, it needs a special status since it may take precedence over knowledge from the knowledge source 'legislation'.

An exception to the prevailing character of legislation can be found in 'regelend recht' (*jus dispositivum*). In Dutch law this phenomenon gives parties the power to make legally valuable arrangements by themselves, while also giving binding rules to cover the situation that occurs if parties refrain from making arrangements. While making a contract, however, the parties are allowed to make deviations of the rules given by a statute, and thus overrule the statutory provisions. In the Dutch legal system this is allowed if done 'bona fides'. A problem with this is that the status of the applicable part of the knowledge source 'legislation' becomes unclear. It is either extracted from the statutory provisions and as such subject to alterations in a contract (of course this then has to be checked by the expert system), or it emanates from contractual provisions. In the last case it should be checked whether these provisions are 'bona fides' and whether the contract is legally valid in other respects. This requires extra knowledge about a subject, contract law, which is not the easiest with which to deal [Gardner 1987].

Even if it were feasible to evaluate the contractual provisions, two other problems would arise. The first relates to the dynamic character of this knowledge: given the variations in contractual terms the knowledge will usually only be used once or twice and it may change all the time. The second problem is due to the fact that the input of the contractual provisions will have to be carried out by the user. This problem can partly be dealt with by having the system help the user to insert that knowledge in a special format and within specific frameworks. However, the general maintenance problems will have to be dealt with.[4] Furthermore it may affect the reliability of the system to have users sorting and inputting this amount of knowledge.

Summarizing, we may say that contractual provisions are of a dynamic nature and that there are differences in reliability between legislative rules (input by the knowledge engineer) and rules based on contractual provisions (input by the user). This implies that it is not wise to put the rules of contracts in the same source as legislation. If one wants to provide the use of contractual provisions a different solution has to be found. An extra knowledge source, holding temporary knowledge, could be inserted at this level.

4.3. THE SECOND LAYER

The second layer, although useful, is not really necessary for the model itself. It is intended to enhance an effective search for possible relevant knowledge. This layer contains what I call a 'reference file'. This reference file gives cross-references to all knowledge connected in any way to any specific unit of knowledge. It facilitates the recovery of all related knowledge, without regard to the knowledge source in which it is to be found. For instance, in the knowledge source 'case law' we would find references to the legislation to which the judgement refers. One might also find references to other relevant case law as well as to knowledge within the knowledge source called 'legal literature' if the literature discussed the judgement or refers to similar interpretations. Thus, the reference file gives a complete overview of all the interrelations within the knowledge base.[5]

Such a reference file makes explicit the knowledge that can be extracted by the use of an inferencing engine. Explicit representation has the advantage of permitting a more rapid retrieval of relevant knowledge. When correctly constructed this reference file which

[4] It has to become clear what other knowledge is or can be affected by the new knowledge. How can consistency be enforced? What is the impact of the new knowledge on the metaknowledge?

[5] The reference file does not attempt to resolve contradictions that might arise because a specific concept is explained in a conflicting way.

makes accessible all possible connections serves as a control mechanism that facilitates maintenance. Finally, the reference file guards against inconsistency in the knowledge base. By juxtaposing the contradictions that may exist within the entire knowledge bases the user is permitted to make explicit choices among competing sources of knowledge.

It is not important what these references look like. One could embody them in an 'inverted file' similar to the methods used to construct certain databases. Another possibility is the use of 'vector retrieval', a technique that is also proposed for legal databases. One would need to modify these techniques for the use I propose by using the methods suggested by Kracht *et al.* [1988, pp. 34–41].

There exists a potentially more efficient way for creating such a reference file. One could add such references to the represented knowledge itself, provided that it can easily be recognized as additional knowledge and thus be separated from the 'real' knowledge. In the Prolexs system, for instance, this is realized by encoding this knowledge in the form of 'forward' and 'backward' pointers in a so called 'static' knowledge base of rules. Forward pointers refer to the productions that have the fact in their conditions, backward pointers refer to rules which have the fact in their action. Whenever the fact appears on the blackboard[6] the system will thus 'know' which rules it can 'call'. See also [Walker *et al.* 1989, pp. 233–259].

4.4. THE THIRD LAYER

In the third layer we find the general legal metaknowledge, legal metaknowledge that is generally applicable to every legal domain. The domain-specific metaknowledge is found in the fourth layer. Examples of general metaknowledge are the knowledge that reflects the hierarchy 'legislation, case law, legal literature', the hierarchy of the courts, or the hierarchy of different types of legislation. Another example is knowledge about the principles 'lex superior derogat legi inferiori' (Higher law takes away the effect of the lower laws), 'lex posterior derogat legi priori' (Later statute takes away the effect of the prior one) and 'lex specialis derogat legi generalis' (Special statutes take away the effects of more general statutes). The larger part of the knowledge in this layer will be available in a written form, for this is the kind of knowledge which has to be taught to law students. However, this does not mean that it will also be easy to reflect this knowledge in expert systems.

In the first place it will have to be indicated as such, which means that the knowledge engineer and the expert will have to recognize it as general legal metaknowledge. In the second place, it may not not be clear whether such a general metarule is applicable: it must be evident that a special (part of a) statute referring to a specific problem exists. Moreover, one can often find borderline cases where it is not clear whether 'lex generalis' or a 'lex specialis' should govern. Yet a failure to recognize these borderline cases will result in the expert system giving a wrong answer. This means that (for certain domains) metarules have to be found that indicate the applicability of a *lex specialis* [see also Prakken 1991, pp. 165–174]. This is also a problem of knowledge representation.

[6] The Prolexs system is a blackboard system.

Working with analogies, either to real or hypothetical cases, may also be a solution. Prakken & Schrick [1991] give two examples of formalising a *lex specialis* without violation of the isomorphism.

4.5. THE FOURTH LAYER

The fourth layer contains the metaknowledge that is directly related to the domain and the task of the system. This metaknowledge can make it possible for an expert system to reason on various levels, from shallow to deep, depending on the situation and using various knowledge sources [cf. Bench Capon 1989, p. 42]. It is this form of knowledge that will make it possible to find the appropriate knowledge and apply it in the correct way. It will also help to ensure that the system only consults relevant knowledge. To do this, the system can use the reference file in the second layer to find possibly relevant knowledge. Then it has to circumscribe this knowledge with the help of metaknowledge from the third and fourth layer. I give three (rather obvious) examples, which, as will become clear, are parts of a strategy. There are other possibilities which include combining aspects of the examples given. The choices will, among others things, depend on task and domain.

