Legal knowledge based systems
JURIX 93
Intelligent Tools for Drafting Legislation, Computer - Supported Comparison of Law

The Foundation for Legal Knowledge Systems
Editors:
J.S. Svensson
J.G.J. Wassink
B. van Buggenhout

Jaap Hage, An information network for legislative engineering, in: J.S. Svensson, J.G.J. Wassink, B. van Buggenhout (eds.), Legal knowledge based systems JURIX 95: Intelligent Tools for Drafting Legislation, Computer - Supported Comparison of Law, Lelystad: Koninklijke Vermande, pp. 43-52, 1993 ISBN 90 5458 089 5.

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

AN INFORMATION NETWORK FOR LEGISLATIVE ENGINEERING[1]

Jaap Hage

Summary

This paper identifies three representation-modes of legal knowledge that are relevant for legislative engineering[2]. It introduces a tool that assists in relating these representation-modes to each other. This tool, called an Information Network, is shown to support legislative drafting, maintenance of knowledge bases, version management, documentation of rationales of legislation and linking related information. At the end of the paper some psychological considerations with regard to information networks are presented.

1. Three modes of representation

The law that is to be laid down in a new statute will be represented in at least two forms. Almost everybody is familiar with one form: the 'natural' language formulation in which the sections of statutes are cast. This natural language formulation is preceded by another one. The legislator will have types of cases in mind, to which he intends the new legislation to apply. Moreover, he will also have cases in mind to which the legislation should not apply[3]. This 'mental model'[4] of cases that fall or fall not under the new law, is to be translated into the natural language formulations of the statute. The legislator aims at such a formulation of the statute, that all the cases he intends to cover, and only those, fall under the formulation in which the statute casts the law.

1.1. Extensional representations of the law

Let us call the representation of the law in terms of cases the *extensional representation*. What such an extensional representation amounts to, is most easily demonstrated by means of an example. Sections 3:86 and 3:87 of the Dutch Civil Code deal with the case that somebody obtains some good from a person that does not have the power to dispose over this good. Whether, and under what conditions, the obtainer becomes the new owner of the good depends on a number of circumstances. These circumstances include whether the obtainer acted in good faith, whether he got hold of the good, whether the good is moveable, is money, or is a note to hand or a note of bearer, whether the good was stolen, whether the obtainer paid a reasonable price, etc.
By means of these characteristics, it is possible to identify a number of generic cases, and indicate for each of these cases whether the obtainer has become the new owner. In this manner we obtain a table with on the left hand side an enumeration of all relevant types of cases, and on the right hand side for each type of case an answer to the question whether the obtainer has become the new owner. The table gives no information about how the characteristics on the left hand side contributed to the legal effect on the right hand side; it only enumerates the possible types of cases and their corresponding legal effects. Such a pure enumeration is an extensional representation of the law in this particular domain[5].

1.2. Intensional representations of the law

Only in very simple domains it is possible to translate an extensional representation literally into a natural language representation that describes the table. In our example, such a translation would lead to an extensive list of possible situations that provides little insight into the rationales behind the legal effects. Therefore the legislator prefers to formulate rules that specify which factors contribute to the legal effects and in what manner they do so.

J.S. Svensson, J.G.J. Wassink and B. van Buggenhout (eds.) (1993). Legal Knowledge Based Systems: Jurix '93: Intelligent Tools for Drafting Legislation, Computer-Supported Comparison of Law.

The rules that are formulated reflect the underlying legal principles and indicate how the legislator weighed these possibly conflicting principles. As a consequence, the rules can be read on different levels: they give the main rule, but also exceptions and exceptions to exceptions etc. The final result of the interacting rules should be identical to the results that are extensionally represented in the table. But in this case, the legal information is represented intensionally, that is, in the form of rules that indicate how the factors mentioned in the rule conditions contribute to the effects mentioned in the rule consequents. Natural language representations in the form of legal rules are *intensional* representations of the law.
Ideally, the intensional representation of a legal domain leads to the same results as the extensional representation. If this ideal is achieved, the wordings and the purpose of the regulation coincide. However, it is possible that for some cases the intentional representation leads to other results than the extensional representation. In those cases the wordings of the regulation deviate from the regulation's purpose. It goes without saying that a legislator will try to avoid such a discrepancy between wordings and purpose[6].

