Legal knowledge based systems
JURIX 92
Information Technology and Law

The Foundation for Legal Knowledge Systems
Editors:
C.A.F.M. Grütters
J.A.P.J. Breuker

H.J. Van den Herik
A.H.J. Schmidt
C.N.J. De Vey Mestdagh

A. Valente and J.A.P.J. Breuker, A Model-based Approach to Legal Knowledge Engineering, in: C.A.F.M. Grütters, J.A.P.J. Breuker, H.J. Van den Herik, A.H.J. Schmidt, C.N.J. De Vey Mestdagh (eds.), Legal knowledge based systems JURIX 92: Information Technology and Law , The Foundation for Legal Knowledge Systems, Lelystad: Koninklijke Vermande, pp. 123-134, 1994 ISBN 90 5458 031 3.

More information about the JURIX foundation and its activities can be obtained by contacting the JURIX secretariat:

Mr. C.N.J. de Vey Mestdagh
University of Groningen, Faculty of Law
Oude Kijk in 't Jatstraat 26
P.O. Box 716
9700 AS Groningen
Tel: +31 50 3635790/5433
Fax: +31 50 3635603
Email: sesam@rechten.rug.nl

A MODEL-BASED APPROACH TO LEGAL KNOWLEDGE ENGINEERING

A. VALENTE and J.A.P.J. BREUKER
Department of Computer Science and Law, University of Amsterdam, Amsterdam, The Netherlands

Summary

Despite interesting efforts in the last years, major approaches to legal knowledge engineering have failed in bringing up a coherent theory about legal knowledge and legal reasoning. Meanwhile, important results have been obtained in other application areas of AI by the use of a ***model-based*** *approach to knowledge engineering. This article discusses the model-based paradigm, criticizes present approaches to legal knowledge engineering and proposes a* ***model-based approach to legal knowledge engineering*** *as a new direction towards the solution of the main issues in the field.*

1. Introduction

Since the mid-eighties, a shift was made in knowledge engineering towards what was called the *model-based* paradigm. Today, knowledge engineering is viewed as a modeling activity. The development of a knowledge-based system is seen as the construction of a set of models of problem solving behaviour, such that the system is a computational realisation of these models [Wielinga et al., 1992]. Methodologies, architectures and tools for knowledge-based knowledge engineering have been developed. Finally, the modeling paradigm has become a standard in most important application areas (e.g. engineering, medicine). However, legal knowledge engineering has not yet profited from this evolution in knowledge engineering. We claim that the modeling paradigm should be adopted also in legal knowledge enginering, particularly in its emphasis on *modeling* aspects and the use of explicit conceptual models in the knowledge level.

This article is organized as follows. In section 2, we discuss the model-based view on knowledge engineering. In section 3, we use this perspective to discuss the main problems in legal knowledge engineering and propose that the focus should be the development of an ontology of the legal domain. In section 4, we discuss and criticize present approaches to legal knowledge engineering. In section 5, we present the guidelines of a model-based approach to legal knowledge enginering, and discuss some of the work we have been doing in this direction. Finally, in section 6, we present our concluding remarks.

2. The model-based view on knowledge engineering

The model-based view on knowledge engineering is formulated in figure 1.

The model-based view implies two major changes in knowledge engineering. In the first place, the process of developing a knowledge-based system is divided in two phases: the formulation and development of models (i.e. the *modeling phase*) on the one hand, and the design and implementation of computational systems that realise them (i.e. the *realisation phase*) on the other. This separation has several practical implications in the knowledge engineering process, and particularly in knowledge acquisition. The modeling phase is essentially conceptual and dissociated from the design/implementation phase. Consequently, there is much more emphasis on the *analysis* of the problem and the domain, independent of implementational problems.