4.5.1. Imitating the Expert

A first possibility is to imitate the working method of an expert. By this I mean that the metaknowledge reflects how an expert handles a specific problem, which knowledge sources he consults, and in which sequence.[7] Referring to specific problems, the paths through the knowledge can be dictated beforehand, just like in a tree structure (see also model 2 in subsection 3.2.). However, the difference is that here the path is dictated in a structure above the knowledge and not by using the knowledge itself. Thus, within the path there need only be indications that it is necessary to consult the knowledge source 'case law' first rather than express pointers to a particular judgment. The choice of a particular case will be deduced from the facts that are given to the system and may vary according to the circumstances. Following this particular path should only be one of many possibilities which is selected because of the specific circumstances of the case.

An example: a tenant believes that he pays too much rent. He consults an expert system to verify this. The expert system 'sees' that the facts of the case can be matched to the goal 'rent reduction' more than to any other aim. Therefore it decides that it will first try to achieve the goal 'rent reduction'. To achieve this goal the expert system can choose between several paths. Two of those seem the most appropriate: 'low quality' and 'rent reduction'. The system now has to decide which it should try first, considering the facts given by the user. In doing so it will, among other things, compare the facts of the case with facts of applicable case law. If that is not successful it can try other paths. (An

[7] Some problems can be dealt with very well in this way. For instance, when the user knows what he wants to achieve and asks the system whether this can be realized with the facts he has given the system. Then the system can easily verify this by use of backward chaining. An example is given below.

extended version of this example can be found in [Walker *et al.* 1991, pp. 42–49]). When used in this way, a decision-tree structure will not necessarily render the system inflexible provided that there exist different ways and sequences of consulting the knowledge that can be invoked on the fly.

4.5.2. Clustering the References

A second possibility of using a metaknowledge strategy is combining and activating all the knowledge which may be necessary for solving problems in the domain. This can be problem oriented or subdomain oriented. The link to the domain makes it fourth layer metaknowledge. The combination meant here need not be a fixed combination, i.e. that the knowledge itself is clustered, like in model 1 containing clusters. It can, for instance, also be realized by grouping the references to that knowledge on a level above the knowledge itself. This will make it possible for the system to gather the knowledge related to a specific subdomain or a specific problem rather quickly.

An example for these types of clusters is found in the so called logical groups in the Prolexs system [Walker & Van den Berg 1988, pp. 11–12; Walker *et al.* 1989, pp. 244–243]. These logical groups are all related to a specific subject like ‘rent reduction’ or ‘maintenance’. A logical group uses knowledge from all knowledge sources related to that specific subject. In the example of subsection 4.5.1 the logical group ‘rent reduction’ would be selected and the system would start reasoning with the knowledge collected in this logical group. A classification network shows the connections between the various logical groups. This also indicates which logical groups can be relevant to achieve a specific aim, since the logical groups are connected in a specific sequence. In this example it would also be necessary to activate the logical group ‘maintenance’ to check whether maintenance deficiencies would be a cause for rent reduction. The classification network as such is a form of metaknowledge, lying over the logical groups. See Figure 1 for an example.

4.5.3. Using Heuristic Knowledge

A third possibility is to have the system find its way among the knowledge using heuristic knowledge and metaknowledge from the third and fourth layers, the latter especially reflected in rules of thumb. The system could, for instance, start with forward chaining, by mainly using rules and inference chains. Using experiential knowledge in the form of rules of thumb or strategy rules, the system ‘knows’ which knowledge it has to consult. This experiential knowledge can also be achieved by the system itself. That would be a self learning system. The rule which regularly leads to a solution and which seldom fails would then be used more often.

4.5.4. Combination

As indicated by the example in subsection 4.5.2 a combination of several possibilities would make the system more powerful, flexible and effective. Consider the following

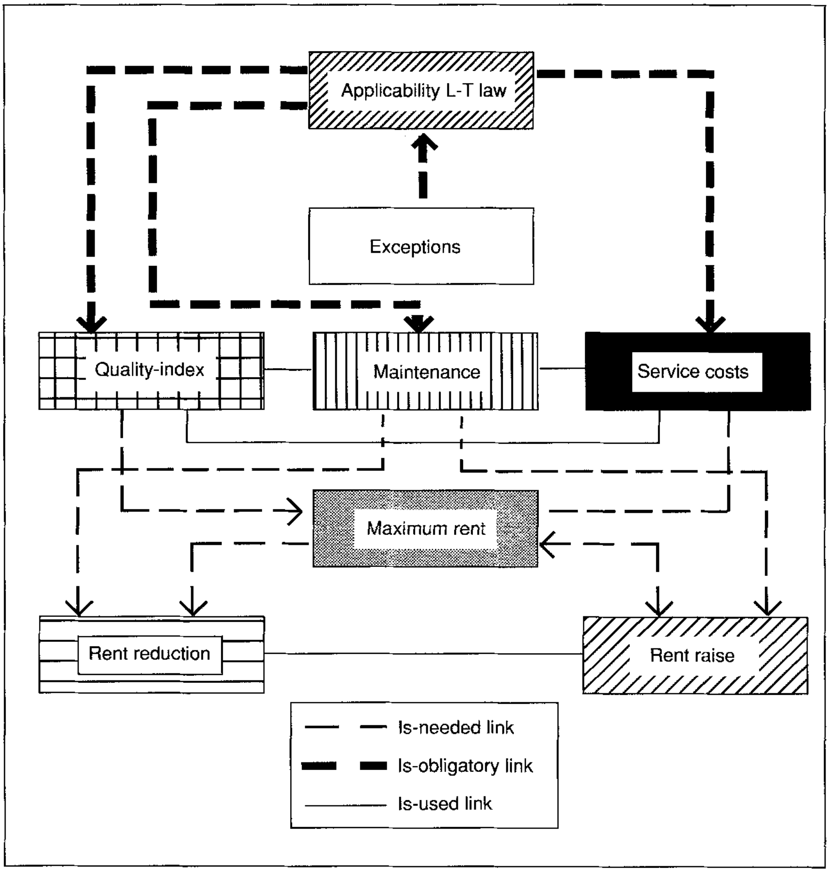

Fig. 1.

example: When only specific facts are known and the system knows nothing about the aim it has to achieve, the system will start with forward chaining. It will start with backward chaining when the user indicates the aim that the system will have to achieve[8]. To solve simple or standard problems choosing only rules of thumb may be sufficient. Flexibility is assured when the system can change its strategy during the reasoning process. For instance, when the system tries to achieve a solution by forward chaining it may be necessary to establish the validity of a condition by backward chaining. The system may also want to test an alternative solution by using a different strategy during the reasoning process to see whether that gives better results or a different outcome. By clustering the

[8] Here the user is introduced in a more active role [Cf. Purdy 1989, pp. 201–202]. The user could also, for instance, indicate a specific way in which he wants the knowledge to be handled, or a specific aim he wants to achieve. In the example above (in Section 4.5.2.) the user would indicate that he wants the system to achieve the aim 'rent reduction'.

knowledge (possibility 2), the reasoning process can take place effectively and efficiently: the system will only need to reason with the knowledge which is relevant (or may be relevant) for the problem at hand.

