Open Texture and Ontologies in Legal Information Systems

T.J.M. Bench-Capon and P.R.S. Visser
LIAL - Legal Informatics At Liverpool,
Department of Computer Science,
The University of Liverpool, Liverpool,
L69 7ZF,
UK.

Abstract

The problem of open texture is long established in legal information systems. It is pervasive and needs to be addressed by any system able to deal with cases which are not entirely straightforward. Many systems have tried to deal with this problem, and normally they are divided into rule based, case based and statistical approaches. In this paper we look at these systems from a different perspective, that of ontologies. We first descibe what ontologies are, and the roles that they can play for legal information systems. Next we review a range of approaches to open texture, paying particular attention to the domain conceptualisation which underlies them. We then discuss two ontologies which have been proposed for use in legal information systems, with particular reference to their treatment of open texture. Finally we identify a number of choices which help determine the approach to open texture, and which are orthogonal to the particular representation technique used.

1. Introduction

1.1 Ontologies and Knowledge Based Systems.

In recent years the desire to provide firm engineering foundations on which to build legal information systems has led to the idea of producing *ontologies* for such systems. In the context of information systems, "ontology" has lost much of its original philosophical meaning. In this context we can follow (Gruber 1995) and define an ontology for a knowledge based system as *an explicit specification of the conceptualisation of its domain*. As such, the ontology makes available many of the assumptions implicit in the way the knowledge base has been constructed. Such an ontology has many similarities with the conceptual schema of a database: it determines the vocabulary than can be used to pose, and will be used to answer, queries to and from the resource. Ontologies have been shown to have benefits in a number of areas:

- knowledge sharing;
- knowledge reuse;
- verification and validation;
- domain theory development;
- knowledge acquisition; process.

1.2 Open Texture in Law

In this paper we will explore the benefits that can be obtained by adopting an ontological perspective with respect to one specific, but problematic, area for the development of legal KBS, namely that of *open texture*. A concept is said to be *open textured* if the conditions for its application are not tightly stated, but instead are left to be resolved by the courts in the light of the circumstances of individual cases. That many legal concepts are open textured is beyond dispute: some would go further and argue that all legal concepts are open textured. Hart (1961), for example, describes open textured concepts in terms of "a core of settled meaning" and a "penumbra of debatable cases" and says that "a penumbra of uncertainty must surround all legal rules".

Open texture presents a severe challenge to the automation of legal reasoning. Hart concludes from the observation that a penumbra of uncertainty surrounds all legal rules that "their application to specific cases in the penumbral area cannot be a matter of deduction". But, it has been argued (e.g. Susskind (1987)) that since deduction is very much the paradigm for the reasoning of knowledge based systems, this argument undercuts the very possibility of applying knowledge based systems to law. The usual response of those whose primary interest has been to build practical systems has been to divide cases into "easy" and "hard" cases (a distinction introduced in Gardner (1987)), the former falling into the core of certainty and the latter into the penumbra of doubt. Having drawn this distinction, knowledge based systems are restricted to the easy cases. But this is not very satisfactory. Apart from the difficulty in separating out easy from hard cases so as to decide which cases can appropriately be placed before the system, much of the essence of legal reasoning seems to reside in the resolution of the hard cases. As an alternative, some have placed a greater onus on the user, requiring the user to resolve all questions of open texture, and confining the role of the system to deducing the deductive consequences of the decisions made by the user. Again this considerably restricts the scope of the system. Thus open texture is a problem that must be confronted by anyone building a knowledge based system in the legal domain, and is in itself an important topic of AI and Law research. Similarly because open texture is so central a part of the legal domain, it must be addressed in any ontology which purports to specify the conceptualisation of that domain.

In this paper we will discuss how ontologies can be used to gain insight into the problem of applying open textured concepts in the context of an automated system. Our discussion will begin by reviewing the variety of attempts to address open texture that have been proposed in existing legal systems. From this discussion we will attempt to draw out the key points in the conceptualisations that underlie these approaches. We will then look at ontologies that have been designed for the legal domain, and in particular how these accommodate open texture. We will then draw these threads together by discussing the key design choices that emerge, and which have been obscured by an over concentration of the representation formalism used. We hope that this will show how taking an ontological view of attempts to tackle this problem can improve our understanding of these approaches, and provide a framework within which we can make useful comparisons and contrasts.