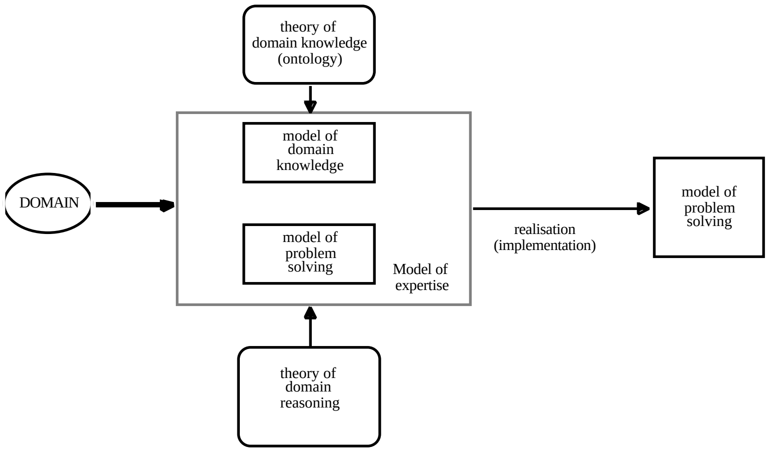

Figure 1: The role of models and theories within an approach to knowledge engineering.

In the second place, the modeling paradigm proposes that problem-solving knowledge and domain knowledge should be represented in two separate models. That is, a *model of domain knowledge* and a *model of problem-solving* form together a model of expertise. The first advantage of this separation is that domain models can be developed independently of specific tasks and problems, and can therefore have a high degree of reusability. The second advantage is that general problem solving models can also be used and adapted to domain-specific tasks, generating relatively domain- independent tasks. This was the essence of e.g. the interpretation models of KADS-I [Breuker et al., 1987] and Chandrasekaran's generic tasks [Bylander & Chandrasekaran, 1988]. This independence is of course not complete: not all knowledge can be separated from its use. Moreover, it may be methodologically useful or convenient to provide packages of domain and problem-solving knowledge.

The development of domain-specific models should, whenever possible, be supported by *theories*. Following the separation of domain and problem-solving knowledge, there can be two theories: a theory of domain reasoning and a theory of domain knowledge. The first specifies the specific characteristics of reasoning in the domain. This is made by specifying common reasoning modes and structures for typical tasks in the domain. The second explains how the domain is organized by containing an *ontology* of the domain. Domain ontologies are a key aspect. An ontology is a conceptual description of what the universe of discourse in the domain consists of - in other words, what categories of knowledge we should distinguish in order to represent domain knowledge, and how they relate to each other. An ontology must contain, in addition to the definition of the knowledge categories, some kind of "calculus" that makes explicit how these categories are connected and operate. An example of an ontology of space is Euclidean geometry. It recognizes three knowledge categories: points, lines and planes. The axioms of geometry provide the calculus: they define the relations between the elements and allow some inferences to be made. There may be more than one possible ontology for a certain domain: for instance, two different ontologies for qualitative reasoning about physical systems have been developed and are largely used. One of them, by De Kleer and Brown [Kleer & Brown, 1984], sees physical systems as a set of components; another, by Forbus [Forbus, 1984] sees them as processes. Additionally, there may also be more than one calculus for a given set of categories. For instance, Euclid and Lobachevski geometries share the same knowledge categories but provide different calculi by assuming different axioms.

It is argued that the models should be expressed in what Newell called the *knowledge level* [Newell, 1982], in contrast with the *symbol level*. Although each main knowledge engineering research group has its own interpretation and adaptation of Newell's knowledge level principle [Bylander & Chandrasekaran, 1988; McDermott, 1988; Steels, 1990; Wielinga et al., 1992], it is generally agreed that the conceptual, knowledge-oriented aspects should be separated from representation and computational issues in the structuring and development of knowledge-based systems. That is, the adoption of the knowledge-level as the expressive medium for models allows more independence of the pragmatical system-development aspects during the modeling phase, which will reflect on a better analysis and better systems later on.

3. Legal knowledge engineering: what are the problems?

All approaches to legal knowledge engineering have as an implicit and general goal the development of (better) legal knowledge-based systems (KBS). This implies two categories of problems. The first category consists of knowledge engineering and software engineering problems. Issues like system-user cooperation, knowledge acquisition, isomorphism, frames versus rules versus semantic nets, connectionism, forward versus backward chaining, impacts of the KBS in the environment, large scale expert systems and the like are not specific to legal KBS, but apply to the development of KBS in general. These problems correspond to the horizontal arrows in figure 1, comprehending both the modeling and realisation phases.