The wish to have this combination also puts demands on the shells to be used. The Prolexs shell has the potential to implement the model proposed in this part. It already reflects parts of this model.[9] The Prolexs system distinguishes between the five knowledge sources as described in Section 2. Four of them are situated at the lowest level, while metaknowledge (especially domain-dependent metaknowledge) is situated at a level above. The knowledge sources can each be represented in their own way. However, a further distinction between the knowledge of each source has not yet been realized. There is also no clear distinction between general legal metaknowledge and domain specific metaknowledge. Currently the metaknowledge is mainly manifested in so called 'logical groups', consisting of knowledge referring to a specific topic in the domain. The knowledge may come from the various knowledge groups. Logical groups are not dependent on the way in which the knowledge is represented. Another form of metaknowledge, situated at the highest level in the Prolexs system, is formed by the classification network, which combines the logical groups. It is both used to classify a problem and as a means to decide which topic to concentrate on. It thus expresses classifying and strategic expert knowledge. Figure 1 expresses a part of the classification network of the Prolexs landlord-tenant system.

The Prolexs shell itself is independent of any representation technique and inference mechanism. Knowledge, which is separated from the inference engine, can be inserted as needed. To deal with the results offered by the various inference mechanisms, a truth maintenance facility and an explanation facility is provided. The communication between the knowledge groups,[10] each with its own inference engine, takes place via a black board. Detailed descriptions of the Prolexs system can be found in [Walker & Van den Berg 1988; Walker *et al.* 1989, 1991; see also Walker 1992].

4.6. COMPARING THE FOUR LAYER MODEL TO THE MODELS 1,2 AND 3

The four layer model for constructing legal knowledge bases proposed in this paper is based on an integration of the models described 3.1, 3.2 and 3.3. The basic ideas behind these models are reflected in the four layer model. However, in an attempt to overcome some of the disadvantages of these models, I disconnected the structure dictated by the domain from the knowledge. This domain structure was put at a higher level (level 4). Also I enlarged the perspective. In this way the method used in each of these models forms only one of several possibilities to consult the knowledge. However, it has not yet become fully clear what the final influence of model 3 has been, as this model has not

[9] As was indicated before, the current version of the Prolexs system is based on a earlier version of this model [Oskamp 1989].

[10] A knowledge group is a cluster of knowledge represented in the same representation formalism. In the current version of the Prolexs system each knowledge source is represented in the same way. This is not an optimal way of knowledge representation. The present research concentrates, among other things, on finding better ways of representing each form of knowledge.

explicitly been discussed before. Notwithstanding, it does have its influence. For instance, in the four layer model it is possible to examine the possibility of solving a problem by using simple legal rules while using metaknowledge from the third and fourth layers. Efficient expert systems should first try this possibility because one shouldn't take the hard way when it is possible to do it in a simple way. If this cannot be achieved, or in Gardner's words 'when the rules run out' [Gardner 1987, p. 33], the system will have to be able to proceed and look beyond the (simple) rules. In fact, this is the phase which makes expert systems really interesting. Basically, it is rather easy to develop an expert system for a problem that can be solved by (simple) rules. Only when we are talking about exceptions, including the hard cases, it is necessary to get away from these rules of thumb. Gardner tries to achieve this by using different knowledge, mainly based on natural language or case law for solving problems coming, for instance, from open-textured concepts. In the four layer model the system will check whether these problems can be solved by consulting further knowledge from the different knowledge sources. This can also include knowledge coming from natural language, which can be part of the knowledge source 'expert knowledge'. At the same time, the four layer model is much more general than Gardner's. For when one goes from a small and well-described domain such as Gardner used, one has to cope with problems related to the size of the knowledge base, and closely connected to that, with problems related to control and maintenance. The networks, proposed by Gardner, would probably become too large. This will make it unclear which knowledge is implemented and where.

4.7. THE GENERAL NATURE OF THE LAYERS 1, 2 AND 3

The most important advantage of the four layer model is that the knowledge will have to be implemented only once. A second advantage is that the use of layers in the system opens the possibility to (re)use a substantial part of the knowledge and of all references, when the task of the expert system will be changed. In fact, one could say that only the fourth layer is specifically connected to the task. The knowledge in the first layer is represented as objectively as possible and should as such be reusable for different tasks. The reference file in the second layer, provided that it is complete, will not change when the task changes. This also goes for the general legal metaknowledge in the third layer. There we find general principles that are valid for every legal task. Only in the fourth layer task-related knowledge is found. This knowledge will definitely have to be altered when the task changes.

The previous section is written in a very prudent way. The reason for that is that there are a number of 'ifs' to the statements on the general nature of the three lowest layers. For one, one would really have to be able to represent the different kinds of knowledge as closely related to the original structure as possible (but see Section 5.6.). A lot of research is necessary on this issue, chiefly because the expert system must still be able to reason with this knowledge in an appropriate way. This may depend on the task of the system [cf. Breuker & Wielinga 1988]. Whether that wil be the case is also still a point of research.

There is another snake in the grass. The generality of the first layer, and consequently of the second layer, can only be achieved when all legal and related non-legal knowledge referring to a specific domain is inserted into the knowledge base. Only that would make it possible to switch tasks without affecting the three lowest levels. However, it will be more likely that the knowledge will be gathered and selected in relation to the task and the demands of the users of the system. This is already necessary on purely practical grounds.

For instance, if all published case law were inserted, the knowledge base would become so large that the response times will become unacceptably long. Furthermore, not all published case law would be relevant [cf. Franken 1983, p. 20]. Therefore, although part of the knowledge in the first three layers will be reusable when the task of the expert system changes, it will not be sufficient to alter only the knowledge in the fourth layer.

5. Representation of Knowledge

5.1. INTRODUCTION

This chapter pays attention to the format in which the knowledge has to be represented in the system. The best way for doing justice to the nature of the knowledge is to represent it in a format that is as close as possible to its original format. At the same time one can thus distinguish between the various ways in which knowledge is used and between the various kinds of knowledge. This principle of isomorphism has been urged by others [e.g. Bench Capon 1989, p. 40; Bench Capon & Coenen 1991, p. 62; Karpf 1989]. To accomplish this I think that knowledge must be expressed according to its purpose. This involves employing different representations of one knowledge source. However, forcing knowledge into a format that is not really close to its original format, creates the risk that too much information will be lost. Thus the interpretation of the knowledge might not come from the domain and task – in my opinion the only acceptable reasons for interpretation and even then only when they are sufficiently supported by the expert – but from the need to fit the knowledge into a particular representational format. This would affect the principle of isomorphism [Prakken & Schrick 1991]. It may even affect the quality of the expert system.