2. Approaches to Open Texture

In this section we will describe some attempts to accommodate open texture in legal information systems. We will describe a range of typical approaches to building legal information systems, and discuss how they handle open texture. We will discuss the approaches in three groups: approaches based on rules, approaches based on cases and approaches based on statistical techniques.

2.1 Rule Based Approaches

First consider the "law as a logic program" approach, typified by the Imperial College British Nationality Act program (Sergot et al 1986). In this approach law is conceived as definitions, and represented as a set of extended Horn Clauses. Each clause is a sufficient condition for the truth of its head, and where there are several clauses for the same head they are taken as jointly necessary. This interpretation, designed to allow negation as failure, means that all definitions are taken as complete. In consequence there can be no room for open texture in the definitions: all open texture will be in the primitives used in the bodies of the clauses, and the onus for resolving it falls on the user, who must say whether or not these primitives apply. With regard to the primitives used in the bodies of the clauses no distinction is made between those which are open textured (e.g. “showed good clause for late claim”) and those which may be thought to be capable of definitive application (e.g. “aged over 65 years”). In a sense, therefore, this conceptualisation excludes open texture: from the perspective of the program, all predicates can be given definite truth values, and it is only when the program is in use - and so users have the possibility to give different answers to similar questions - that an element of open texture occurs.

If we relax the constraint that clauses with the same head jointly provide necessary conditions, we have the situation where at least some predicates are defined in terms of a (possibly incomplete) set of sufficient conditions. Such an approach can be found in Schild (1989). Here the concept is open textured if failure to satisfy any of the sufficient conditions does not mean the concept is inapplicable, but only that its applicability is not established by the facts presented. There may be no difference in the knowledge based used here from that used in the BNA style approach: the difference resides in the interpretation placed on failure to satisfy any of the sufficient conditions.

This approach can be extended, as in Schild (1993), to allow for sufficient conditions for the inapplicability of the concept to be stated. In this case, we may be able to say that the concept is applicable, or that it is inapplicable, or, if none of the sufficient conditions are satisfied, nothing. In one variety of this approach the sufficient conditions for applicability and the sufficient conditions for inapplicability are mutually exclusive. In this case the conditions represent cases within the core of certainty, and the system is silent in the penumbra. If, however, the two sets of conditions are not exclusive, it may be that a sufficient condition for inapplicability *and* a sufficient condition for applicability are satisfied. Here cases in the penumbra are represented by having a line of argument for *both* sides of the question. Now the possibility of conflict has arisen, some mechanism must be supplied to resolve these conflicts. In the simplest case both lines of reasoning are presented and resolution is left to the user. It is, however, also possible to supply some meta principles, such as *lex specialis*, to attempt resolution automatically (e.g. Prakken (1993)). The approach of using meta principles is presented in a highly developed form in Poulin et al (1993). They allow different interpretations of open textured concepts to co-exist in their system, and then use a variety of metarules, based on principles of legal interpretation, to resolve cases of conflict.

Perhaps if we allow conflicts between our rules it is no longer proper to use the term “sufficient conditions”, since they may prove insufficient in the light of a better rule. Perhaps they are now better seen as a “default rule” which hold *ceteris paribus,* but which may be rejected if a “stronger” rule applies.

This leads us to the topic of defeasible rules. Prakken’s work, mentioned above was designed not with open texture in mind, but rather to address the problem of defeasibility, and conflict amongst norms. The similarity lies in the fact that both for defeasible norms and for open texture we have arguments both for and against the application of the concept, which need to be adjudicated between. If we wish we could see rules representing sufficient conditions for the application of an open textured concepts as defeasible rules, and then draw on approaches in the literature to the problem as defeasibility to address open texture also.

One interesting example is provided by the work of Hage and his colleagues (e.g. Hage 1996). One feature of their approach is that the conditions of applicability are construed not as statements. but as *reasons*, and Hage supplies a logic - Reason Based Logic - for reasoning with them. In cases where the answer is not clear, the reasons will conflict, and so the notion of competing pro and contra factors assumes a central role here. The mechanism for deciding which reasons are taken as persuasive is through a relation "outweighs" which exists between reasons. The approach has now become quite sophisticated and mechanisms have been described to allow several reasons to overturn a reason stronger than any of them individually, and to allow for the outweighs relation itself to be supported by reasons and so subject to argument.