The second category includes the issues in the representation of legal knowledge, and legal reasoning. Here, we can again separate two sub-categories. On the one hand, one may focus on the solution of legal knowledge representation and legal reasoning problems by the application of AI techniques like default reasoning, sorted logics, non-monotonic reasoning, etc. and specific mechanisms like theorem proving or truth-maintenance systems. On the other hand, one may focus on the law "side" of the problem, trying to understand the nature of legal knowledge and legal reasoning. These two sides correpond to the vertical arrows of figure 1 - that is, the development and use of theories of legal knowledge and legal problem-solving.

We claim that the most important problems are the ones related to the understanding of the nature of legal knowledge and reasoning. McCarty recently argued that the key issue in present AI & Law is knowledge representation [McCarty, 1990]. We partially agree with him: knowledge representation problems are indeed the most important and difficult ones, but we consider that in particular the critical issue is the comprehension of the nature of legal knowledge. We must try to understand in which respects, if any, law and legal problem-solving are different from, for instance, engineering, medicine or mathematics. That is, the main problem we should tackle is the development of an *ontology of the legal domain*. This ontology should then serve as the basis for a theory of legal knowledge and a theory of legal reasoning. It should be noticed that the formulation of these theories and ontology is not of purely academic concern. They are necessary to provide principles that can explain and help improving the results obtained in actual legal KBS applications.

4. Present approaches to legal knowledge engineering

It is possible to distinguish three main approaches in legal knowledge engineering. The first consists on the use of logics (particularly deontic logics) as a tool for representing legal knowledge and legal reasoning, which we will call the (*deontic*) *logic approach*.The second consists on the use of case-based reasoning to solve legal problems, which we will call the *case-based approach*. The third adopts the use of first generation AI technology [Steels, 1990] to develop legal expert systems - we will call this the *pragmatic approach*. In the following sections we will discuss each of these approaches at the light of the modeling perspective discussed in this paper. We should warn that the definition of the main approaches to legal knowledge engineering as described here is not well-delimited enough to make possible an unequivocal classification of each research group, neither is so definite to cover the details of all research that would belong to each approach.

4.1. The (deontic) logic approach

Although the interaction of logics and Law goes back to the study of arguments and argumentation techniques, it is through the use of modal logics (and particularly deontic logics) that a major stream in legal knowledge engineering was formed. Many researchers argue that a (modal) deontic logic is the best representation for legal knowledge, and that legal reasoning is essentially non-monotonic reasoning. Within the large umbrella of the use of logics, however, many different positions can be identified. This happens because there isnot a single way to use logics in AI (and AI & Law) in particular. Some authors (e.g. Moore [Moore, 1984] and Chandrasekaran [Chandrasekaran, 1992]) have discussed the possible roles for logics in AI. Chandrasekaran proposes five possible roles for logics:

1. *Logic as the study of justification patterns* consists of using the rules of logic (either deduction rules or non-monotonic). In this way, logics can be for instance a tool for verifying coherence in arguments.
2. *Logical rules as generators of thought.* It is possible to use the same rules to generate thoughts instead of only justifying them. Logics, here, are seen as the basic mechanism of intelligence, the ultimate inference engine.
3. *Logic as a language for describing the world and tasks.* Logics can be used as a language to describe parts of the world. This has in principle nothing to do with the use of logics as the internal representation used to inference; only to the use of logics as a tool to write formal models and theories.
4. *Logic as a language for semantics of natural language.* Logics can be used as a basis for grounding the semantics of natural language utterances.
5. *Logic as the language for semantics of computer programs.* Logics can also be used to ground the semantics of computer programs and, more specifically, as a tool to computational aspects of AI (e.g.computability and tractability).

Although treated separately, the first and third roles as proposed by Chandrasekaran are generally used together: in writing formal theories, logical deduction is normally used to verify internal coherence. In legal knowledge engineering, they frequently have been used together with the second role - that is, it is also assumed the use of a logical apparatus as an inference engine, mostly a non- monotonic one. The most common case, in Law, is the proposal of a logical formalism (in particular a deontic logic) for both representing legal knowledge and legal problem solving. In practice, this means the formulation of a set of axioms, mostly on top of some well-known logic (e.g. first order). We will focus in this type of use of logics as the most representative of the logic approach. In addition, logics can be used as implementation, symbol-level formalisms, e.g. by using Prolog as an implementation language. This use is *not* our concern in this paper, since, in these situations, logics are not being used as a conceptual tool, but rather as an implementation tool at least in principle as good as any other.