1.3. The need for a formal intensional representation
It is not so easy to make an intensional natural language representation of a piece of law that coincides with its intended extensional representation. The main reason for this is that the mental model of the domain will be incomplete. It is easily imaginable that the legislator has a few stereotypical cases in mind that occur relatively frequent, and to which he wants the regulation to apply. The formulated rules cover these stereotypes, but also a number of cases the legislator did not have in mind. Often this will not cause problems, because if the legislator had thought of these cases, the rules would not have been different. Sometimes, however, the legislation in its natural language form covers cases that the legislator, had he thought of them, would not have wanted to be covered. The same counts, of course, for cases that are not covered, but should have been covered.
To avoid these situations, it would be useful to have a tool that spells out the extensional representation of a domain on the basis of an intensional representation. The legislator is then in a position to compare the extensional representation with his mental model and draw conclusions as seems fit. Such tools have already been developed [Nieuwenhuis, 1990, p. 93 f.], but they presuppose a *formal* intensional representation of the domain[7]. There are as yet no tools available that understand natural language representations of rules, and translate them into extensional models of the domain. So, if we want to use tools that automatically generate extensional domain models, we also need a language to describe a legal domain both formally and intensionally. Such a language amounts to a third form in which the domain can be represented. The three resulting forms of representation are:

1. extensional representation;
2. intensional representation in a formal language;
3. intensional representation in natural language.

1.4. The connection between the three forms of representation
During the process of legislative drafting, it should be possible to use the three forms of representation alternatingly and without a fixed order. The legislator should dispose of a formal language in which he can make drafts of a piece of legislation. Rules in this formal language have counterparts in natural language formulations, but also in parts of tables that give extensional representations.
The relations between elements of these different representation forms will not always have a one to one nature. (Cf. fig. 1) One statutory section in natural language form may have several formal rules as its counterparts. Similarly, one formal rule may have effects on several entries in the extensional table. A legislator will want to know what are the extensional effects of a reformulation on natural language level. Therefore, it should be possible to jump easily from one representation mode of the legal domain to another one, in such a manner that the jumping is guided by important relationships which exist between parts of the different representations.

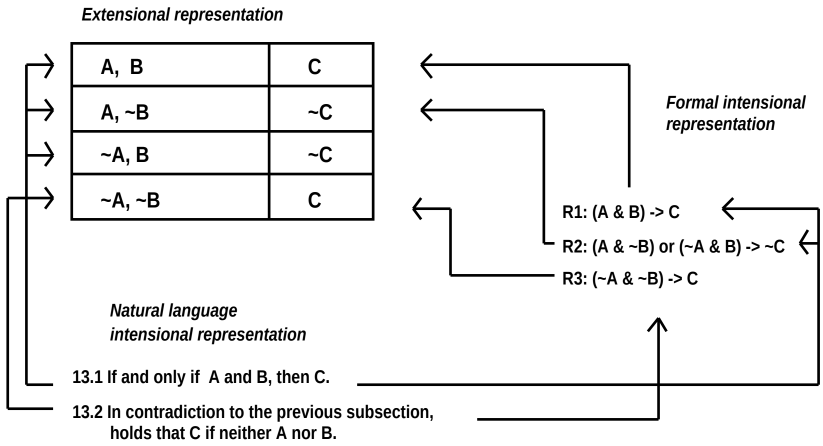

Figure 1: connections between three representation forms

It is not the purpose of this paper to discuss the characteristics of a formal language that is powerful enough to formalize legal rules, or to derive extensional knowledge representations from. Instead I will describe a tool that, amongst others, enables the legislative drafter to keep track of the relations that exist between knowledge elements in the three forms of representation. This tool is the *information network*.