The outweighs relation in Hage's work is not quantified, but one could imagine a variation on this approach which did attach quantitative weights to the various reasons.

2.2 Case Based Approaches

Given that the resolution of open texture is in the context of decisions made in individual cases, a natural approach is to represent past decisions, and to attempt to infer the decision in a new case from the decisions made in those past cases. In these approaches the law is not conceptualised as supplying definitions or sufficient conditions, but as consisting of cases.

In the simplest form of case based reasoning a case is conceptualised as consisting of a number of features, one of which is the outcome of the case. Existing cases are represented, and when a new case is presented its features are compared with those of the existing cases: the most relevant of these cases is identified, and its outcome applied to the new case. Different systems are characterised by different approaches to what constitutes a match. It is possible to identify some features as mandatory for a case to be considered matching, for this to be conditional on the presence of other features, and to attach weights to features to give a quantitative measure of similarity (e.g. Kowalski 1991). Once we have a quantitative measure, there are several possibilities: we can simply choose the best precedent, which gives us a conflict free answer; or we can weigh precedents, and even allow weak precedents to accumulate to overcome a stronger one.

In the basic application of case based reasoning, cases, and hences matches, are considered as indivisible units. In the best known Case Based Reasoning system applied to law, HYPO, (most fully described in Ashley 1990), however, the conceptualisation of a case is extended by introducing the notion of *dimensions.* A dimension represents an issue which arose in a case, and is derived from a combination of case features. This allows us to separate out features of a case considered important from the case as a whole. A given feature may contribute to several dimensions, whilst other features (like name) appear in no dimensions and will play no role in determining similarity. Matching is performed on the basis of these dimensions, rather than the features, and the dimensions are used to encapsulate relationships between features, and to distinguish between the significance of various features. In addition dimensions have *direction*, so that in matching not only is closeness considered, but also the direction in which the dimension differs. Thus in Trade Secrets - the domain of HYPO - an important dimension is how many people knew about the secret. The more people that knew, the less likely a case is to succeed. Therefore where a past case decided for the plaintiff has fewer disclosures the match is unaffected, whereas even a small increase affects the closeness of the match. Matching in HYPO is not quantitative: when presented with a new case HYPO forms a case lattice which imposes a partial ordering on past cases according to how the dimensions match.

Matches in HYPO are not quantified. It is, however, possible to incorporate dimensions into a quantitative approach. An example is provided in Montezeri et al (1997). A means of quantifying the similarity of two cases along a dimension is provided, and each dimension is given a weight determined from a training set of past cases, and the new case is ranked against the cases in the database by means of a weighted sum of the similarities of the new case with existing case along each dimension.

With this sort of system the approach starts to have a great deal in common with Information Retrieval. In typical retrieval systems, the case is not conceptualised as being analysed into features or dimensions, but rather as the text of the decision on the case. Using any of a variety of Information Retrieval techniques it is possible to rank existing cases in order of relevance to a new case. Such systems have, however, been used mainly to present cases to a reader who will make the decision, rather than as a component in an automated reasoning system. A good example is provided by Flexlaw (Smith et al 1995).

2.3 Statistical Approaches

The third group of approaches again take cases as their starting point, but instead of matching cases attempts to learn some kind of formula from the past cases which can then be applied to new cases.

A variety of statistical techniques can be used: examples include least squares discriminant analysis (in FINDER (Tyree 1989), and multiple linear regression (Groendijk and Tragter 1996). Perhaps the most popular variant recently, however, has been the use of Artificial Neural Networks, which embody a statistical technique based on back propagation. Examples of neural nets can be found in Bench-Capon (1993) and in Straneri and Zeleznikow (1995).