First order logics are too general as a tool to represent knowledge. By using first order logics we get a calculus for free, but this is only part of the problem. As a conceptual knowledge representation language, logics should also provide specific ontological support. That is why specialized logics are proposed to represent specific domains: there are knowledge categories and structures that need a special apparatus to be adequately represented. However, n the formulation of a (deontic) logic it is not always clear that the logic is being used as a formalism to represent *models* of legal problems. In this way, the set of axioms that define the logic constitute (and are the formalization of) a *theory* of the domain: they contain an implicit *ontology* of the domain-specific knowledge. Moreover, the particular inference engines specified by the use of non-monotonic, defeasible or default logics act as a theory of legal reasoning.

In this way, the logic approach uses but hides both models and theories to develop legal knowledge-based systems. In particular, it hides the ontologies and concepts used in the formulation of the axioms, so that a conceptual discussion about these ontologies is almost absent. All comparisons made between the various formalisms are based on the internal coherence of their axiomatization - e.g. verifying if they do not contain any logical paradox. Although alternative axiomatizations are proposed, they are not compared by how well they represent the ontology they assume, and even less by how well they represent the domain. More attention is paid to the quality of the logic as a product than to its adequacy with respect to its application. This is not to say that those issues should not be investigated. What we do mean is that there should be less emphasis on the logical internal problems of the formalisms and more on the connection between the formalisms and the reality they attempt to model. Moreover, because the models are not explicit, the approach does not give any guideline to the modeling process, which limits its practical applicability. They provide almost no practical aid to a knowledge engineer.

Legal problem solving is generally modeled by some logical inference engine (i.e. assuming the second role of logic commented above). In our view, this is a mistake. Although non-monotonic logics or default logics model interesting aspects of reasoning (and that might be useful in legal reasoning), they are far from reflecting the way legal practitioners solve specific real problems. In reality, to describe problem solving it is necessary to use a different scope, and explicitly represent concepts like goal, task, task decomposition, and even control (at the knowledge level). If we structure legal reasoning by defining specific problem solving strategies for each typical task in the domain (and afterwards fro each application task) we are much more close to obtain effective results, and will probably obtain a much larger degree of tractability and computational efficiency.

4.2. The case-based approach

Case-based techniques have widespread use in artificial intelligence [Kolodner & Riesbeck, 1986]. In legal knowledge engineering, the case-based approach is particularly dominant, compared to the logic approach. For instance, in the last International Conference on Artificial Intelligence and Law, twice as many articles are concerned with cased-based techniques than with logics [ACM, 1991].[1]

The case-based approach has foundations in cognitive science works [Riesbeck & Schank, 1989] that tried to model analogical reasoning, or rather to solve actual problems (cases) by retrieving similar ones, and look whether their solution applies - or by straightforward copying the solution. Schank's major examples were in legal reasoning and legal education (e.g. the JUDGE system). Therefore, this approach that seems so natural to law has a strong appeal to legal knowledge engineering. The major assumptions for applying case-based techniques are not made very explicit in most research within this approach. We can nevertheless identify two assumptions: that legal knowledge is organized in cases and that (legal) reasoning with cases is analogical.1

The first assumption is obviously false in a strong sense. Not only in western continental law, but also in common law systems, precedents are not the only legal sources. However, in common law, the use of precedents is fairly dominant, which would justify an important, but not exclusive role for case based reasoning. Indeed case based reasoning techniques are far more popular in the anglo-saxon countries, in particular the USA, than elsewhere.

Also the second assumption, that legal reasoning with cases is analogical, is false. For instance, one can abstract or generalise from a *set* of cases. Assessing the similarity between cases may look strongly like analogical reasoning, but it is hard to distinguish from interpreting a case in generic terms, as it is a component of typical assessment tasks. Generic terms are e.g. a system of laws, in such a way that, from a modeling perspective, there is not necessarily a difference in reasoning with two cases - the actual and the precedent one - or with one case - the actual one - and laws (i.e. generic case descriptions). In fact, the schema for indexing cases acts as a generic model that "describes" cases. This is not to say that case based, analogical reasoning is not part of legal reasoning. It is, but then it is also true that analogical reasoning is not necessarily case based. Finally one should also keep in mind that in constructing artificial problem solvers, one not necessarily has to mimick the way humans solve problems. If people are good in seeing similarities and analogies between cases, there are no good algorithms known to achieve this competence on machines, and most likely our current machines will not be suited at all for these types of performance. This limitation of the role of case based reasoning is by no means superfluous. Riesbeck & Schank [1989, p. 7] claim that "Case based reasoning is in essence how human reasoning works", which is definitely false. However, this claim is not shared by all researchers in the case-based aproach to legal knowledge engineering. Their aims and results are rather practical ones, and most do not intend the approach to cover all legal reasoning. Notable exceptions are for instance Rissland [Skalak & Rissland, 1991] and Ashley [Ashley, 1991].