2. Introducing the information network

An information network can best be thought of as a network of links between pieces of information. The pieces of information are contained in independent data- or knowledge bases. By means of the network, related pieces of information are connected to each other, and are made accessible by means of uniform programming and user interfaces.

The connections between independent data and knowledge bases are realized by creating counterparts of information elements. These counterparts do not contain information themselves, but have pointers that indicate where the real information is to be found, and methods to fetch the information and present it to the user. The links between information elements are really links between counterparts.

Let me illustrate these ideas by means of the example of three representational modes. Assume that each mode of representation is held in a database. One database holds plain text; this is the one that contains the natural language representation of the statute. Another database consists of a number of table entries; it holds the extensional representation of the statute. And finally there is a knowledge base that contains rules in a formal language; it contains the formal intensional representation.

All of these data- and knowledge bases retain their original form. They should remain accessible to programs that are suited to their original formats, such as word processors, knowledge base-editors, and database-applications. At the same time, the related elements from each of these information-carriers should be linked to each other. This is done by creating counterparts of each element and connecting these counterparts to each other.

The counterparts in the information network are dealt with in an object-like fashion. This is reflected in their name: they are called *information objects*. The links between information objects are typed, where the link-types indicate the nature of the relation between the linked information objects. For instance, the relation between a natural language rule in the form of a subsection of a statute is linked to a formal rule by a link of type 'is_formalised_by'. The inverse of this link type is 'formalizes'. The relation between intensional representations of the law and an extensional representation can be dealt with by links of the type 'effects' and, inversely, 'is_effected_by'. One row in a table may be effected by more than one rule, formalized or not[8].

Figure 2 illustrates these aspects of an information network for legislative drafting. The little circles and (rounded) rectangles above the horizontal dashed line represent the information objects. They are the counterparts of parts of the data-and knowledge bases depicted at the bottom of the illustration. The rounded rectangles are counterparts of formalized rules, the circles represent natural language rules, and the squares stand for table entries.

Notice that the elements of the data and knowledge bases (let us call them the external information bases) are not connected themselves, but that all connections are made in the network by counterparts of the original information. Moreover, the external information bases need not be internally structured along the lines suggested by the network. It is possible that the information objects, by means of their pointers to parts of the information bases, impose a structure on them that is not otherwise present[9].

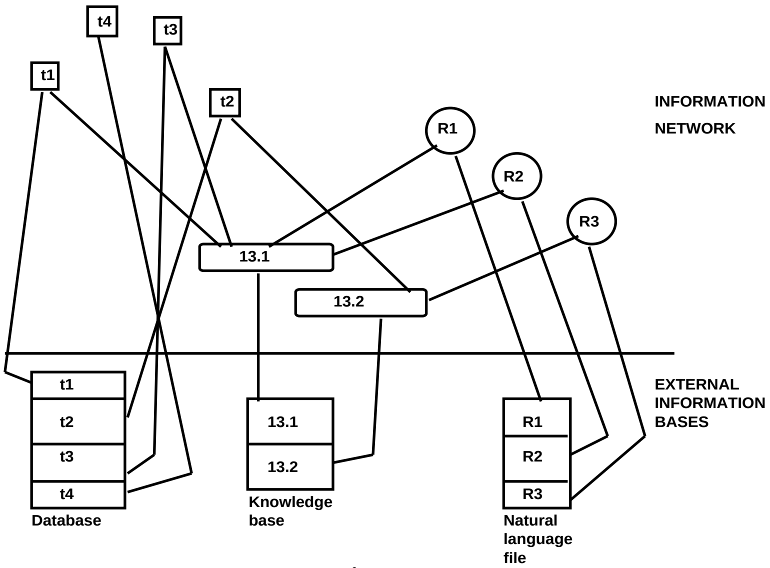

Figure 2

3. Uses of the information network

It is possible to distinguish two aspects of an information network. One aspect is the software that makes it possible to create, modify and delete information objects and links between them. The other part consists of the information contained in the network. (The information in the information bases can be accessed from the network, but is strictly not part of it.) The software part is a kind of 'information network shell'[10]. It is a software library which can be used by applications that make a particular use of the shell.

