

A Contradiction Management System

Paul Bratley*, Pierre St-Vincent* and Daniel Poulin**

*Département d'informatique et de r.o.

**Centre de recherche en droit public

Université de Montréal

C.P. 6128, Succursale A

Montréal (Québec) Canada H3C 3J7

Abstract

In most existing systems that give expert advice, efforts are made to ensure that the rules in the knowledge base do not give rise to contradictions. This is partly to avoid confusing the user, and partly for fear that a logical system that generates even a single contradiction must collapse. We argue that not only can contradictions be tolerated, but they can even be useful. In a legal context, it is in fact essential to allow contradictions and to handle them correctly. A legal expert system that can see only one side of an argument would not be useful in practice; an advocate needs to know the strengths and weaknesses of the opposing argument, too. Thus a legal expert system must allow for different interpretations of the underlying legal texts. With this in view, we propose a two level architecture, where the object level may include contradictory rules, and the metalevel resolves these conflicts. Furthermore, the 'closed world assumption' gives different results depending on which side one is arguing. Once such an architecture is in place, there are good reasons for allowing contradictions in wider contexts, too. In everyday reasoning, this is a common mechanism for testing an argument.

1 Introduction

In most existing rule-based expert systems, contradictions are anathema. Reasons for this are not far to seek. On the one hand, users consult expert systems to obtain reliable advice, not to be confused more than they are already by conflicting conclusions and recommendations. On the other, designers of such systems are no doubt influenced, consciously or unconsciously, by exposure to classical logic, where any contradiction in a logical theory causes the whole deductive edifice to collapse. The classical situation is summed up by the Latin tag *ex falso quodlibet*: from a contradiction, one can derive any conclusion whatsoever. For a system that is supposed to give expert advice, this is clearly undesirable.

The influence of this attitude is discernible in other areas of artificial intelligence, too. Such names as belief-revision systems, or truth-maintenance systems, strongly suggest that the domain of interest to be represented on the computer has only one tenable model, and that it is the job of the system to find and use only this unique 'true' model of the world. If a new fact contradicts a conclusion already established, then something must be wrong, and either the new fact or one of the old premisses must be rejected or modified. The whole development of nonmonotonic logics may not unreasonably be attributed to the desire to absorb apparently contradictory facts into a system without thereby generating contradictory conclusions.

In this paper we argue that such an attitude is untenable in the context of legal expert systems, where contradictions are omnipresent, and any attempt to suppress them grossly distorts the legal reasoning process. No useful legal expert system can argue for just one side of a case, so designers of such systems are obliged to cope with contradictions whether they like it or not. Contradictions and ambiguities arise in several ways. In fields where the law is determined by jurisprudence rather than by statute, a number of researchers have described systems able to argue for both sides of a question, usually employing case-based reasoning [Rissland 87; Ashley 90]. However little work of this kind has been done in the area of statute-based law except some recent proposals from Rissland and Skalak to use these techniques to augment rule-based systems in statutory law [Rissland 89; Skalak 92]. But nothing has been done to

address the part of the problem caused by the fact that legal texts are subject to interpretation, and their apparent meaning may change depending among other things on the user's point of view. We believe that this can be handled using metaprogramming; in this paper we propose a two level architecture where the object level represents the possible meanings of items in the legal texts, and the metalevel implements different interpretations and different points of view. Once this requirement has been faced, other advantages follow from allowing contradictions in rule-based systems.

In the next section we spell out why it is essential to permit contradictions in legal expert systems. Sections 3 and 4 outline how we intend to implement such systems using metaprogramming, and in section 5 we add further reasons which argue in favour of admitting contradictions in a more general context. Section 6 sums up. We believe that this novel perspective has important implications for the design of expert systems in general.

2 The Legal Context

Our team in the Faculty of Law at the University of Montreal is engaged, among other things, in the design of expert systems to give advice to lawyers, clients, and officials in such areas as unemployment insurance, consumer rights, and so on. In such a context, any plausible system design must to some extent mimic the reasoning processes of the lawyers and administrators who apply the law. Legal reasoning, not merely the law itself, has been an object of study for centuries. Indeed there can be few fields where the participants are so acutely concerned not merely with the conclusions of their arguments, but with the methods of argument involved. An expert system which, when presented with a set of facts, replied simply, “Your client is right: go to court,” or “Your client is wrong: take no action,” without being able to explain in some detail how the conclusion was reached, would be even less acceptable in the legal domain than elsewhere.