The problems with these hidden assumptions of the case-based approach indicate that it does not contain a broad domain theory, neither concentrates on modeling aspects. Additionally, these assumptions limit the validity of the approach to legal systems of the Common Law family.

4.3. The pragmatic approach

Many works in legal knowledge engineering have taken a highly pragmatical approach to the development of legal knowledge-based systems. They are based on the use of what we can call first generation AI technology, in contrast to a second generation of expert systems as described by Steels [Steels, 1985]. More than half of the papers presented in the last International Conference on Artificial Intelligence and Law belong to this category [ACM, 1991].

This approach has as major positive aspect the production of practical (even if limited) results in terms of deployed applications. In legal knowledge engineering, as much as in any other application domain for AI, there is a need for a more technological line of research that develops systems and produce practical results. This helps "grounding" the work being done in the more scientific side of the field.

However, the pragmatic approach is not playing this role adequately. First, the view adopted in some works is somewhat narrow, because the AI technology used is not the latest available. Frequently, expert system shells are used as both the implementation tool and the knowledge representation formalism. These shells have many limitations: they use a generic knowledge representation scheme that is very seldom adequate for domain-specific problems. Second, the focus is not on domain specific problems, but rather on knowledge engineering problems which are applicable to any domain. For instance, knowledge acquisition may have interesting aspects which are specific to Law, but knowledge acquisition *per se* or in isolation is not the main problem we need to solve in legal knowledge enginering, but rather a side issue.

The major problem is that the systems developed under these pragmatic goals do not rely on any theory or model. Although interesting results may be obtained, there is no underlying basis to explain them, so that almost no interesting feedback can be made. Most of the time, the use of a rapid prototyping strategy leads to narrow knowledge-based systems that do not reflect properly the peculiarities of the domain and in this way makes a deeper analysis of the problems of the application impossible.

The existence of a more technological, development-oriented side of an AI application domain as legal knowledge enginering is of course very important, but the pragmatic approach should correct its path by, first, trying to use the last developments in terms of AI techniques and, second, relying on explicit and well-formulated theories and models of the legal domain.

5. Model-based legal knowledge engineering

We have proposed that the major problems we have in legal knowledge engineering call for a closer focus on theories and models. We also have discussed the major present approaches to the domain and concluded that they do not provide adequate support for such a focus. In this section, we discuss domain specific aspects of a *model-based approach to legal knowledge engineering*. We have already discussed, in section 2, some principles and views of the modeling paradigm in general. In this section, we discuss a number of principles and guidelines that address law-specific issues and define an approach to legal knowledge engineering. Then, we discuss some issues which are part of our current efforts in this direction.

5.1. Principles and guidelines

The following principles and guidelines specify the main lines of a model-based approach to legal knowledge engineering.

* *The use of models of legal argument and argumentation*
  Legal arguments and argumentation play a significant role in legal reasoning [McCarty & Sridharan, 1982; Gardner, 1987; Skalak & Rissland, 1991]. We use as a principle the representation of legal arguments as a basis for representing legal reasoning. By legal arguments we mean generic *structured argumentation*: arguments in conceptual, structured form, not only (and not necessarily) in natural language. Legal arguments may be used to represent legal knowledge and legal reasoning, independent of their cognitive validity. That is, even if lawyers and legal practitioners in general do not use legal arguments to reason or do not reason around legal arguments, yet such representations are valid as a basis for artificial legal reasoning tools.