Depending on the nature of the application, the information network can be used for legislative drafting, maintenance of legal knowledge systems, version management of legislation, and developmental documentation of legislation. I will shortly discuss these possible applications in turn to illustrate the potential uses of information networks for legislative engineering.

3.1. Legislative drafting

The use of an information network for legislative drafting has already been hinted at. The network makes it possible to link connected pieces of information from different representations of the law. The following example shows how an application can use the information network to support legislative drafting through such connections.

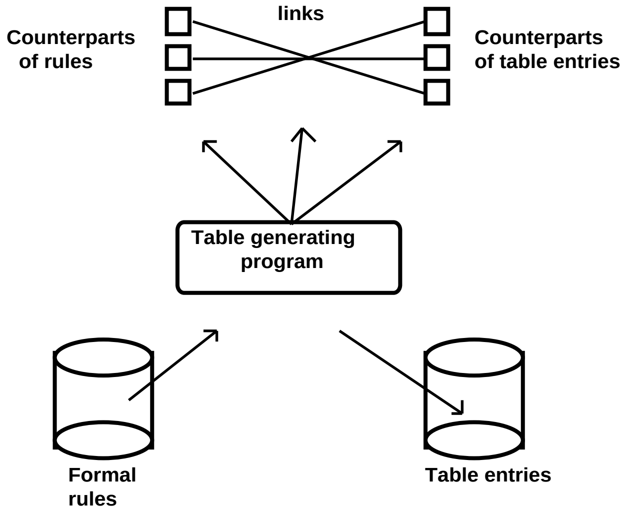

Figure 3

Suppose that we have a formal language which is, on the one hand, powerful enough to represent legal rules, and which, on the other hand, allows the (semi-)automatic generation of an extensional representation of the formalized rules. The program that creates the extensional representation can be adapted to let it generate also counterparts in the information network of both the formal rules that it processes and the table entries it creates. Moreover, the program should create links between the counterparts of the rules and the counterparts of the table entries, to document the rules on which the table entries are based. (Cf. fig. 3)
In this manner the information network is partly filled during the creation of the extensional representation. This part of the information network is useful in case the comparison between the generated table and the mental model of the legislator urges changes in the draft of the legislation. It is easy to follow back the links from the 'wrong' table entries to the rules that caused them. Moreover, if one or more of these rules are changed, the effects in the extensional representation can simply be traced by following the links from the changed rules to the corresponding table entries.
The generated network can also be used as a stepping-stone for the natural language representation of the law. An editor in which the natural language version of the law is created can be made to update the information network by creating counterparts of the natural language rules (sections or subsections) and linking them to the counterparts of the formal rules to which they correspond. (Cf. fig. 1 and 2) As a result of this added information in the information network, it has become possible to trace changes in the logical design of the legislation (the formal rules and/or the table entries) to the natural language version of the draft, and the other way round. Moreover, this connection also forms the basis for the maintenance of a legal knowledge base with regard to this particular domain of the law.

3.2. Maintenance of legal knowledge bases
It has often been noticed that the relatively frequent changes of the law pose a problem for the maintenance of legal knowledge bases [e.g. Weusten, 1993]. One of the measures that are proposed to keep this problem manageable is the so-called 'isomorphic' or 'one-to-one' representation of the law [Karpf, 1989; Nieuwenhuis, 1990, p. 53 f.; Routen and Bench-Capon, 1991; Prakken and Schrickx, 1991]. The idea is that one rule in the knowledge base corresponds to one rule in the natural language version of the law, and that the structure of the legislation (cross-references, exceptions etc.) is reflected in the formal representation. In this manner it would be relatively easy to trace changes in the legislation to necessary changes in the knowledge bases.