Furthermore lawyers are famous—notorious even—for their ability to take both (or several) sides of an argument. An advocate's job is to defend his clients' interests, and to this end the advice he requires is not how to attain some abstract notion of justice, but how, within the limits set by professional ethics and accepted grounds of argument, to defend one particular point of view. If he is wise, he will also take advice about how his opponent is likely to defend a contradictory position, so as to be ready to refute this counterargument. It follows that a useful legal expert system must be able to find and explain arguments both for and against any given position: in other words, it must be able to reason in a context where contradictions are the rule rather than the exception.

Such contradictions arise because both facts and laws are subject to interpretation. The factual aspect may be resolved by appeal to precedent:

“In particular, one tries to resolve interpretation problems by considering past applications of the rules and terms in question: by examining precedent cases, comparing and contrasting these with the instant case, and arguing why a previous interpretation can (or cannot) be applied to the new case.” [Rissland 89, page 46]

But in statute-based law, interpretation (in the widest sense) involves also determining both the meaning and the applicability of rules set out in a legal text [Côté 92]. Examples abound of rules that are open to differing interpretations: whether a mobile home in a trailer park is a house or a motor vehicle, whether a couple can be regarded as married in the absence of a formal legal ceremony, and so on. Indeed many notions invoked in the text of, say, a statute may be deliberately left undefined so that the law can be adapted to unforeseen circumstances: the law is said to have ‘open texture.’ Depending on the interpretative framework adopted by the user, different, and often contradictory, meanings may be ascribed to the same legal text.

It is almost universally accepted by jurists that not all the law applicable in a particular area is found in the source texts. Lawyers use not merely the text of the law, but also rules of interpretation: general principles concerning what is and is not legal, common-sense concepts, and so on. A number of attempts have been made to model this interpretative process. Côté, for example [Côté 92], divides the applicable techniques into six classes: besides grammatical and literal, systematic and logical, teleological, and historical approaches, one may appeal to various authorities, or rely on pragmatic considerations. In some cases all, or almost all, these principles may indicate that some particular interpretation is to be preferred; but there are also situations where different principles lead to different results. The lawyer on one side may thus plead the literal meaning of the text, while his opponent relies on the apparent intent of the legislator.

In a legal expert system, situations of the first type, where one interpretation seems indisputably preferable, are relatively easy to handle. Situations of the second type, where contradictions arise, create

difficulties. It is not impossible to imagine metarules of interpretation. MacCormick and Summers, for example, propose that interpretative arguments should be ordered into three classes: first linguistic arguments, second systemic arguments, and third teleological and evaluative arguments [MacCormick 91]. If there is a clear interpretation based on the first class of arguments, it is to be preferred; if not, but if there is a clear interpretation at level two, then it is preferred; and only if there is no clear interpretation based on the first two classes of argument are arguments of type three considered. However no real consensus exists about the applicability of such metarules.

Problems of interpretation can arise at every stage of a legal argument. They may concern questions of fact (Is a skate-board a vehicle?), of the applicability of legal rules (Are John and Jane to be treated as a married couple?), and of their meaning (What does it mean to do something 'immediately' in a given context?). Moreover lawyers do not feel constrained to be consistent in their choice of interpretative techniques. If the first half of an argument can be supported by a literal interpretation of the law, while the second half requires an argument based on the legislators' intent, an advocate will not be embarrassed by this change of tack, but will switch from one interpretative technique to another as required.

We conclude that an adequate legal expert system must be able to handle arguments based on differing, possibly contradictory, interpretations both of the law and of the facts in any particular case. If different arguments lead to different conclusions, then the system must be able to discover at least the most plausible reasoning that could be employed by each side. It must therefore neither eliminate contradictions from its knowledge base, nor collapse into incoherence when a contradiction is encountered.

A second characteristic of legal reasoning will prove important in the sequel. If legal arguments may, generally speaking, be said to be very broadly based, with a wealth of interpretations possible for every rule that is applied, they are on the other hand not usually deep, in the sense that conclusions are reached after the application of a relatively small number of rules. Proving a case in law is not like proving a theorem in mathematics, which may require layer after layer of lemmas, previously proved theorems, and so on. For this reason, we can envisage the use in a legal expert system of complicated inference mechanisms that might prove impossible to apply in a field characterized by much longer chains of reasoning.

