Legal knowledge based systems
JURIX '95
Telecommunication and AI & Law

The Foundation for Legal Knowledge Systems
Editors:
J.C. Hage
T.J.M. Bench-Capon
M.J. Cohen
H.J. van den Herik

Niels Peek, Structure-preserving representations of complex references, in: J.C. Hage, T.J.M. Bench-Capon, M.J. Cohen, H.J. van den Herik (eds.), *Legal knowledge based systems JURIX '95: Telecommunication and AI & Law*, Lelystad: Koninklijke Vermande, 1995, 95-104, ISBN 90 5458 252 9.

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

 http://jurix.bsk.utwente.nl/

STRUCTURE-PRESERVING REPRESENTATIONS OF COMPLEX REFERENCES

Niels Peek*
Department of Law and Computer Science
University of Leiden, P.O. Box 9521, 2300 RA Leiden, the Netherlands
email: jfrinp@ruljur.leidenuniv.nl

Abstract

This paper presents a new formalism for isomorphic representations of legal knowledge, based on *feature structures*. The formalism, which follows the legal-theoretical distinctions of Van Kralingen's (1995) conceptual modeling framework, preserves the structural composition of non-primitive concepts in the representation. We claim this is necessary when the interpretation of a non-primitive concept is partially modified by a deeming provision, because occurrences of the concept in legislation then refer both to the concept as a whole and to its component parts.

1 Introduction

The promulgation of norms is often attended with the creation of a highly-specialized discourse in which many concepts have a technical meaning. Generally, substantial parts of legislation are devoted to *legal definitions* and *deeming provisions* (Bench-Capon *et al.*, 1987), establishing the intended interpretation of relevant legal concepts. Legal definitions create a legal interpretation of the concept described. Deeming provisions, in contrast, modify an existing (either common-sense or legal) interpretation of the concept described (Visser, 1995), by explicitly excluding parts of this interpretation or including other interpretations. So, to determine the interpretation of a concept described by a deeming provision, the original interpretation is needed.

The concepts used in legislation may be described by simple terms, such as 'employer' and 'blind', compound terms, such as 'suitable employment', or even complete phrases, such as 'working hours a calendar week' and 'importing the drugs referred to in Section 2'. Often, parts of such a compound term or phrase also occur independently in the legislation; for instance, the terms 'employment' and 'drugs' denote independent concepts defined elsewhere. We then say that the compound term or phrase denotes a *non-primitive* or *composite* concept; otherwise we speak of a *primitive* concept.

Normally, the interpretation of a composite concept depends solely on its component parts. This situation changes when the legislator modifies the interpretation of a composite concept using a deeming provision. Then, usage of the concept may refer either to the original, compositional interpretation, or to modified interpretation of the concept as a whole. We therefore define the usage of a term or phrase denoting a composite concept that is subject to a deeming provision to be a *complex reference.* As we will see, complex references pose problems for formalizations that do not preserve the structural properties of composite concepts in the representation.

In this paper, we present a *structure-preserving* representation formalism that allows for structured references to composite concepts. Our formalism basically adjusts the *feature structure formalism* (Shieber, 1986; Pollard and Sag, 1987) to represent legal knowledge. We follow the legal-theoretical distinctions of Van Kralingen's (1995) conceptual modeling framework, and strive for representing legal sources in an isomorphic fashion (Bench-Capon and Coenen, 1992).

* This research is sponsored by the Foundation for Law and Government (Reob), which is part of the Netherlands Organization for Scientific Research (NWO).

This paper is organized as follows: in section 2, an example of a complex reference, taken from the *Dutch Opium Act*, is discussed; in section 3, the basic notions of the feature structure formalism are introduced. In section 4, we show how feature structures can be used for the representation of legal knowledge, and in section 5, how they can be used for the representation of complex references. The paper is rounded off with some conclusions and directions for further research in section 6.

2 An example of a complex reference

Consider the following fragment (P1 for Provision 1) of Section 2, Subsection 1 of the Dutch Opium Act (DOA), prohibiting the import of so-called 'harddrugs',[1] formally defined as drugs either listed on Schedule I to the DOA or designated by virtue of the second or third subsection of Section 2 of the DOA:

(P1) **Section 2.**
— 1. It is forbidden:
A. to import into […] the territory of the Netherlands; […]
a. the drugs listed in Schedule I to this Act;
b. the drugs designated by virtue of Subsections 2 and 3 of this Section.