In earlier publications on this subject I worked on the assumption that it would be possible to use a single representation formalism to represent a whole knowledge source [Oskamp 1989, p. 195]. However, further research made it clear that the type of knowledge sources does not necessarily dictate the most appropriate representation formalisms. In addition to the nature of knowledge one must also consider the use to which the knowledge is put.

With the exception of the metaknowledge source we find in all knowledge sources at least two kinds of knowledge: directive and declarative. It is necessary to further refine these groups to be able to find, or develop, appropriate representation formalisms. Since this is still the subject of research within the Prolexs project I can only offer some tentative suggestions on the 'best' representation formalisms for different kinds of knowledge.

5.2. LEGISLATION

In the knowledge source 'legislation' I distinguish three different manifestations of knowledge. Directive knowledge gives rules. Descriptive knowledge gives descriptions and explanations of concepts. Terminological knowledge gives additional information with which the system will not reason.

5.2.1. Directive Knowledge

The directive knowledge in legislation consists of rules that are mostly represented as IF THEN production rules in expert systems [Nieuwenhuis 1989, p. 48]. At first sight it looks like the rules of statutes are rather structured and are more or less formalized [Schauss 1988, p. 36–37]. When a lawyer has defined a legal problem, she will always check the legal rules that refer to the case at hand. In statutory based systems these are, for the most part, legislative rules. Next, she will try to apply these rules to the problem.

Searching for those rules is often an action that most lawyers perform without thinking. Yet, it is a real action and absolutely necessary whenever a legal provision can be found in the domain [cf. Guibourg 1986, p. 70]. A rule-guided expert system would take similar actions: it compares the facts it has obtained from the user with the conditions in the rules. Only at that moment will it become clear, both for the human lawyer and for the expert system, whether it will be possible to match the facts and the conditions of the rules. Often it will be necessary to bring facts and conditions more closely together through interpretation, for which other knowledge is needed. Such an interpretation can, for instance, be made by comparing the facts and/or the circumstances of the case with knowledge that gives an interpretation of the conditions in the rule.

Thus, it can be argued that directive knowledge can best be represented in rules. Yet, it should be added that research has made it clear that even for legislative knowledge this is often not as simple as it looks [e.g. Sergot *et al.* 1986; Nieuwenhuis 1987, p. 29]. In my opinion this indicates that there is a need for further research towards more refined and 'better' representation formalisms. Those would have to be more specific and more dedicated to the various kinds of knowledge to be found in the directive knowledge group which encompasses legislation [Karpf 1989, p. 429]. Appropriate representation formalisms developed from this research will probably be based on rule-like structures.[11] However, I do not exclude that some kinds of directive knowledge can be put in 'ordinary' rules without many problems.

5.2.2. Declarative Knowledge

Part of the knowledge from legislation is of a declarative nature. Here we often find explanations of concepts. It is not so easy to express this kind of knowledge in rules.

[11] Such a research project began as a part of the Prolexs project in the autumn of 1992.

Consider the following example, taken from legislation on the Dutch landlord-tenant law (art. 1.3 Huurwet).

This statute contains the following:

a. price
the constellation of the obligations, which is accepted by the tenant towards the landlord with or on account of rent and lease;

b. rent
the price which is due on account of rent and lease for merely using a real property;

c. built real property
a building, or part thereof, if this part forms an independent working accommodation, all this with its normal immovable appurtenances.

Irrespective of the many vague concepts (not only due to the translation) contained in this article it will be very difficult to force this knowledge into a rule-like representation. Knowledge like this is meant to give meaning to the concepts that are to be used in that specific part of legislation. Even though, as is the case here, they often raise more questions than they answer. Putting such knowledge into rules would often result in the creation of meaningless rules so one should try to find other forms of representation. Frames and object-oriented programming could, for instance, be considered as a starting point for both the search for, and development of, such techniques.

5.2.3. Terminological Knowledge

The third type of knowledge we find in statutes is terminological knowledge. An example of the use of terminology is a phrase that often occurs in Dutch statutes and that refers to the responsible minister, for instance in the remaining part of art. 1.3 Huurwet:

d. Our Minister
Our Minister in charge of the care for public housing.

It is not difficult to introduce such knowledge in an expert system. It can be expressed by rules, frames or otherwise. The system will not reason with terminological knowledge, but it may have use it to specify the concept it refers to in the output. Such knowledge will not necessarily be part of the knowledge base, but could also be implemented in a definitional database [cf. Hage 1988].

5.3. CASE LAW

Case law can be used in various ways, dependent on task, domain and legal system. In the Dutch legal system case law has three main functions: giving (additional) rules (directive knowledge), giving definitions and explanations of vague rules and open textured concepts (declarative knowledge) and giving examples.

5.3.1. Directive Knowledge

Case law, and especially case law delivered by the High Court, may formulate rules that can become generally applicable in the domain and so become a rule of law. Although in the Dutch legal system the lower courts are not legally bound to the judgements of the higher courts, they will follow them most of the time, especially the judgements of the High Court. Furthermore, rules can be extracted from constellations of cases which exhibit consistent patterns. These rules will be placed, however, either in the knowledge source 'legal literature' or in the knowledge source 'expert knowledge', since directive knowledge in this context refers only to identifiable cases.

It will depend on the case itself whether it can be solely represented in a rule-like structure, just like directive knowledge in legislation or whether additional representation formalisms are necessary. When the case is only important because of the rule it gives, it will not have to be implemented fully. It will be sufficient to implement the rule of law and an identification of the case. This can be realized in a rule like structure. If the case also has other aims, either interpretation or giving an example, a mix of representation formalisms may be necessary. In general it can be said, however, that all directive knowledge in case law is best represented in a rule-like formalism.