However, if the chains of reasoning involved are shorter, on the other hand jurists often proceed in a cumulative fashion. A lawyer is more likely to present several arguments, all tending to the same conclusion, and each quite short, than to construct an elaborate logical edifice depending on a long chain of reasoning. Furthermore, as we have already pointed out, it is almost always possible to produce a counterargument to even the best line of reasoning. This is why in the legal domain, unlike, say, in mathematics, it is always better to have three arguments available than two or just one.

3 Representing Multilayered Knowledge

To implement an expert system capable of handling such considerations, we propose to use a two level architecture, where the inference mechanism is controlled at the metalevel.

At the lower, or object level, we propose that the substantive provisions of the law should be expressed using rules. The rule base is designed to reflect the surface structure of the legislation; that is, in the usual terminology of this field, it is isomorphic to the source text of the law [Sergot 86, B.-Capon 92]. Where we differ from the usual isomorphic approach, however, is that we take into account the one-to-many relationship between the legislative text and the possible and reasonable interpretations of this text. Thus for each provision in the law, there may be several corresponding rules in the knowledge base. This multiplicity of possible rules reflects the open texture [Hart 61] of the law. At the object level, the system works in the usual way. Inferences can be made either by forward or backward chaining, depending

whether the call from the metalevel is trying to deduce all the judicial consequences of a given situation, or to establish some particular conclusion.

Above the isomorphic object level, we propose using a metalevel to take account of the possible, differing interpretations. As early as 1983, Lenat *et al.* [Lenat 83] pointed out that, in many fields, experts use metaknowledge. This is particularly true of lawyers. In particular, several models of interpretation and justification have recourse to metaknowledge [MacCormick 91, Wróblewski 88]. Interpretative models are also closely connected with what McCarty calls ‘deep knowledge’ (see the classic article [McCarty 83]). Metarules are particularly important when problems arise in connection with the ambiguity or the indefinite nature of certain legal dispositions. Hamfelt [Hamfelt 92] uses several levels of metaknowledge to formalize interpretation of the law. In his system, items in the law are represented by rule-schemes, from which metarules derive concrete instances of judicial rules to be applied. In our system, however, all the required rules are already present, and the problem is to control their use.

Our system will use four types of metaknowledge: general, procedural, adversarial, and inferential.

a) General The interpretative models mentioned above can be expressed formally in the metalevel and thus be made available to guide and control the inferences made by a legal expert system. They can be used in a very general way to provide a framework for the inference process and to reduce the search space. In particular, they can be used to control the inference process when the rule base includes contradictory formulations of the judicial constraints. They also serve to weigh the force of different arguments: one may be based on the most likely interpretation, another on a less likely but still plausible model, and so on. Finally the interpretative models implemented in the metalevel are invaluable as an aid to producing explanations and justifications of the legal arguments thus constructed.

b) Procedural Applying a legal text requires more than the ability to produce one or more acceptable interpretations. A legal expert system must also include representations of procedural and methodological knowledge related to particular situations. Knowledge of this type, intimately related to the skills of an expert jurist, reduces the search space, and helps organise the dialogue with the user. However it cannot conveniently be incorporated within the object level rules. In the isomorphic approach to the legal texts, the principle that the structure of the object level rules should correspond to the structure of the legal texts provides no guidance as to how these rules should be used. Even carefully-written laws are not designed like programs, and lawyers who use them invariably use other, supplementary knowledge [Waterman 86]. The jurist knows, for instance, that he must first determine whether the current situation falls into some particular class, whether some particular paragraph of the law applies in this case, and so on.

Several advantages arise when procedural knowledge is separated from the representation of the rules. Aiello [Aiello 88, pp. 246-247] notes that if the way the rule base is to be applied is encoded explicitly at the object level, then this single use is the only one possible. It is preferable to maintain the generality of the rule base. However ‘general’ rules cannot easily be used in practice unless the system incorporates procedural knowledge. Here, such procedural knowledge is represented at the metalevel. A rule base free from control information, structured in the same way as the legislative text, is easier, too, to modify as the law evolves than one where other considerations, such as control of the inference mechanism, complicate the rules.

c) Adversarial Much as for procedural knowledge, we propose to use metaknowledge to represent the different strategies of argument that can be based on the legal rules depending on the point of view of the user. In a legal knowledge base that adopted only one particular interpretation of the law, such ‘points of view’ would be unthinkable, and whatever line of argument the user wished to adopt, the result would be the same. In the system we propose, the presence of alternative rules and interpretations allows us to produce different inferences starting from the same facts. It is at the metalevel that this flexibility is invoked to produce inferences favourable to one or the other of the parties involved.