In all of the statistical approaches the domain is again conceptualised as a set of cases, the features of which form the inputs to the statistical analysis. These features may be presented to the technique as a flat structure, as in Bench-Capon (1993). Alternatively the features may be grouped into issues, as in Straneri and Zeleznikow (1995). This grouping into issues has obvious correspondences with the analysis of features into dimensions in some of the case based approaches. It might be argued that this grouping is undesirable since it imposes preconceptions on the features and the appropriate grouping might rather be expected to be "discovered" from the net. However, the grouping might be desirable if it reflects a cogent analysis of the domain, and, where data is limited, it may be necessary to divide features up in order for the net to perform acceptably, since too many inputs relative to cases will be likely to give strange results.

One point to note with statistical approaches is that all the inputs have to be numeric. This makes the approach more natural for some domains than for others. Boolean features can be converted to 1 and 0 straightforwardly, but enumerated ranges give more problems. Here the choice is between imposing an ordering on the possible values, or proliferating inputs by assigning a different Boolean input to each of the possible values.

3. Ontologies of the Above Approaches

If we want to take an ontological perspective on the above work, we need to rise above the details, and consider how the domain is being conceptualised. We will begin with a detailed look at the conceptualisation underlying the rule based approaches, and then attempt to relate the conceptualisations of the other approaches to it.

3.1 Conceptualisation of the Rule Based Approaches

Common to all the rule based approaches is the idea that some set of facts is given to the system as input, and that these facts are used to trigger rules which permit the deduction of other concepts described in terms of these facts. The user is expected to provide these facts: of course, there is the possibility that the user may have to resolve some open texture in order to provide the answers.

The nature of the rules, however, differs in the approaches, depending on whether the rules are supposed to represent (by “rule” here we understand not a single clause, but the set of clauses relating to a particular predicate):

1) definitions: this is the pure logic programming style, in which failure of positive conditions for a predicate is treated as sufficient for the truth of the negation of the predicate. Such definitions will, therefore always be complete.

2) partial definitions: in this conception negation is *not* treated as failure. Thus while a rule may establish the truth of a predicate, the negation must be established by explicit rules. In this approach. moreover, the

conditions for the truth and falsity of a predicate are supposed not to be co-tenable: thus conflict is excluded, and the definition is partial, in that in some cases it may not produce an answer.

3) reasons: here conflict is permitted, and so rules licensing both a predicate and its negation may be applicable. Here conclusions may be defeated by other rules.

In the last case the conceptualisation must be extended to include some way of resolving conflicts between rules.

1) General principles: here rules are evaluated against one another according to their properties, such as their specificity or provenance.

2) Specific information: here rules are evaluated according to a spefic relation that holds between them, such as the outweighs of Hage: this relation may be expressed in absolute or in quantative tetrms.

3) User decision: here the users are simply presented with the conflict and invited to choose the line of argument they prefer.

An additional consideration relevant to the conception, but which may be overlooked if we concentrate only on the treatment of open texture, is how intermediate predicates are determined. In all approaches there will be predicates which group together the inputs into intermediate rules which are used to determine the outputs. This grouping can follow one of several principles:

1) The intermediate predicates represent issues; such as is the employment suitable;

2) The intermediate predicates reflect the structure of the source; such as a particylar section of an Act;

3) The intermediate predicates are chosen on software engineering grounds; for example to limit the number of clauses in a body of a rule.

All of these can, on occasion, lead to the same decisions, but often there can be differences; an issue may bring together material from several parts of a source, and a section may result in too long a rule.

3.2 Conceptualisation of the Case Based Approaches

In the case based approaches we again have a number of inputs to the system, but these are grouped together as a case. Inference is not by deduction but by matching. At first sight they look rather different from the rule based approaches.

If, however, we look deeper we can see some similarities. A past case provides a reason for its outcome to be applied to the new case. Since a past case is unlikely to apply in its entirety to a new case, we introduce a looser notion of match than satisfaction of *all* the conditions of some rule. If we equate the matching of cases with the firing of rules in the rule based approaches, we see that we have the same possibilities as above; we can choose a single precedent, or we can attempt to weigh precedents against each other. We can resolve conflicts by applying meta-principles such as “prefer the most recent case”, or we can attempt to provide a means of determining a specific *outweighs* relation between pairs of cases, or we can leave it to the user to decide. We can use qualitative or quantitative measures.