5.3.2. Declarative Knowledge

The importance of a specific judgement can also lay in the explanation it gives of concepts or rules from, for instance, legislation. The judgement will then be inserted in the knowledge base to describe the explanation of the concept and possibly the circumstances of the case. Thus the possibility is offered to make a comparison with the circumstances of the case at hand. For this purpose only this description, and perhaps the circumstances of the case, will have to be put into the knowledge base. Of course, in such a situation we have to cope with the question how much of the explanation of the concept has to be reflected. For instance, will the circumstances that have led to the explanation, also have to be considered, and if so in detail? Or could these circumstances be reflected in a more general way? Both possibilities have advantages and disadvantages. When the knowledge engineer, in close cooperation with the expert, abandons the case while representing the explanation of the concept indicated, she will generalize the circumstances of the case. She will also generalize the explanation of the concepts as given in the case. This generalization is a form of interpretation. However, when she does take the circumstances into account, a lot more knowledge has to be represented resulting in an expansion of the knowledge base.

Declarative knowledge in case law will also have to be represented in a format that reflects the original format as closely as possible. Just like it was the case for legislative knowledge this will hardly be possible when using a rule-based formalism. We should rather think of formalisms based on frames or on object-oriented programming. Schlobohm and Waterman [1987, p. 21] give an example of frames that give such a general explanation of a specific concept. The content of their frames is comparable to the declarative knowledge which I describe. Although these frames are quite simple and are only meant to give an explanation of the output of the system, it led them to remark:

> We feel the use of a frame mechanism provides a more natural way of handling definitional explanations and facilitates comparing and contrasting alternative estate plans.

5.3.3. The Example-Giving Function

Case law is used in an example giving role when one points out the similarities with, or the differences from specific judgements with the case at hand in order to convince someone, usually the judge, about one's position. In this role case law is not so much used for explanation as for qualification. One tries to prove, with the help of prior judgements that one's case does or does not fall under a specific legal rule. A lot of details are involved. Those details are often related to each other. The decision whether the case qualifies under a certain legal rule is usually taken considering the combination of those details. Therefore cases that are to be used as examples can hardly be compressed in rules.

I have to point out here that the use of case law in the Dutch system is slightly different from its use in common law systems. The oral arguments are less important than in a common law system [Rissland 1989, p. 111]. For instance, a Dutch lawyer can argue that a certain court has already dealt with a similar case, and use this case as an example. Although it will improve her position she cannot derive any rights from that judgement. Case law is therefore used in a less intensive way than in the common law systems.

Giving an example is a characteristic function of case law which has no equivalent in legislation. To use case law in this way it may be necessary to argue with the case as a whole, or at least with a significant part of the case. The circumstances of the case play an important part. This will put a very large claim on the knowledge base. Since case law will rarely be used in this way in the Dutch legal system one can not justify the implementation of case law for this purpose. It is therefore not surprising that most case-based reasoning projects are found in the United States. The HYPO-system, for instance, gives a good example of the way in which to represent knowledge for such purposes [Ashley 1988].

5.4. LEGAL LITERATURE AND EXPERT KNOWLEDGE

The nature of the knowledge in the sources 'legal literature' and 'expert knowledge' is rather diverse, as both contain many different kinds of knowledge. We find knowledge that resembles legislation in directives and guidelines, or case law which has instances of hypothetical cases described in literature or unpublished cases known to the expert. Furthermore, in both knowledge sources we find other forms of directive knowledge, such as rules that are extracted from a number of cases by the expert. We also find declarative knowledge such as explanations. The general rules referring to representation formalisms, described in the previous sections can be applied to these forms of knowledge. Directive knowledge can be best expressed in rule-like representations. For declarative knowledge frame-like representation formats and forms of object oriented programming can be suitable formats. Knowledge from the sources 'legal literature' and 'expert knowledge', which is based on a constellation of other knowledge, may benefit from the use of techniques like neural networks [Van Opdrop & Walker 1990; Van Opdorp *et al.* 1991]. For these knowledge sources the optimal representation formalism requires further research also applicable for other forms of knowledge that I have not yet indentified.

5.5. METAKNOWLEDGE

5.5.1. The Various Forms of Metaknowledge

Before discussing formats for the representation of metaknowledge I will first describe the most important forms of metaknowledge for legal expert systems.

5.5.1.1. *The Hierarchy of the Various Types of Knowledge.* Consider the following examples of legal metaknowledge, taken from the Dutch legal system: the hierarchy 'legislation – case law – legal literature', or a hierarchy like 'High Court – Court of Appeals – District Court – Police Court'. Next to this, principles like *lex superior derogat legi inferiori, lex specialis derogat lex generalis, lex posterior derogat legi priori* are examples of this kind of metaknowledge.

5.5.1.2. *The Overview that an Expert Has of a Specific Domain.* This enables him to choose quickly and effectively which knowledge he has to use to solve the problem at hand. This part of legal metaknowledge can be introduced by composing classification networks of the domain. In such a classification network the various parts of a specific domain are shown. It also indicates their place within the domain and the various relations between each other are made clear. Figure 1 gives an example of the classification network made for a small part of the Prolexs system. [Walker *et al.* 1988, p. 13, 1989, p. 250]

5.5.1.3. *Rules of thumb, rules containing strategy, and application rules to solve problems in the domain.* These metarules indicate which statutory rules or which case law can be used effectively in a specific situation. It can also indicate in which sequence this knowledge can be consulted most significantly. In some situations a concept from a statutory rule will best be explained using case law. In other cases legal literature can be used more effectively. [Bellairs 1987, p. 253] This type of metaknowledge should typically come from the expert. Another published example of this form of metaknowledge is transition law.

An example referring to the rules of strategy is the following. A landlord who wants a tenant to leave has five different grounds for giving notice. To reach the conclusion that the landlord has given notice in a legally valid way, the Prolexs system will have to reach one of five goals[12] which embody the legally valid grounds. The system could, by asking the necessary questions, reach those goals one at a time. However, that is not what

[12] The five grounds are 'bad behaviour of the tenant', 'cancellation of the rent contract based on the zoning scheme', 'termination based on the necessity for own use', 'offering a new contract for the rented object' and 'expiration of the term'.

the expert would do. He would, for instance, start to try to reach the goal that is most common. In the Prolexs system that would be the goal of necessity for own use. A different strategy could be to start with the goal that can be reached most easily. Another strategy would be to use some 'key-questions'. These are questions designed to check whether a characteristic fact is present to reach a specific goal relevant to the case. Based on the answers of these questions, the system will than select a specific goal. The system could ask, for instance, whether the tenant has caused any specific problems. A confirmative answer would indicate that it might be possible to end the contract because of bad behaviour of the tenant. Defining the strategies that are most suitable to domain and task of the system is the task of the domain expert.

5.5.1.4. *The Evaluation of Knowledge.* By this I mean predictive knowledge that makes it possible to estimate the value and validity of a specific solution. The attachment of certainty factors is an example of this form of metaknowledge.

5.5.1.5. Finally, the selection of the knowledge to be implemented into the system is a form of metaknowledge as well because it says something about the value of the knowledge implemented and implicitly of the knowledge not implemented. This form of metaknowledge becomes only clear after selection of the knowledge.