Consider, for example, the field of law concerning payments to individuals by the government. On the one hand, it is to be supposed that the people administering the law favour careful use of public funds, so they will try to avoid giving money to anybody not entitled to it. On the other, voluntary associations that defend the needy try to maximize the payments to their members. Although both parties found their legal arguments on the same texts, there is frequently a considerable difference between the interpretations of these texts that they advance. The lawyer for one party, even if he is convinced he should win, does not stop his examination of the situation on discovering the first argument that can be used in his favour. On the contrary, he is still interested to know what arguments can be advanced against his position, so he may prepare answers in advance. The rules of inference at the metalevel must implement a similar strategy for using the object level knowledge.

d) Inferential For an experienced jurist, the ambiguity inherent in legal rules poses no serious problem. A legal expert system must also be capable of giving a plain answer based on the most straightforward reading of the law. In the field mentioned previously, concerning government payments to individuals, it is often clear to everybody involved that X is entitled to such-and-such a payment, and that Y is not. Even with a knowledge base incorporating different interpretations of the statutory rules, such plain answers must be possible. Thus the system must know which rules represent on the face of things the most evident readings of the legal texts, so as to give the ‘straightforward’ answer when required. It must also be able to build alternative chains of reasoning when no single, obvious answer can be found. Finally it must be able to produce an argument supporting a predetermined point of view.

On top of all this, for any given conclusion, the system must also be able to indicate how the opposite conclusion could be sustained. More often than not in the legal domain it is insufficient to know that an argument can be constructed showing that X is A; it is also necessary to ensure that no better, contradictory arguments showing that X is not A are available. Alternatively, if such arguments *do* exist, then the advocate had better be aware of them. To achieve this, the system must be provided with strategies which, for a given conclusion and the representation of the rules that support this conclusion, can identify points in the argument where a different course might have been taken, and which can indicate the consequences of following the alternative path. With such a mechanism, the system will be able to produce alternative inferences leading to different conclusions.

In this connection it is interesting to note the analogy between the logician's ‘closed world assumption’ and the lawyer's ‘burden of proof’. In many current systems, it is assumed that an individual mentioned in the knowledge base does not have some particular property (is *not* a student, say, to use one familiar example), unless the contrary is explicitly stated in the assertions or derivable by the rules. Similarly in many legal situations one side has the burden of proof: in the criminal courts, for example, one is *not* guilty unless this can be explicitly proved. In other contexts the burden of proof may shift: if X is out of work and has paid enough unemployment insurance, he is entitled to benefits unless there is proof that he left his job voluntarily; in that case he is not entitled to benefits unless he can prove he had a valid reason; and so on. Thus the inferential metarules in our system must be able to use the closed world assumption on different sides of an argument at different stages.

Finally the system must be capable of explaining and justifying the inferences it has made in terms of the interpretative model used in any particular case. MacCormick and Summers [MacCormick 91] provide one such model, including strategies for resolving conflicts between rules. These strategies can also serve to explain and justify particular inferences.

4 Metareasoning or Truth Maintenance?

Using contradictory knowledge is a challenge to the systems designer. First, it is obviously necessary that different interpretations of the law should be employed in a useful and productive way, not simply

suppressed; this is the whole point of the exercise. If the system includes metarules which suggest that some arguments are better than others, contradictory knowledge must also take these into account. Overall, the system must produce chains of reasoning that correspond to the interpretative model or models it implements, and that can be supported in terms of these models.

From the implementer's point of view, we may sketch such a system as follows. The object level will contain an 'isomorphic' translation of the law, as explained earlier. Thus the knowledge base at this level is inconsistent, and may contain contradictions. The system must be capable of exploring the consequences of a particular interpretation, and then of backtracking to the point where a critical choice was made, so as to follow up a different interpretation. Exploration carried out in this way, in the presence of contradictory rules, is inevitably nonmonotonic.

'Classic' nonmonotonicity arises in an argument whenever a conclusion derived from some set of facts has to be withdrawn when further facts are added [Konolige 88]. Legal reasoning is frequently nonmonotone. For example, we may believe that X is entitled to unemployment benefits because he is out of work after being employed long enough to qualify; if we subsequently learn that he is out of work as a result of industrial action taken against his employer by a union, we may change our opinion; if we later find that X is not a member of the union concerned, we may change our conclusion again, and so on. Thus 'classical' nonmonotonicity of this type must be handled correctly by the system.