* *Generality in legal systems and disciplines*
  It is generally agreed that there are substantial differences between the various families of legal systems and each legal system within these families. Moreover, within some legal systems there are sub- systems that are referred to as different disciplines (e.g. Civil and Penal Law) which also bear strong differences. However, it is our claim that legal systems and legal disciplines have more in common than what is usually assumed, and that we should search for this common ground. We take as a principle that, whenever possible, the theories and models developed within the model-based approach should be general - i.e. valid for all major legal systems. As a result, we should also look for theories that explain these differences in terms of the knowledge categories each system or discipline contains and/or how they are used in specific legal problem solving. Moreover, as a corollary, of the previous principle, we adopt the principle of unified (but not uniform) representation of cases and statutes.

* *Broad ontologies of law*
  Most theoretical efforts in legal knowledge engineering have concentrated on the characteristics and roles of open-textured concepts. Although this is indeed a very important aspect, few works have been dedicated to represent the differences between normal rules of law (norms) and rights, legal positions, etc.

* *The use of theories from legal philosophy*
  In all this enterprise, we should come as close as possible to already existing and long debated theories from legal philosophy. In particular, computational theories of law do not need to be divorced from philosophical ones, but rather borrow from their investigations and avoid reinventing the wheel. In this sense, we support Susskind's claim for "pragmatism within a jurisprudential purist framework" [Susskind, 1991]. Indeed, we believe that the focus on models and modeling provides precisely the bridge between the two positions as he proposed them, and pave the way for a closer collaboration between them.

* *Legal knowledge and commonsense knowledge*
  A great deal of discussion about the relation between legal reasoning and commonsense reasoning has already been done, not only in AI & Law but also in legal philosophy. Within AI, commonsense is a swampy subject, and we are still far from an acute understanding of the phenomenon. Precisely for that reason, we adopt the principle of separating legal knowledge and commonsense knowledge. Some claim that this is impossible, because law is incurably permeated with commonsense knowledge. The key for such a separation, we believe, rely on a broader discussion about ontologies of law.

5.2. Legal ontologies and problem solving tasks

To illustrate the flavour of what is intended with this modelling approach, we will discuss briefly some of our current work on legal ontologies and reasoning.

A first step in developing ontologies of law is a distinction of legal knowledge in terms of *law functions*, and is discussed in [Valente & Breuker, 1991]. For each function, the typical categories differ. For instance, definitional legal knowledge consists of taxonomies of terms. Their role is to act as an interface between common- sense, or world knowledge and, for instance, prescriptive knowledge, in which an agent in a situation is prescribed some action (or to refrain from an action). These categories are further refined. For instance, legal agents perform different roles (actor, party, responsible, etc.). Another example of an ontology for legal domains is McCarty's work on a Language for Legal Discourse (LLD) [McCarty, 1989]. Some of the categories he distinguishes as needed for such a language are time, causation, space, mass, permission, belief, etc. Indeed, in legal domains these categories may play an important role, but the question is whether this constitutes a typical *legal* ontology. Many of these terms refer to common sense notions, and are in no way specific for legal domains. Mass, space and time are rather categories to describe physical than social events. This is also a major problem we encountered in designing our ontologies: we need to decide where a legal view starts, and common sense ends. Common sense reasoning is not only very problematic, but is also difficult to capture in a comprehensive ontology as the CYC project shows [Lenat et al., 1986]. A more adequate solution is in the first place to separate the knowledge contained in legal sources (statutes, precedents) from that of the world these sources refer to [Breuker, 1990]. This is a system-architecture solution, which does not solve the epistemological question. However, this world representation is not some common sense domain, but an abstracted view of how the law, or policymakers, see the working of some subsystem in society. These subsystems are domains of law, and each domain has its particular driving concepts. In tax law it is "earning money", while in civil law it is probably "property". Because they are highly abstract, legal domains may lack common sense categories which the layman still perceives there. For instance, in legal reasoning about traffic, categories like time and mass are absent, and space is reduced to a two dimensional, qualitative one, despite the fact that this domain of law looks at first sight a highly physical, mechanical one [Breuker, 1991. Therefore, one should look for common denominators in these different legal domains as a heuristic to trace the typical legal ontologies. For instance, all legal domains are concerned with the question whether the relationship between an actor and an event is that of "intended causation" (see also [Hart & Honore, 1985]). In short, we believe that a legal ontology can be developed by abstracting from common sense notions, but that for each specific legal domain additional and typical conceptual classes have to be added to this ontology.