The sources of the various forms of metaknowledge vary. A part of metaknowledge can be extracted from the written sources. In the Dutch legal system it could come from the 'Wet algemene bepalingen' (General Provisions Act) or from specific transition laws. Some forms of metaknowledge are known by every lawyer, while other forms are characteristically known only by lawyers with expertise in a particular area of law.

5.5.2. Directive Knowledge

The knowledge described in 5.5.1.1. and 5.5.1.3. are forms of directive knowledge. However, one has to realize that the directive knowledge indicated in 5.5.1.1. is situated at the third level of the model and the knowledge indicated in 5.5.1.2. at the fourth. These are different levels from the directive knowledge in the other four knowledge sources. Their purpose is to direct the application of the knowledge from the other sources. Although for directive metaknowledge a rule-like representation formalism will probably be the most appropriate, it still has to be identifiable as metaknowledge. One should not try to combine this metaknowledge (via chaining) with the knowledge from the other knowledge sources at the first level.

5.5.3. Knowledge Giving an Overview

The knowledge described in 5.5.1.2. gives an overview. We will find this type of knowledge mostly at the fourth level where, for instance, it gives an overview of the domain. The representation formats for this knowledge can well be based on (semantic) nets, for instance in the form of classification networks such as are also developed for conceptual information retrieval [cf. Hafner 1981].

5.5.4. Selection of the Knowledge

In Section 5.5.1.5. I indicated that the selection of knowledge is a form of metaknowledge. A characteristic of legal systems is that a substantial part of the knowledge can be obtained from published sources. Although this may facilitate gathering all possibly relevant knowledge, it also makes it difficult to select this knowledge. As in non-legal domains where gathering the knowledge constitutes a bottleneck the bottleneck of selection will be even more pressing in the legal domain.

Literature hardly pays attention to how legal knowledge has to be gathered and selected, nor to the question by whom this has to be done. In the opinion of [Philipps 1986, p. 703] the rules have to be introduced into the system by the same people who use the rules in their daily work. Greenleaf *et al.* [1987, pp. 11–12] do not use a knowledge engineer for practical reasons: the knowledge engineer might wrongly see himself as the expert. Secondly, they believe that the intervention of a knowledge engineer in research projects would be too expensive. Susskind first states that the role of the expert can be restricted to tuning the system. By this he means experimenting with the system and refining it. [Susskind 1987, p. 59]. Later, after developing a practical expert system, he correctly concludes:

> ... experts must be involved throughout the construction of the academic legal knowledge. Their understanding, interpretation and modelling of even the primary and secondary sources is likely to be of a sophistication and nature quite unlike that of the competent legal researcher [Susskind 1989, p. 25].

It is essential that, along with a knowledge engineer, a domain expert be involved with the development team of a legal expert system from the start. He must play a critical role in selecting the knowledge. Considering the complex issues of knowledge acquisition and selection in developing legal expert systems, a knowledge engineer cannot work without a domain expert.

In the selection process one could distinguish two questions. The first is what knowledge should be part of the knowledge base? The second what knowledge could be part of the knowledge base? The answers to these questions are related to the knowledge source. Referring to the knowledge source 'legislation' one could argue persuasively that all relevant knowledge should be part of the knowledge base.

Methods of limiting the size of this knowledge source make this feasible. Limits of relevancy could be set by the narrowing the domain and the task of the system. A more critical selection process is required for the knowledge sources 'case law' and 'legal literature'. Here the limits set by domain and task will not reduce the knowledge sufficiently. Thus the participation of the expert is even more important because the selection of knowledge is such a subjective matter. By selecting the knowledge the expert puts his personal mark on the system.

The expert will also put his mark on the system with the gathering and selection of expert knowledge and metaknowledge. Some general metaknowledge will be found in books, as this is the kind of knowledge that is taught to students. For the remaining part we will find the knowledge mainly 'in the head of the expert'. This kind of knowledge is also used in expert systems in non-legal domains. The knowledge engineer will try to extract the expert knowledge from the expert by using the various methods of knowledge elicitation described elsewhere in the literature on expert systems.

5.6. INTERPRETATION WITH REPRESENTATION

The content of the original sources of law must be clarified before it can be represented and implemented in an expert system. As long as computers can not process natural or legal language the knowledge bases of legal expert systems will encompass the subjective interpretations of the domain expert no matter how hard one might try to eliminate such interpretations [cf. Allen & Saxon 1991, pp. 53–61; Bourcier 1987, p. 188]. The transposition of texts asks for making choices, as long as the original texts are not formulated in exactly the form in which they are to be represented [contra Nieuwenhuis 1989, pp. 48–62]. These choices can easily be made explicit by using hypertext to juxtapose the original source with the represented form [Greenleaf *et al.* 1991].

In the previous sections I have already stressed the importance of finding representation formalisms that come as close to the original knowledge as possible, in order to diminish the interpretation associated with representation. I have indicated that a lot of research still has to be carried out to develop the 'best' representation formalism for the various kinds of knowledge. The development of such representation formalisms would not only facilitate the representation process as such but would also make possible the representation of the knowledge in a more isomorphic form.

I agree with the advantages that an isomorphic representation would offer [cf. Karpf 1989; Bench Capon 1989; Bench Capon & Coenen 1991]. I believe that the model proposed in Section 4 is another step in that direction, but as I said before to accomplish a truly isomorphic representation one still needs a lot of research [Karpf 1989, p. 429]. However, it is far from certain that this can be realized. It may turn out in the end that it is possible to give an isomorphic representation only for the rule-giving part of legislation. It may also be possible to isomorphically represent similar rule-giving knowledge in other sources. However, I think that it will be almost impossible to realize isomorphism for declarative and example giving knowledge. In case law, for instance, it will hardly ever be possible to represent the whole case. It will be necessary to define and limit the knowledge to the important features of a case [cf. Ashley 1989, p. 93]. Any shortening or summarizing of a case means interpretation.

One also has to realize that this point of view makes very severe demands on the tools that are used to develop the expert system. They will have to offer a variety of representation formalisms and be able to reason adequately with these various forms of knowledge representation. The Prolexs shell already has the features that are necessary for such a tool. In his thesis, Walker extended the Prolexs shell for these purposes. He further developed it to the EXPANDER architecture that stands for Expert System Architecture allowing Non-homogeneous Distributed Knowledge Representation. The knowledge can be kept in separate and autonomous modules, each with its own representation formalism and reasoning methods. It is up to the user of the architecture to choose which representation formalisms and reasoning methods to use or to develop. Then those can be inserted into the EXPANDER architecture. The architecture also implements a protocol which specifies how to share knowledge, provide explanation, solve contradictions, search efficiently, detect inconsistencies, etc. [Walker 1992].