It is, however, not very clear what exactly amounts to one rule in a natural language version of the law. The natural division of this version is in books, titles, sections, subsections etc. and none of these units necessarily corresponds to what intuitively is called a rule. Moreover, legal theorists are divided over the question what counts as a (full) rule [Raz, 1980, p. 70 f.]. And even if we assume that an intuitive understanding of what counts as a rule suffices for practical purposes, it has turned out that it is not always possible to use a 'one-to-one' representation of the law [Nieuwenhuis, 1990, p. 56].
The information network provides a way out of the difficulties connected to isomorphic representations of the law. Even if it is not possible to identify a matching unit for each knowledge base rule in the natural language version of the law, the connection between natural language rules and formal rules can be maintained by links in the information network. The network makes it possible to have one to one, one to many, many to one, and many to many relations between arbitrary units on the natural language level and rules in the knowledge base. As a consequence, it is not necessary anymore to force the formal representation of the law into an isomorphic correspondence to natural language units. Adaptation of the formal representation because of changes in the legislation is simplified by links between counterparts of the several representation forms in the information network.

3.3. Version management[11]

Both during the drafting stage of a particular statute and in the development of a piece of legislation over time, there may be different versions of a (proposed) piece of law. These versions will usually have many elements in common, while other elements differ in different versions. An information network makes it possible to keep track of the several versions of a piece of law as it develops in time. The use of *compound information objects* provides an efficient way to do this.
A compound information object is the counterpart of a piece of information that consists of parts that have counterparts of their own. The compound object contains information about the mutual relations between its components. The counterpart of a statute, for instance, contains information about the order of the section-counterparts that are part of it. The components are not actual parts of the compound object, but are linked to it by means of composition-links.

Assume that we have the counterpart of a statute. This counterpart is linked to its composing section-objects. Each of these section-objects has a time-stamp. If a section changes, a new section-counterpart is inserted in to the network, with a new time-stamp. At the same time, a new statute-counterpart is added. This statute-counterpart has composition-links to the unchanged section-counterparts of the old statute and to the new section-counterparts. Moreover, the related changed sections in the two versions are linked too. In this way it is clear how the old and the new version of the statute are related to each other. (Cf. fig. 4.)
It goes without saying that these natural language counterparts can at the same time also be linked to the formal and the extensional representation of the law. Several functions of an information network are realized by different applications which can make use of a common content of the network. The information in the network that is not relevant for an application can remain hidden for that application[12].

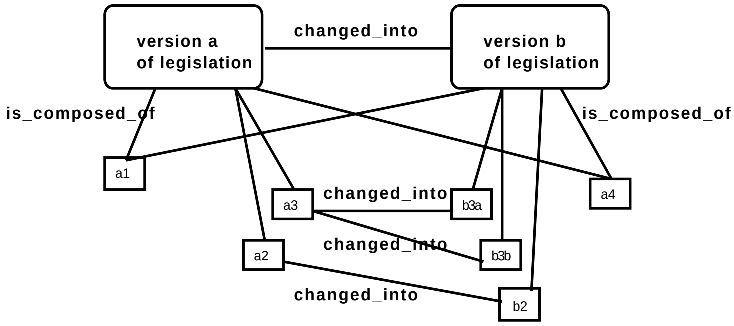

Figure 4

3.4. Developmental documentation

The need to draft a piece of legislation does (at least most of the times) not fall out of the blue. The legislation must serve particular purposes, such as for instance political goals, or perceived defects in present legislation. One view of legislative drafting is that it is a kind of problem solving, where the problem is how the purpose of the legislation is served best. Often there will be different possible solutions to this problem, each with its own advantages and disadvantages. Moreover, each solution can bring new problems with it. For instance, it may be that the legislator decides to allow a person, who obtains a moveable good in good faith from somebody lacking the power to transfer property, to become the new owner. In that case, the resulting problem is how to protect the former owner who lost his good by theft. Such new problems have possible solutions of their own, which in their turn may bring problems with them[13].