A second topic of research is concerned with the reasoning process for legal problem solving tasks. As part of the development of a library of models within the KADS-II project [Aamodt et al., 1992], we are analysing typical legal tasks. It is intended to incorporate and extend the "old" library of KADS interpration models [Breuker et al., 1987]. Typical problem solving tasks such as diagnosis, assessment or design are still among the central entries in the library. As a preliminary result, we have found that there are sub-tasks or sub-problems that are common to most real-life legal tasks. In particular, legal real-life tasks normally contain an *assessment* problem. Assessment is the task in which a case is matched against a set of norms, i.e. knowledge that indicates expected or indicated values for abstract characteristics of the case, to decide whether the case is or not in accordance to the norms (i.e. if all the values are the expected or intended ones). Even in legal tasks where the application of norms is not the major concern, the legal assessment of a situation may still play a significant role. For instance, in order to establish plans or give advice about legal situations, the actual situation and maybe some hypothetical ones might be assessed as cases in order to verify possible outcomes. The results of these assessment sub-tasks can then be used to produce a plan or give the advice needed. Specific models of legal assessment are being elaborated to verify this in detail.

6. Concluding remarks

In many application areas a model-based approach to knowledge engineering has been successfully used. In this article we have discussed a modeling view on knowledge engineering. With this background we have set the main problems to be tackled in legal knowledge engineering, discussed some of the main present approaches to legal knowledge engineering and concluded that they could not yet bring up a coherent theory of the legal domain. Then, we proposed a set of principles and guidelines to a model-based approach to legal knowledge engineering.

A model-based approach to legal knoweldge engineering can provide an alternative to traditional approaches to AI & Law by (1) requiring a coherent domain theory based on explicit ontologies; (2) focusing on the use of knowledge-level models and (3) viewing knowledge engineering as a modeling task.

7. Acknowledgements

A. Valente is supported by the grant number 203182/90.1 of Conselho Nacional de Desenvolvimento Cientifico e Tecnologico (CNPq), Brazil. The research reported here was also partially funded by contract P5248, KADS-II, of the ESPRIT programme of the EC. Partners in this project are: Cap-Sogetti Innovation (F), Touche-Ross (GB), Siemens (D), IBM (F), Vrije Universiteit Brussel (B), SICS (S), CSLogic (S) and the University of Amsterdam. The opinions presented in this article are solely of the authors, and are not necessarily shared by CNPq or members of the consortium.

8. Note

1 Case-based reasoning can be viewed as a technique (e.g. in the form of a shell). However, we discuss it here from a knowledge level standpoint, rather than a symbol level one.

9. References

This article may only be cited as: Valente, A. and J.A.P.J. Breuker, A Model-Based Approach to Legal Knowledge Engineering. In: Grütters, C.A.F.M., J.A.P.J. Breuker, H.J. van den Herik, A.H.J. Schmidt and C.N.J. de Vey Mestdagh (eds), *Legal Knowledge Based Systems: Information Technology & Law, JURIX'92*, Koninklijke Vermande, Lelystad, NL, 1992.

[Aamodt et al., 1992] Aamodt, A., B. Bredeweg, J. Breuker, C. Duursma, C. Lockenhoff, K. Osvarn and J. Top, A. Valente and W. van de Velde, *The commonkads library*, KADS-II Deliverable Task I.3.a KADS-II/TI.3/VUB/RR/005/0.1, ECN, SICS, Siemens, UvA, VUB, April 1992.

[ACM, 1991] ACM, *Proceedings of the Third International Conference on Artificial Intelligence and Law*, Oxford, England, 1991.

[Ashley, 1991] Ashley, K., *Modeling legal argument: reasoning with case and cases and hypotheticals*, MIT Press, Cambridge, 1991.

[Breuker, 1990] Breuker, J., *EUROHELP: Developing Intelligent Help Systems*, EC, Amsterdam, 1990.

[Breuker, 1991] Breuker, J., *Artificial Intelligence and Common Sense in Law*, Inaugural Speech, Amsterdam, 1991.

[Breuker et al., 1987] Breuker, J., B. Wielinga, M. van Someren, R. de Hoog, G. Schreiber, P. de Greef, B. Bredeweg, J. Wielemaker, J.-P. Billault, M. Davoodi and S. Hayward, *Model Driven Knowledge Acquisition: Interpretation Models*, ESPRIT Project P1098 Deliverable D1 (task A1), University of Amsterdam and STL Ltd, 1987.