6. Concluding remarks

The knowledge that is used for applying the law can be found in five separate knowledge sources: legislation, case law, legal literature, expert knowledge and legal metaknowledge. The first four knowledge sources contain the basic knowledge on which law application is founded. It can often be found in written and published form. Only in the knowledge source 'expert knowledge' do we find some non-written knowledge, mainly consisting of facts. The fifth, legal metaknowledge, is a knowledge source of a different nature. It consists knowledge that directs the use of the knowledge in the other four sources. The form of this knowledge can be very diverse. It consists of rules of thumb and rules of strategy, but it also reflects the experience of a lawyer, for instance in the overview he has of a certain domain. Part of metaknowledge consists of skills that lawyers have gained during their studies and in legal practice.

The specific nature of metaknowledge is an important reason to distinguish several layers in a knowledge base. In the model presented Section 4, four layers are distinguished. The four basic knowledge sources are placed at the first level. Each unit of knowledge should only be represented once, and be identifiable to its original knowledge source. The second layer of the system is an optional layer and consists of a list of all cross references among the knowledge. The third layer contains the general legal metaknowledge, i.e. legal metaknowledge that is not related to a specific domain. The domain related metaknowledge is situated at the fourth and highest level. This four layer model makes it possible to build flexible and maintainable knowledge bases. The knowledge is identifiable to origin. The relations to other knowledge are clear. The reasoning strategies are not part of the knowledge itself, but situated at a level above the knowledge. This means that the system can use each knowledge unit for various purposes. Also strategies can easily be changed, should that be necessary. In addition the model is transparent, as the strategies as such are explicit, and placed at a separate level.

The ordering in the four basic knowledge sources can only been seen as a first ordering. Each knowledge source may contain different forms of knowledge, as well as hierarchies of knowledge within the source itself. An example is the distinction that can be made between directive and declarative knowledge in each of the four knowledge sources. However, other differences may exist. These differences can depend on domain and task. The conclusion that within each knowledge source several types of knowledge can be distinguished is important in relation to the representation of this knowledge. One of the assumptions of this model is that the knowledge should be implemented only once irrespective of its purposes and no matter how often it is used. This implies that the representation of the knowledge should be as close to the original text and form of the knowledge as possible. Interpretation should be avoided as much as possible at this level, although that will not always be feasible. Therefore one should try to develop representation formalism that come as close as possible to the original format and purpose of the knowledge. Existing formalisms will not always be suitable, although they can be a starting point.

References

Allen, L. E. & Saxon, C. S. 1991. More IA needed in AI: Interpretation Assistance for Coping with the Problem of Multiple Structural Interpretations. In Proceedings of the *Third International Conference on Artificial Intelligence and Law*, 53. Oxford: Association for Computing Machinery.

Ashley, K. D. 1989. Toward a Computational Theory of Arguing with Precedents. In Proceedings of the *Second International Conference on Artificial Intelligence and Law*, 93. Vancouver: Association for Computing Machinery.

Ashley, K. D. 1988. Modelling Legal Argument: Reasoning with Cases and Hypotheticals. Ph.D. Dissertation, COINS Technical report 88–01, University of Massachusetts, Amherst.

Bench Capon, T. & Coenen, F. 1991. Exploiting Isomorphism: Development of a KBS to Support British Coal Insurance Claims. In Proceedings of the *Third International Conference on Artificial Intelligence and Law*, 62. Oxford: Association for Computing Machinery.

Bench Capon, T. J. M. 1989. Deep Models, Normative Reasoning and Legal Expert Systems. In Proceedings of the *Second International Conference on Artificial Intelligence and Law*. Vancouver: Association for Computing Machinery, 37.

Berman, D. H. & Hafner, C. 1987. Indeterminacy: A Challenge to Logic Based Models of Legal Reasoning. In *Yearbook of Law Computers and Technology*. Volume 3, Butterworths, London, 1.

Biagioli, C., Mariani, P. & Tiscornia, D. 1987. ESPLEX: A Rule and Conceptual Model for Representing Statutes. In Proceedings of the *First International Conference on Artificial Intelligence and Law*, 240. Boston: Association for Computing Machinery.

Bing, J. 1987. Designing Text Retrieval Systems for Conceptual Searching. In Proceedings of the *First International Conference on Artificial Intelligence and Law*. 43. Boston: Association for Computing Machinery.

Bourcier, D. 1987. Ces systèmes dits experts, ou comment passer du droit, in *Informatique et droit: 20 ans d'experience*, ADIJ, vol. IV, Paris, 183.

Breuker, J. & Wiellinga, B. 1988. Models of Expertise in Knowledge Acquisition. Memorandum 103 of the VF-project 'Acquisition of expertise', University of Amsterdam, Department of Social Science Informatics.

Capper, P. & Susskind, R. E. 1988. *Latent Damage law – The Expert System*. London: Butterworths.

Franken, H. 1983. Jurist en computer: theoretische achtergronden. In Wild, A. H. de, Eilders, B. (eds.), *Jurist en computer*, 13. Deventer: Kluwer.

Gardner, A. 1989. Representing Developing Legal Doctrine. In Proceedings of the *Second International Conference on Artificial Intelligence and Law*, 16. Vancouver: Association for Computing Machinery.

Gardener, A. v.d. L. 1987. *An Artificial Intelligence Approach to Legal Reasoning*. Cambridge, MA: MIT Press.

Goldman, S. R., Dyer, M. G. & Flowers, M. 1987. Precedent-Based Legal Reasoning and Knowledge Acquisition in Contract Law: A Process Model. In Proceedings of the *First International Conference on Artificial Intelligence and Law*, 210. Boston: Association for Computing Machinery.

Greenleaf, G., Mowbray, A. & Tyree, A. L. 1987. Expert Systems in Law: The Datalex Project. In Proceedings of the *First International Conference on Artificial Intelligence and Law*, 9. Boston: Association for Computing Machinery.

Greenleaf, G., Mowbray, A. & Thyree, A. L. 1991. The Datalex Legal Workstation – Integrating Tools for Lawyers. In Proceedings of the *Third International Conference on Artificial Intelligence and Law*, 215. Oxford: Association for Computing Machinery.

Guibourg, R. A. 1986. Sequences and Levels in the Legal System, in Martino, A. A. & Socci Natali, F. (eds.), *Automated Analysis of Legal Texts*, 69. Amsterdam/New York/Oxford/Tokyo: North Holland.