Legislative drafting asks for a number of decisions that must be taken and that are not independent of each other. The decision to adopt a particular solution for one problem may cause the need to take a decision for a resulting new problem. Moreover, the solution adopted for one problem may constrain the solutions of other problems. It is, for example, not possible to protect fully both the original owner who lost his good by theft, and the new obtainer who acted in good faith.

As a result, there is a number of problems, possible solutions, and decisions regarding solutions which are mutually related and which together document the rationale behind the development and the design of a piece of legislation. This documentation can be used for the creation of the preparatory documents. It can also be used to support later changes in the legislation, for instance to avoid inconsistencies in legislative purposes.

The mutual relation between problems, solutions and decisions can be represented as a graph in an information network. Such a representation is in keeping with the version management function of the network, since decisions are laid down in legal texts, while alternative solutions are often alternative versions of the legislation and are therefore also laid down in legal texts.

If the development of a piece of legislation is documented in an information network, it becomes possible to add more information to the documentation. If there are scientific publications with regard to the subject of the legislation, they may be attached electronically to those parts of the documentation to which they relate. For instance, if a particular problem with a piece of legislation has been described in the legal literature, references to this literature may be added to the information network. The same counts for comments in the literature on earlier drafts of a statute. If the relevant literature is available in an electronic form, the references can consist of pointers to the relevant parts of the external information bases that contain the concerning publications. In that case, the relevant literature will be available on-line.

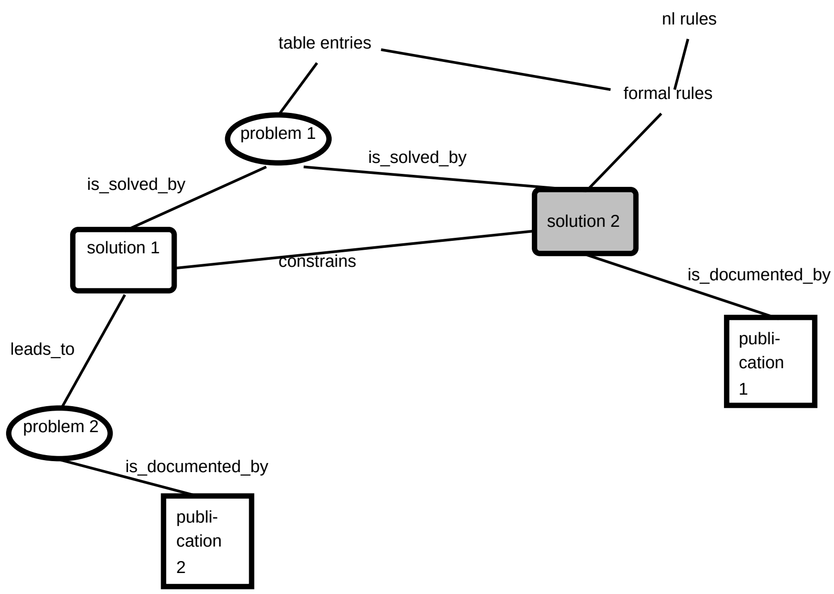

Figure 5

A graph for developmental documentation is pictured in figure 5, where the grayed rectangle is the preferred solution.

4. Concluding psychological considerations

The previous sections illustrated how information networks can support several tasks that belong to the field of legislative engineering. It is no accident that information networks can be used for several legislative tasks. On the contrary, these networks are prone to be useful for a much wider area of applications than legislative support alone. Let me explain why.

The human mind/brain contains an enormous amount of information on all kinds of different subjects and of a nature that would, if it were presented to human senses (where possible), have many different formats. Think for instance, of sound, images, texts, procedural knowledge, motor skills etc. These pieces of information are not neatly stored in different compartments of the brain, nor are they disconnected. Activation of one piece of information can lead to activation of connected other pieces by means of links which are called associative links. These links are essential for the way in which humans process information.