If we admit dimensions into our conceptualisation, we go even closer to “reasons”. Now the structure of features into cases is only for convenience: the real work is done by subsets of the features of the case, and the case as a whole plays no role in the matching. In a sense these dimensions play a similar role in the structuring of the inputs as is played by the intermediate predicates of the rule based approaches.

3.3 Conceptualisation of the Statistical Approaches

Here too we have a set of inputs relating to a case, and an output. The conceptualisation here, however, requires the output to be a function, in the mathematical sense, of the inputs. Some of the degrees of freedom present above are therefore excluded in this approach: the approach is necessarily quantitative, and there will be a single decision, albeit possibly qualified with a number representing confidence in the classification. All conflict and weighing and accumulation of reasons is subsumed into the formula.

There does, however, remain the possibility of admitting structure into the conceptualisation: it is possible to decompose the overall problem into sub-problems, each with its own associated function. As in the above approaches this decomposition may be on the basis of legal considerations (“issues”), or purely software engineering considerations, to get more maintainable functions, perhaps.

4. Open Texture in Ontologies

In this section we turn to work on ontologies proper. This work has the advantage of being later than most of the systems described above. However, it is interesting to see how ontologies derived from first principles may relate to those recovered from systems designed without explicit consideration of ontologies.

So far two reasonably well developed ontologies for legal systems have been produced: the functional ontology of Valente (1995) and the frame based ontology of van Kralingen and Visser (van Kralingen (1995) and Visser and Bench-Capon (1996)). For a fuller description and a more general comparison of these ontologies, see Bench-Capon and Visser (1997).
Valente's Functional Ontology

4.1 Valente’s Functional Ontology

For the purposes of open texture, the important class of knowledge in Valente’s ontology is *world knowledge*, since it is this that is intended to allow us to move from concepts in the real world to concepts defined in law. It is precisely in this transition that the open texture problem arises. Valente distinguishes two types of open texture:

1) *incomplete definitions*, where some, but not all, sufficient conditions are given for the application and non-application of the concept, and
2) *primitive concepts*, which are not defined and the application of which is thus left to whoever is applying the concepts.

This is a distinction we met in all the approaches above. All rely on information solicited from the user, and if the question which solicits the information is open textured this will introduce open texture into the system. The incomplete definitions seem to lean towards the rule based view, but might equally well accommodate dimensions.

Valente concedes that he offers no solution to the open texture problem, but claims that his proposed formalism can help in the representation of such concepts by being rich enough to permit incomplete definitions. Such definitions may conflict, in which case it is the purpose of the *meta-legal knowledge* to resolve them. Valente’s chief example of meta-legal knowledge is *lex specialis,* which suggests that his ontology falls into the category of qualitative resolution using general principles. No support is offered for the resolution of primitive concepts; here the user must decide.

4.2 The Frame Based Ontology of van Kralingen and Visser

From the point of view of open texture, the interesting class of entities are the *concept descriptions*. As can be seen from the above, within concept descriptions they distinguish *definitions*, *deeming provisions* and *factors*. Any of these can be used to describe an open textured concept. Definitions and deeming provisions are not required either to be complete, or to be conflict free, and so any of the rule based styles of 2.1 can be accommodated here. Factors are used to give a description for concepts for which a statistical approach is appropriate: in the application developed from this ontology which was in the domain of Dutch Unemployment Benefit Law, only one such concept was modelled, that of "suitable employment". This concept has a long standing body of case law associated with it which has to a very great extent established its boundaries. A statistical analysis by de Wildt (1993), is derived from this case law, and results in just such a weighted formula, which can be expressed in their ontology.

This aspect of the ontology of van Kralingen and Visser draws on the work of Quast and others at Leiden and has been reported in more detail in Quast et al (1996), which describes a well elaborated model for open textured concepts, which they term the "qualification model". In this model they include both decisive and non-decisive circumstances. We can see decisive circumstances as corresponding to definitions, and non-decisive circumstances to reasons or factors. In the case of factors, circumstances must be given weights, and a threshold which the weighted sum must exceed if the concept is to be attributed, as in the particular case of Dutch Unemployment Benefit Law, and the formula of de Wildt. The model has been used, with success, to analyse five different domains in Dutch law.