Hafner, C. D. 1981. *An Information Retrieval System Based on a Computer Model of Legal Knowledge*. Ann Arbor: UMI Research press.

Haft, F., Jones, R. P. & Wetter, Th. 1987. A Natural Language Based Expert System for Consultation and Tutoring – The LEX Project. In Proceedings of the *First International Conference on Artificial Intelligence and Law*, 75. Boston: Association for Computing Machinery.

Hage, J. C. 1988. Non-inferentiële kennis in juridische expertsystemen. Proceedings of the *First Dutch AI Conference*, 65

Hayes-Roth, F., Waterman, D. A. & Lenat, D. B. (eds.). 1983. *Building Expert Systems*. Reading, Ma: Addison-Wesley.

Karpf, J. 1989. Quality Assurance of Legal Expert Systems. In Martino, A. A. (ed.), Pre-Proceedings of the *III International Conference on LOGICA, INFORMATICA, DIRITTO*, Vol. I, 411.

Kracht, D., Smiths, J. M. & Weusten, M. C. M. (1988), Advisory Systems for Legal Questions. In Herrestad, H. & Maesel, D. S., *Five Articles on AI and Legal Expert Systems*, COMPLEX 13/88, Universitetsforlaget, Oslo, 34.

McCarty, L. T. 1987. Intelligent Legal Information Systems: An Update. Law and Computers, *The Journal of the Law and Computers Association of Japan* 5: 196.

Nieuwenhuis, M. A. 1989. *Tessec: een expert systeem voor de Algemene Bijstandswet*. Deventer: Kluwer.

Nieuwenhuis, M. A. 1987. *Constructie en instandhouding van een zuiver kennisbestand, Kennissystemen*, jaargang 1, nr. 2, 25.

Noortwijk, C. 1990. Criteria in the Juricas Project. In Kracht, D., Vey Mestdagh, C. N. J. & Svensson, J. S. (eds.), *Legal Knowledge Based Systems, an Overview of Criteria for Practical Validation and Use*. Lelystad: Koninklijke Vermande.

Opdorp, G. J. van, Walker, R. F., Schrickx, J. A., Groendijk, C. & Berg, P. H. van den. 1991. Networks at Work: A Connectionist Approach to Non-Deductive Legal Reasoning. In Proceedings of the *Third International Conference on Artifical Intelligence and Law*, 278. Oxford: Association for Computing Machinery.

Opdorp, G. J. van & Walker, R. F. [1990]. A Neural Network Approach to Open Texture. In Kaspersen, H. W. K. & Oskamp, A. (eds.), *Amongst Friends in Computers and Law*, 279. Deventer: Kluwer.

Oskamp. A. 1990. *Het ontwikkelen van juridische expertsystemen. Een theoretische beschouwing*. Deventer: Kluwer.

Oskamp, A., Walker, R. F, Schrickx, J. A. & Berg, P. H. van den 1989. PROLEXS, Divide and Rule: A Legal Application. In Proceedings of the *Second International Conference on Artificial Intelligence and Law*, 54. Vancouver: Association for Computing Machinery.

Oskamp, A. 1989. Knowledge, representation and Legal Expert systems. In Vandenberghe, G. (ed.), *Advanced Topics in Computer/law*, 195. Deventer: Kluwer.

Oskamp, A., 1986. Expertsystemen en hun toepassing in het recht, Ars Aequi, Special Rechtsinformatica, 692.

Philipps, L. 1986. Using an Expert System in Testing Legal Rules. In Martino, A. A. & Socci Natali, F. (eds.), *Automatic Analysis of Legal Texts*, 703. Amsterdam/New York/Oxford/Tokyo: North Holland.

Prakken, H. & Schrickx, J. 1991. Isomorphic Models for Rules and Exceptions in Legislation. In Breuker, J. A., Mulder, R. V. De, Haje, J. C. (eds.) *Legal Knowledge Based Systems, Legal Reasoning*, 17-Lelystad: Vermande.

Prakken, H. 1991. A Tool in Modelling Disagreement in Law: Preferring the Most Specific Argument. In Proceedings of the *Third International Conference on Artificial Intelligence and Law*, 165. Oxford: Association for Computing Machinery.

Purdy, R. D. 1989. Knowledge and Tools in Building Grandjur 1.1. In the Proceedings of the *Second International Conference on Artificial Intelligence and Law*, 201. Vancouver: Association for Computing Machinery.

Rissland, E. L. 1989. Dimension-Based Analysis of Hypotheticals from Supreme Court Oral Argument. In Proceedings of the *Second International Conference on Artificial Intelligence and Law*, III. Vancouver: Association for Computing Machinery.

Schauss, M. (ed.). 1988. *Systèmes experts et droit*. Brussels: Story Scientia.

Schlobohm, D. A. & Waterman, D. A. 1987. Explanation for an Expert System that Performs Estate Planning. In the Proceedings of the *First International Conference on Artificial Intelligence and Law*, 18. Boston: Association for Computing Machinery.

Sergot, M. 1991. *The Representation of Law in Computer Programs: A Survey and Comparison. Complex* 1/91, Tano A. S. , Oslo.

Sergot, M., Cory, T., Hammond, P., Kowalski, R., Kriwaczek, F. & Sadri, F. 1986. Formalisation of the British Nationality Act, in *Yearbook of Law Computers & Technology*, vol. 2, Butterworths, 40.

Susskind, R. E. 1987. *Expert Systems in Law*. Oxford: Claredon Press.

Susskind, R. E. 1989. The Latent Damage System: A Jurisprudential Analysis. In the Proceedings of the *Second International Conference on Artificial Intelligence and Law*, 23. Vancouver: Association for Computing Machinery.

Walker, R. F. 1992. An Expert System Architecture for Heterogeneous Domains. A Case Study in the Legal Field. Thesis Vrije Universiteit Amsterdam, A-D Druk B. V., Zeist.

Walker, R. F., Oskamp, A., Schrickx, J. A. Opdorp, G. J. van & Berg, P. H. van den. 1991. PROLEXS: Creating Law and Order in a Heterogeneous Domain. *Int. J. Man-Machine Studies* 35.

Walker, R. F., Zeinstra, P. G. N. & Berg, P. H. v.d. 1989. A Model to Model Knowledge About Knowledge. In Vandenberghe, G. (ed.), *Advanced Topics in Computer/Law*, 235. Deventer: Kluwer.

Walker, R. F. & Berg, P. H. van den 1988. Prolexs, and Object Oriented Legal Expert System. In Herrestad, H. & Maesel, D. S., *Five articles on AI and legal expert Systems*, COMPLEX 13/88, Universitetsforlaget Oslo, 8.