However, most available software tools that provide humans with information are not capable of handling the multifarious interconnections between strongly diverging types of information. Software tends to be orderly in its information handling. But order is alien to the human mind, as appears from the difficulties most humans have with 'clear thinking'[14]. Therefore there will also be a place for software that leaves room for the chaos of human information processing. In the field of information retrieval, such software should enable the user as much as possible to follow associative links in the unpredictable ways the human mind asks for.

Recent developments in the fields of neural nets and hypertext applications reflect, in very different ways, this need for associative information processing[15] However, neural nets are not yet sufficiently powerful to represent the detailed information humans need. Hypertext applications fare a lot better. In a sense, information networks are hypertext applications, because of the links that connect information items of very diverse kinds.

Although there are important similarities between information networks and hypertext applications, there are also at least two connected differences. One difference concerns the amount and the diversity of the links between information objects. Hypertext applications are meant for interactive use by humans. The limited information processing capabilities of humans limit the number of hypertextual links that can usefully be exploited in software systems. Moreover, if the links are of a very diverse nature, this may confuse users. On the other hand, large amounts and great diversity of links, respectively link types are characteristic for the way in which humans link information in their mind. Therefore in information networks, many links of many types should be exploited[16].

The tension that derives from this opposition leads to the second difference between information networks and hypertext applications. information networks are to be used by software applications and not necessarily directly by human users[17]. The applications selectively make use of the many links and link types that are available in the network and avoid in that way that the enormous amount of information leads to confusion. However, since the information network may contain very diverging kinds of information and links, it does not pose severe restrictions to the information processing capabilities of the applications using the network.

Concluding, I dare to predict that information networks, complemented by applications that make effectively use of the 'associative links' between the information objects in the network, have a promising future in the fields of information retrieval and information processing, including the field of legislative engineering.

Acknowledgments

The research for this paper was partly financed by the Foundation for Knowledge Based Systems (SKBS), which seeks to improve the level of expertise in the Netherlands in the field of knowledge based systems, and to promote the transfer of knowledge in this field between universities and business companies.

Notes

1. This paper is partly the result of a conversation with J. Svensson about tools for legislative drafting. Svensson is, however, not responsible for the contents of this paper. The author thanks B. Verheij, G. Span and A. Lodder for valuable comments on a draft of this paper.
2. The term 'legislative engineering' is used as an analogon of 'software engineering'. The idea behind this analogy is that the development and the maintenance of legislation has a lot in common with the development and the maintenance of software. This was pointed out to me by A.H.J. Schmidt in personal conversation. Cf. also Schmidt, 1985.
   Stretching the analogy between the development of legislation and the development of software, we can speak of intelligent CALE-tools, tools for computer assisted legal engineering.
3. Vgl. aanwijzing 7 onder 2 van de Aanwijzigingen voor regelgeving, waar nadrukkelijk wordt verwezen naar de doelstellingen van de te maken regelgeving.
4. The role of mental models in human information processing is extensively discussed by Johnson-Laird, 1983.
5. Cf. Reichenbach, 1947, p. 27 f. on the distinction intensional/extensional.
6. Cf. the account of how the wordings of legislation can be contrary to the purpose of the legislation by Hage, 1992.
7. Moreover, the possibility of such tools presupposes that the formal language satisfies rather severe constraints.
8. Examples of other link types which may be relevant for legal information networks can be found in Verheij, 1993.
9. A more extensive description of INs and their role in legal and medical applications is to be found in Hage *et al.* forthcoming.
10. An information network shell is under development at the University of Limburg in connection with the ARCHIMEDES-project.
11. This section was inspired by the work on the INCA-project. This project is described in Braspenning *et al.*, 1990.
12. Cf. the concluding section of this paper.