The metalevel implements the inference mechanisms corresponding to different ways of interpreting the law. As we see it at the moment, communication between the metalevel and the object level will be based on a 'subtask management' architecture [van Harmelen 89]. In this type of architecture the metalevel delegates subtasks required in the proof to the object level, which includes a standard inference engine. The object level subsequently returns the result obtained from its efforts to find a proof, in our case augmented by the proof tree produced. The metalevel can then, if necessary, delegate a new subtask, and so on.

In this context contradictions will be handled by assigning interpretative preferences to certain rules. Thus one interpretation might prefer a set of rules which together imply conclusion A. A different reading of the law might give preference to a different set of rules, leading to the conclusion $\neg A$. Thus different preferences in the system would correspond to different interpretations.

The conclusion reached by the system according to a particular point of view will be returned, again augmented by the corresponding proof tree. The tree will include not merely the rules involved and their effects, but the reasons why particular rules were chosen, that is, the interpretation chosen for particular prescriptions in the law. Such an augmented tree will allow us to produce explanations firmly tied to the interpretations called into play. Clearly, for an identical set of facts, each metareasoner may produce a different conclusion, corresponding to the particular interpretation or point of view that it implements. The whole system may thus be seen as a 'Contradiction Management System' rather than a 'Truth Maintenance System.'

5 Contradiction in Other Contexts

We have explained why a legal expert system *must* be able to handle contradictory rules and conclusions. Once this problem has been faced, we believe that a system that incorporates contradictions can be useful in a wider context, too.

The principal reason for this is our belief that the force of an argument is most easily measured by comparing it with the best possible defence of the opposite point of view. Numerical measures of certainty or plausibility do not convey the same feeling for the strengths and weaknesses in an argument as does a reasoned attack on the conclusion. To be told "Your starter motor needs changing with

certainty 0.67” is less convincing than to be told “I think your starter motor needs changing and here are the reasons why, but on the other hand I may be wrong, and here are the reasons that support that point of view.” Some such conviction presumably underpins those democratic systems which arrive at governmental decisions by a process of arguing with an official opposition. Whether or not the governing party can be induced to change its mind may be less important than the test of coherence and inevitability of their proposals provided by attempts to contradict them.

“When we risk no contradiction,
It prompts the tongue to deal in fiction.” [Gay 1727]

In many aspects of everyday life, too, when time and cost do not preclude it, the best way to take a complex decision is to take advice from several sources and to compare the different arguments that are offered. One ‘expert’ may have a better reputation or a more competent air than another. Among the conclusions that an expert system is asked to defend, some may rely on rules more universally accepted, or more commonly applied, than others. The system could thus judge the force of its own arguments, and perhaps order them accordingly. However it would not commit itself to one point of view.

Implicit throughout the above discussion, whether in a legal context or more generally, is the requirement that an expert system that can handle contradictions must have a sophisticated way of explaining how it reached any particular decision. What we propose involves a shift of emphasis from *what* the system proposes to *how* such proposals are justified. When the user asks what is the best thing to do, or the best conclusion to reach, he is implicitly asking for the best arguments for and against some line of action or hypothesis. In some cases he will require only one defence and one criticism of a given position, but it is also easy to envisage situations where he would like to see secondary considerations, too. In every case what is needed is not a conclusion, but two or more arguments. The system design must therefore accord considerable importance to the explanations that can be given to the user, not relegate them to a minor role. On the other hand, provided the explanatory module is adequate, there is no longer any need to resort to numerical measures of certainty, probability, or such-like.

6 Conclusion

We have explained why we believe a legal expert system must be able to cope with ambiguous and contradictory rules, and we have sketched an architecture which, we believe, allows this goal to be reached. The knowledge base of such a system includes not only object level, ‘isomorphic’ representations of the underlying texts, but metalevel rules which implement different possible interpretations of the law, as well as different possible viewpoints which the user may wish to adopt. Our claim is that contradiction is not an undesirable property to be eradicated, but a useful ingredient of arguments even in a wider context, where it can serve as a better measure of the strength of an argument than the arbitrary confidence measures typically used at present.

A prototype to test our ideas is currently being implemented at the Université de Montréal. This preliminary version uses an object level and a single metalevel, which is used to implement the different types of metaknowledge. The prototype is written in Prolog, which, while not ideal, is sufficient to allow experiment. For later versions, after we have gained some experience, we believe it will be worthwhile to implement a special language for the metalevel, and possibly to increase the expressiveness of the object level. Sources of inspiration for a specialized language abound: LLD (the Language for Legal Discourse) for the object level [McCarty 89], and the ideas of Bowen [Bowen 85] and Qu-Prolog [Cheng 91] for the meta-level, to cite only a few.