[Bylander & Chandrasekaran, 1988] Bylander, T. and B. Chandrasekaran, Generic tasks in knowledge-based reasoning: The right level of abstraction for knowledge acquisition. In: Gaines, B. and J. Boose (eds), *Knowledge Acquisition for Knowledge Based Systems*, vol. 1, Academic Press, London, 1988, pp. 65-77.

[Chandrasekaran, 1992] Chandrasekaran, B., *Roles of logic in artificial intelligence*, Technical report, Ohio State University Laboratory for AI Research, 2036 Neil Avenue, Columbus, OH 43210, 1992.

[Forbus, 1984] Forbus, K.D., Qualitative process theory. In: *Artificial Intelligence*, 1984, no. 24, pp. 85-168.

[Gardner, 1987] Lieth Gardner, A. van der, *An Artificial Intelligence Approach to Legal Reasoning*, MIT Press, Cambridge MA, 1987.

[Hart & Honore, 1985] Hart, H.L.A. and T. Honore, *Causation in the Law*, Oxford University Press, New York, second edition, 1985.

[Kleer & Brown, 1984] Kleer, J. de, and J.S. Brown, A qualitative physics based on confluences. In: *Artificial Intelligence*, 1984, no. 24, pp. 7-83.

[Kolodner & Riesbeck, 1986] Kolodner, J. and C. Riesbeck, *Experience, Memory and Reasoning*, Lawrence Erlbaum, Hillsdale, 1986.

[Lenat et al., 1986] Lenat, D., M. Prakash and M. Sheperd, Cyc : Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. In: *AI Magazine*, 1986, pp 65-85.

[McCarty & Sridharan, 1982] McCarty, L.T. and N.S. Sridharan, *A computational theory of legal argument*, Technical Report LRP-TR-13, Laboratory for Computer Science Research, Rutgers University, Brunswick, NJ, 1982.

[McCarty, 1989] McCarty, L.T., A language for legal discourse I. basic structures. In: *Proceedings of the second International Conference on AI and Law*, ACM, Vancouver, 1989.

[McCarty, 1990] McCarty, L.T., AI and Law: How to get there from here. In: *International Conference on Expert Systems in Law*, Bologna, Italy, 1990.

[McDermott, 1988] McDermott, J., Preliminary steps towards a taxonomy of problem-solving methods. In: Marcus, S. (ed), *Automating Knowledge Acquisition for Expert Systems*, Kluwer Academic Publishers, The Netherlands, 1988, pp. 225-255.

[Moore, 1984] Moore, R., The role of logic in artificial intelligence. Note 335. In: *SRI International*, July 1984.

[Newell, 1982] Newell, A., The knowledge level. In: *Artificial Intelligence*, 1982, pp. 87-127.

[Riesbeck & Schank, 1989] Riesbeck, C.K. and R.C. Schank, *Inside Case-Based Reasoning*, Lawrence Erlbaum, Hillsdale, NJ, 1989.

[Skalak & Rissland, 1991] Skalak, D.B. and E.L. Rissland, Argument moves in a rule-guided domain. In: *Proceedings of the Third International Conference on Artificial Intelligence and Law*, Oxford, England, 1991, pp. 1-11.

[Steels, 1985] Steels, L., Second generation expert systems. In: *Future Generation Computer Systems*, vol. 1, no. 4, pp. 213-221.

[Steels, 1990] Steels, L., Components of expertise. In: *AI Magazine*, summer 1990.

[Susskind, 1991] Susskind, R., Pragmatism and purism in AI and legal reasoning. In: Narayanan, A. and M.Bennun (eds), *Law, Computer Science and Artificial Intelligence*, Ablex, Norwood, New Jersey, 1991, pp. 33-44.

[Valente & Breuker, 1991] Valente, A. and J. Breuker, Law functions: Modeling principles in legal reasoning. In: Breuker, J.A.P.J., R.V. de Mulder, J.C. Hage, *Legal Knowledge Based Systems, Model-based legal reasoning, JURIX '91*, 1991, pp. 40-52.

[Wielinga et al., 1992] Wielinga, B.J., A.Th. Schreiber and J.A. Breuker, KADS: A modelling approach to knowledge engineering. In: *Knowledge Acquisition*, Special issue "The KADS approach to knowledge engineering", vol. 4, no. 1, 1992.