This ontology makes no prescription with regard to the representation of open texture: both rule and statistically based approaches can be accommodated within it. Case based approaches sit less comfortably, since there is no explicit notion of precedent within the ontology. Dimensions could be represented as factors, or even as definitions, but to do so involves extracting the *ratio* of a case. The ontology confines itself to providing a variety of forms of concept description. The actual resolution of open texture occurs in the inference mechanisms defined to act on the descriptions. These inference mechanisms are defined not on the *ontology,* but on the representations which realise it in a particular application.

4.3 Key Points From Existing Ontologies

From this examination of reported ontologies we learn that :

1) Two sources of open texture can be identified: loosely defined concepts and the primitive terms used to define concepts.

2) Legal ontologies must accommodate open texture: while some domains may be able to build ontologies based on complete definitions, this will not be possible in the legal domain.

3) Open texture can be expressed in different ways: as incomplete definitions, or as a separate type of concept, where different definitional mechanisms, perhaps reflecting statistical analyses, must be invoked

A distinction can also be drawn between the general approach of Valente in which whilst the definition is incomplete, sufficient conditions are given, which, if they apply, should determine whether or not the concept is applicable. Such sufficient conditions are considered in isolation. Should they conflict appeal must be made to meta-knowledge to resolve the conflict. In contrast other approaches allow for such factors to be considered together, implicitly balancing the strength of pro and contra features within the qualification model.

5. Concluding Remarks

The above discussion shows up some important issues that arise when we attempt to produce an ontology for the legal domain. The issues concern the conceptualisation of the domain, and are orthogonal to the particular representation employed. Since debates about approaches to open texture have tended to centre on rule based *versus* case based *versus* statistical, it is valuable to highlight these more fundamental issues.

1) All three approaches described in section 2 use a notion of cases being described by a set of features which provide the primitive inputs to the system. Thereafter, however, a choice must be made: the case can be viewed as a flat collection of features, or a further level of structure can be provided, grouping features into issues. Examples of both can be found in all three styles of representation. Thus one important choice relates to whether the domain conceptualisation admits issues or not. A system built on a conceptualisation which does admit issues is likely to be a more faithful reflection of the way lawyers conceive of cases, but it will entail a more extensive and sophisticated analysis.
2) A second issue relates to the attitude towards conflicts and their resolution. In some systems these cannot occur, or are resolve implicitly, in others they are resolved explicitly through the use of meta-principles, and in others, like reason based logic and HYPO, they are embraced and exploited, and their resolution is explicitly part of the system behaviour. With regard to open texture, the importance of this choice is that without conflict, the system will be silent about cases in the penumbra of doubt; once conflict is permitted the mechanism for its resolution is what is supposed to reflect reasoning about open texture.
3) The third issue is how answers are ranked. In some approaches only one answer is returned, but in others there can either be a qualitative ranking , or a quantitative ranking giving a total order. In the last case we will commit ourselves to providing weights for different features, and to providing a method by which to determine the appropriate weights. While statistical approaches tend to be quantitative for obvious reasons, examples of each of the other three choices can be found within both rule and case based representations.

Once we have identified these issues which cut across the representation formalism used to implement the system, we see that we must make important ontological choices before we even consider the question of which formalism we should use. We must decide whether:

1) we want to analyse the domain thoroughly to identify and represent issues, or whether we want to be less committal, and either impose no structure, or allow other considerations, such as software engineering considerations, to determine the structure of our representation.;
2) whether we can confine the system to the core of certainty and simply provide no answer in hard cases, or whether we want to deal also with hard cases;
3) if we wish to deal with hard cases, the mechanism we are going to provide for the resolution of conflicting reasons;
4) whether we want a qualitative or a quantitative measure of the strength of reasons.

Considering these ontological issues, and exploring which answers are most appropriate in different situations, will, we believe, give us a much improved understanding of how to address the problem of open texture in legal information systems. Moreover looked at in this way we can open up interesting possibilities for ideas generated within one representation formalism to be applied to systems which use a different formalism.

References

.

Ashley, K.D., (1990). *Modeling legal argument: Reasoning with cases and hypotheticals.* MIT Press, Cambridge Mass.