13. The description of the legislative process in terms of problems, possible solutions, and decisions regarding the possible solutions, is inspired by a comparable approach to software design and maintenance. This approach is described in Desclaux e.a., 1990.
14. Cf. Minsky, 1985 and the 'pandemonium architecture' as described in Dennett, 1991, with references.
15. On the associationist interpretation of neural nets, cf. Bechtel and Abrahamsen, 1991.
16. Cf. Sowa's discussion about the functioning of the mind, and the consequences he draws from this discussion for knowledge representation by means of conceptual graphs in Sowa, 1984.
17. However, one application might be a 'browser' of the network, that uses filters on object and link types to prevent the user to get lost in 'hyperspace'.

References

[Bechtel and Abrahamsen, 1991] W. Bechtel and A. Abrahamsen. *Connectionism and the Mind.* Basil Blackwell, Oxford.

[Braspenning et al., 1990] P.J. Braspenning, J.W.H.M. Uiterwijk, H. Bakker and L.C.J. van Leeuwen. The Development of an Object-Oriented Representation System for Complex Objects in *Analysis and Design: the INCA-project.* Report CS 90-06, University of Limburg, Department of Computer Science (1990).

[Dennett, 1991] D.C. Dennett, *Consciousness explained*. Penguin Books, London e.a., 1991.

[Desclaux et al., 1990] C. Desclaux, J. Pomian, P. Storr, and A. Thierry, *Design Graph Techniques, MACS internal report* wd/de/krep/100190-1.

[Hage, 1992] J.C. Hage. Redeneren met de strekking van rechtsregels. (Reasoning with the purpose of rules) (H. de Swaan Arons, H. Koppelaar and E.J. H. Kerckhoffs eds.) *Proceedings NAIC '92*, Delft, p. 43-52.

[Hage et al. forthcoming] J.C. Hage, M. van der Meulen, and G.P.J. Span .Intelligent Information Retrieval from Multiple Databases, to be published in *Logica, Informatica e Diritto.*

[Johnson-Laird, 1983] P.N. Johnson-Laird. *Mental models*, Cambridge University Pres, Cambridge e.a., 1983.

[Karpf, 1989] J. Karpf, Quality Assurance of legal Expert Systems. *Pre-proc. of the Third International Conference on Logica, Informatica, Diritto,* (A.A. Martino ed.), Florence, 1989, vol. 1, p. 411-440.

[Minsky, 1985] M. Minsky. *The Society of Mind*, Simon and Schuster, New York, 1985.

[Nieuwenhuis, 1989] M.A. Nieuwenhuis. *TESSEC: een expertsysteem voor de Algemene Bijstandwet*, Kluwer, Deventer, 1989.

[Prakken and Schrickx, 1991] H. Prakken J. Schrickx. Isomorphic models for rules and exceptions in legislation. In: J.A. Breuker, R.V. de Mulder and J.C. Hage (eds.) *Legal Knowledge Based Systems. Model-based reasoning*, Koninklijke Vermande BV, Lelystad, 1991, p. 17-27.

[Raz, 1980] J. Raz. *The Concept of a Legal System*, 2nd ed., Clarendon Press, Oxford, 1980.

[Reichenbach, 1947] H. Reichenbach. *Elements of Symbolic Logic*, The Free Press, New York, 1947.

[Routen and Bench-Capon, 1991] T. Routen, and T. Bench-Capon. Hierarchical formalisations. *International Journal for Man-Machine Studies*, vol. 35, (1991) p. 69-93.

[Schmidt, 1985] A.H.J. Schmidt. Spaghettiwetgeving in wetsontwerp 18 764 (v.v. & v.i), *Delikt en Delinkwent*, 15, (1985), p. 181-188.

[Sowa, 1984] J.F. Sowa. *Conceptual Structures.* Addison Wesley, Reading e.a, 1984.

[Verheij, 1993] H.B. Verheij. *Legal Object and Link Types in the ARCHIMEDES* Information Network. Internal paper of the ARCHIMEDES project, SKBS/B3.A/1993-2.

[Weusten, 1993] Weusten, M.C.M. Validation: the key concept in maintenance of legal KBS, *Proceedings of the Fourth International Conference on Artificial Intelligence and Law*, ACM Press, p. 82-89.