Although much remains to be done, results so far suggest that there is no serious difficulty in principle in implementing a contradiction management system.

7 Acknowledgments

The work described in this paper is supported by a grant from the Social Sciences and Humanities Research Council of Canada and by the Fonds FCAR of the Quebec government.

8 References

- Aiello, L. and Levi, G., "The Uses of Metaknowledge in AI Systems, *Meta-Level Architectures and Reflexion* (eds Maes and Nardi), Amsterdam: Elsevier Science Publishers B.V., 1988, pp. 243-254.
- Ashley, K.D., *Modeling Legal Argument*, Cambridge, Ma: The MIT Press, 1990.
- Bench-Capon, T.J.M and Coenen, F.P., "Isomorphism and Legal Knowledge Based Systems", *AI and Law* 1 (1992), 1, pp. 65-86.
- Bowen, K.A., "Meta-level Programming and Knowledge Representation", *New Generation Computing*, 3 (1985), 4, pp. 359-383.
- Cheng, A.S.K., Robinson, P.J., Staples, J., "Higher Level Meta Programming in Qu-Prolog 3.0", *Proc. 10th Intl Conf. on Logic Programming*, (ed. Furukawa), Cambridge, Ma: MIT Press, 1991
- Côté, P.-A., *The Interpretation of Legislation in Canada*, Second Edition, Cowansville QC: Éditions Yvon Blais, 1992.
- Gay, J., *Fables: The Elephant and the Bookseller*, 1727.
- Hamfelt, A., *Metalogic Representation of Multilayered Knowledge*, Phd Thesis, Uppsala University, Sweden, 1992.
- Hart, H.L.A., *The Concept of Law*, Oxford: Oxford University Press, 1961.
- Konolidge, K., "Reasoning By Introspection", *Meta-level Architectures and Reflexion* (eds Maes and Nardi), Amsterdam: Elsevier Science Publishers B.V., 1988, pp. 61-74.
- Lenat, D., Davis, R., Doyle, J., Genesereth, M., Goldstein, I. and Schrobe, H., "Reasoning about Reasoning", *Building Expert Systems* (eds Hayes-Roth, Waterman and Lenat), Reading, Ma: Addison-Wesley, 1983, pp. 219-239.
- MacCormick, D.N. and Summers, S.R., *Interpreting Statutes—A Comparative Study*, Aldershot: Dartmouth, 1991.
- McCarthy, L.T., "A Language for Legal Discourse: 1 - Basic Features", *Proc. 2nd Intl Conf. on AI and Law*, Vancouver, 1989, New York: ACM Press, pp. 180-189.
- McCarty, L.T., "Intelligent Legal Information Systems: Problems and Prospects", *Rutgers Computer & Technology Law Journal* 9 (1983), pp. 265-294.
- Rissland, E.L. and Ashley, K.D., "A Case-Based System for Trade Secrets Law", *Proc. 1st Intl Conf. on AI and Law*, Boston, 1987, New York: ACM Press, pp. 289-297.
- Rissland, E.L. and Skalak, B.D., "Interpreting Statutory Predicates", *Proc. 2nd Intl Conf. on AI and Law*, Vancouver, 1989, New York: ACM Press, pp. 46-53.

Sergot, M.J., Sadri, F., Kowalski, R.A., Kriwaczek, F., Hammond, P. and Cory, H.T., "The British Nationality Act as a Logic Program", *CACM* 29 (1986), 5, pp. 370-386.

Skalak, B.D. and Rissland, E.L., "Arguments and Cases: An Inevitable Intertwining", *AI and Law* 1 (1992), 1, pp. 3-44.

van Harmelen, F., "A Classification of Meta-level Architectures", *Meta-Programming in Logic Programming* (eds Abramson and Rogers), Cambridge, Ma: MIT Press, 1989, pp.103-122.

Waterman, D.A. and Peterson, M.A. "Expert Systems for Legal Decision Making", *Expert Systems* 3 (1986), 4, pp. 212-225.

Wróblewski, J., "Interprétation," *Dictionnaire encyclopédique de théorie et de sociologie du droit* (ed. Arnaud), Paris: Story-Scientia, pp. 199-201.