Bench-Capon, T.J.M., (1993). *Neural Networks and Open Texture,* in Proceedings of the Fourth International Conference on AI and Law, ACM Press: New York.

Gardner, A., (1987). *An Artificial Intelligence Approach to legal Reasoning*, MIT Press, Cambridge, Mass.

Groendijk, C., and Tragter, M., (1995) *Statistical and neural net approaches to smart-money determination,* in Hage, J., et al (eds) Proceedings of JURIX 95, Koninklijke Vermande.

Gruber, T.R. (1992). *Ontolingua: A Mechanism to Support Portable Ontologies*, Knowledge Systems Laboratory, Stanford University, Stanford, United States.

Gruber, T.R., (1995).*Towards Principles for the Design of Ontologies Used for Knowledge Sharing*. International Journal of Human-Computer Interaction, Vol 43, pp 907-928.

Hage, J., (1996).*A Theory of Legal Reasoning and A Logic To Match.* Artificial Intelligence and Law, Vol 4 Nos 3-4.

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

Kowalski, A., (1991). *Case-based Reasoning and the Deep Structure Approach to Knowledge Representation,* in Proceedings of the Third International Conference on AI and Law, ACM Press: New York.

Kralingen, R. van, (1996). *Frame Based Conceptual Models of Law*, Kluwer, Deventer.

Montezeri, Mohammed Ali, Bench-Capon, T.J.M., and Adam, A.M, (1997). *LASER: A system to retrieve UK Employment Law Cases*, Information Technology and Communications Law, in press.

Poulin, D., Bratley, P., Fremont, J., and Mackaay, E., (1993) *Legal Interpretation in Expert Systems,* in Proceedings of the Third International Conference on AI and Law, ACM Press: New York.

Prakken, H., (1991). *A Tool in Modelling Disagreement in Law: Preferring the most specific argument*, in Proceedings of the Third International Conference on AI and Law, ACM Press: New York.

Schild, U.J., (1989). *Open-textured law, expert systems and logic programming*. PhD thesis. Department of Computing, Imperial College, London, 1989.

Schild, U.J., (1993). *The Use of Meta-Rules in Rule Based Legal Computer Systems* in Proceedings of the Fourth International Conference on AI and Law, ACM Press: New York.

Sergot, M.J., Sadri, F., Kowalski, R.A., Kriwaczek, F., Hammond, P., Cory, H.T., (1986). *The British Nationality Act as a logic program.* Communications of the ACM 29, 5 (May 1986), pp 370-386.

Smith, J.C., Gelbart, D., MacCrimmin, K., Atherton, B., Shinehoft, M., and Quintana, L., (1995). *Artificial Intelligence and Legal Discourse*, Artificial Intelligence and Law, Vol 3 Nos 1-2, pp 55-95.

Straneri, A., and Zeleznikow, J., (1995) *The Split-Up system: Integrating neural networks and rule- based reasoning in the legal domain*, in Proceedings of the Fifth International Conference on AI and Law, ACM Press: New York.

Susskind, R.A., (1987), *Expert Systems in Law*, Clarendon Press, Oxford.

Tyree, A.L., (1989), *Expert Systems in Law*, Prentice Hall, Sydney.

Valente, A., (1995). *Legal Knowledge Engineering: A Modelling Approach*. IOS Press, Amsterdam.

Vermesan, A.I., and Bench-Capon, T.J.M., (1995). *Techniques for the Verification and Validation of Knowledge Based Systems: A Survey Based on the Symbol/Knowledge Level Distinction*, Software Testing, Verification and Reliability, vol 5 no 4, 1995, pp 233-72.

Visser, P.R.S., (1995), *Knowledge Specification for Multiple Legal Tasks*, Kluwer.

Visser, P.R.S., and Bench-Capon, T.J.M., (1996). *The Formal Specification of a Legal Ontology*, in van Kralingen et al (eds) Proceedings of JURIX 96, Tilburg University Press, Tilburg.

Visser, P.R.S., and Bench-Capon, T.J.M., (1997), *A comparison of four ontologies for the design of legal information systems*, Artificial Intelligence and Law, in press.

Wildt, J.H. de, (1993). *Rechters en vage normen*. Gouda Quint, Arnhem.