A typical formalization of this provision in predicate logic would yield a rule (here R1) stating that person *p* has violated this norm if each of the separate conditions is fulfilled:

(R1) VIOLATES(*p*,SEC2_SUB1) ←
IMPORT(*p*,*x*,NL) DRUGS(*x*)
(LISTS(SCHEDULE_I,*x*) DESIGNATED(*x*,SEC2_SUB2) DESIGNATED(*x*,SEC2_SUB3))

We notice that in this formalization, it is assumed that the conditions in the antecedent are separate, primitive notions, together making up the applicability of the norm. Unfortunately, the simplicity of the formulation of P1 is misleading, which becomes obvious if we take a look at Section 1, Subsection 4 of the DOA:

(P2) **Section 1.**
— 4. Importing the drugs referred to in Section 2 includes […] any action aimed at further transport […] of the drugs which have been imported […] [2]

This deeming provision extends the interpretation of the concept 'importing the drugs referred to in Section 2', which is exactly what is forbidden by P1. So, P1 applies also in cases not covered by the compositional interpretation of this concept. In order to make a correct formalization of P1 together with P2, we have to adjust our former approach. First of all, we see that 'drugs referred to in Section 2' is treated as a separate notion:

[1] The Dutch Opium Act makes a distinction between "drugs presenting an unacceptable risk to the public health" (harddrugs: e.g., heroin, cocaine, amphetamine) and drugs that do not present this risk (softdrugs: cannabis).

[2] Actually, the provision extends the interpretation of this concept much further, including "importing goods or objects in which the drugs are packed or concealed and any action aimed at further transport, storage, delivery, receipt, or transfer of the drugs which have been imported or of the objects or goods in which the drugs are packed or concealed". We selected this simple case since it contains all generalities possible.

(R2) DRUGS_SEC2(x) ←
DRUGS(x)
(LISTS(SCHEDULE_I,x) DESIGNATED(x,SEC2_SUB2) DESIGNATED(x,SEC2_SUB3))

Second, we need a rule that formalizes the extension of the interpretation of the composite concept 'importing the drugs referred to in Section 2' in accordance with P2. To represent such a composite concept, we typically need a 'hyphenated' predicate (Sergot, 1991, pp. 48-49):

(R3) IMPORT_DRUGS_SEC2(p) ←
TRANSPORT(p,x) IMPORTED(x) DRUGS_SEC2(x)

Third, we need to amend our former formalization (R1) of (P1):

(R1') VIOLATES(p,SEC2_SUB1) ←
IMPORT_DRUGS_SEC2(p)

Fourth, and finally, we have to make explicit the relations between the composite concept 'importing the drugs referred to in Section 2' and its component parts:

(R4) IMPORT_DRUGS_SEC2(p) ←
IMPORT(p,x,NL) DRUGS_SEC2(x)

Although these four rules together make up a correct formalization of P1 and P2, we believe that this formalization has some deficiencies, stemming from the fact that in the legislation, primitive and composite concepts are used intertwined. In the formalization, we have problems using them in this manner. This is best illustrated by R4: because we used a hyphenated predicate to represent the composite concept, we have to add a rule explicating the relations between this concept and its composite parts. However, from a conceptual point of view, the rule tells us nothing at all. Indeed, a similar rule does not occur in the legislation.

The problem is that prohibition P1 refers both to this composite concept as a whole, that is, as defined by deeming provision P2, and to it as a structure, defined by its component parts. So, the antecedent of P1 contains a complex reference. But by using a hyphenated predicate, as in R1', we can only refer to the concept as a whole: the internal structure of the concept is not preserved in the representation. We claim that this problem can be circumvented using a representation that preserves the structural properties of composite concepts. In this paper, we present such a *structure-preserving* representation method, using a different representation formalism. After we have introduced the basic notions of this formalism, we will return to the above example and illustrate our claim.

3 The feature-structure formalism

In this section, we give a short outline of the *feature-structure formalism* (Shieber, 1986; Pollard and Sag, 1987). This formalism was originally designed in the field of computational linguistics to encode linguistic information. However, application of the feature-structure formalism is not limited to any particular domain. In fact, similar structures have been put forward as general mechanisms for knowledge representation (Aït-Kaci, 1985). The feature-structure formalism also resembles *terminological knowledge representation*, a class of formalisms whose origins go back to the knowledge-representation language KL-ONE (Brachman and Schmolze, 1985).

Terminological knowledge representation has recently been suggested as an adequate formalism to model knowledge of legal concepts (Valente, 1995).

Intuitively, a feature structure is just an information-bearing object that describes something by specifying *values* for various *attributes* (*features*) of the described thing[3]. We think of feature structures as providing partial information about the thing described. Various notations exist for feature structures. Here we will use *attribute-value matrices* (AVMs), which are simply lists of features and their corresponding values enclosed in square brackets. For example, matrix M1 represents the feature structure that describes a person named Yossarian who has the rank of captain:

(M1)
$$\begin{bmatrix} \text{NAME} & \text{Yossarian} \\ \text{RANK} & \text{captain} \end{bmatrix}$$

A property of feature structures that is crucial for our purposes is their potential for *hierarchism*, i.e., feature values may themselves be structured:

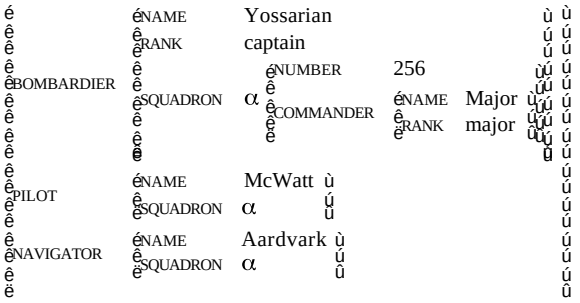
(M2)
$$\begin{bmatrix} \text{NAME} & \text{Yossarian} \\ \text{RANK} & \text{captain} \\ \text{SQUADRON} & \begin{bmatrix} \text{NUMBER} & 256 \\ \text{COMMANDER} & \begin{bmatrix} \text{NAME} & \text{Major} \\ \text{RANK} & \text{major} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Another fundamental aspect of feature structures illustrated by matrices M1 and M2 is that some feature structures are *more informative* than others. In the present case, M2 is more informative than M1, since it contains all the information that M2 contains, and some other information (the SQUADRON specification). We say that M2 *extends* M1, and write M2 $\prec$ M1. Alternatively, we say that M1 *subsumes* M2, denoted by M1 $\succ$ M2. Notice that, in general, if *A* extends *B* (*A* $\prec$ *B*), then any object that could be appropriately described by *A* could also be described by *B*: the less information you have about a thing, the wider the range of possibilities for what that thing might be.

The final characteristic of feature structures is their ability to be *re-entrant* (often also called *structure sharing*). That is, two or more distinct attributes within a feature structure may have a common value specification. We indicate re-entrance in matrices by tagging the multiple occurrences of the common value with a Greek character, as the α in matrix M3. This feature structure describes the crew of a plane, consisting of a bombardier, a pilot, and a navigator. Re-entrance is used to lay down that all three crew members belong to the same squadron. We stress the fact that α is not a variable, but a tag or 'pointer' indicating re-entrance.

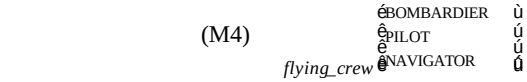
[3] It should be noted that 'thing' may denote any concept, concrete or abstract, material or immaterial.

(M3)



In modeling information, it is often important to distinguish between cases where we *lack* information about the value of a certain attribute, and cases where that attribute is *irrelevant* for the kind of object under description. The feature structure formalism is therefore often augmented with the notion of *type* (Pollard and Sag, 1987). That is, feature structures come in different types, depending on the kind of object they describe, and with each type is associated a set of attributes appropriate for the description of objects of that type. Thus, for example, the appropriate attributes for the feature-structure type *flying_crew* could include BOMBARDIER, PILOT, and NAVIGATOR. When it is necessary to be explicit about the type of a feature structure, we subscript it at the lower left as in matrix M4:

(M4)



Notice that for different attributes, different types of values are appropriate. We may therefore choose to impose type restrictions on attributes of typed feature structures also. For example, we may require that all attributes of feature structures typed *flying_crew* take structured values having attributes NAME, SQUADRON, etc.

The basic equipment of the feature structure formalism outlined above can be extended in various ways, depending on the application. For a detailed discussion of the formalism, its formal properties and various extensions we refer to Shieber (1986) and Rounds and Kasper (1986). In the following sections, we describe how feature structures can be used to represent legal knowledge.

4 Feature structures for legal knowledge representation

In this section, we describe how typed feature structures can be used to model fragments of legislation. We take the conceptual modeling framework developed by Van Kralingen (1995) as a starting point for the typification of the various feature structures that will be used. Therefore, the basic representation structures to be used are descriptions of *norms*, *acts*, and *concepts*.

We strive for representing legal sources in an *isomorphic* fashion (Karpf, 1989), following Prakken and Schrickx (1991) in the description of isomorphism as the situation in which one 'source unit' is formalized in one 'KB-unit'. By a source unit, we mean the smallest identifiable unit of the source from which either a norm, act or concept description can be extracted; by a KB-unit, we mean a feature structure of the corresponding type. It has been argued that isomorphism, yields several benefits from an engineering perspective, including improved validation and maintenance (Bench-Capon and Coenen, 1992).

4.1 Norms

In the description of a norm by a feature structure, M5, we include a reference to the legal SOURCE in which the norm was promulgated, a description of the SUBJECT to whom the norm is addressed, the conditions (COND) which render the norm applicable, the legal modality (MOD), and the OBJECT of the norm, being the act upon which a normative position is taken.

(M5)
$$norm\begin{bmatrix} \text{SOURCE} & source\text{-}reference[\,] \\ \text{SUBJECT} & \\ \text{COND} & \\ \text{MOD} & \\ \text{OBJECT} & act[\,] \end{bmatrix}$$

Source references
A *source reference* makes reference to the promulgation of the norm (or definition, see below). As such, source references can take multiple forms. In this paper, we only discuss one of these forms, namely the reference to some part of a statute; we note, however, that other forms, such as references to case law or legal doctrine could be represented too. The feature-structure type *source-reference* is shown in M6; we include the name of the STATUTE, the relevant SECTION number, and, if necessary, the relevant subsection (SUB), paragraph (PAR) and/or situation (SIT) number.[4] For convenience, we will use a linear format for source references, as in the left-hand part of M6:

(M6)
$$[\text{stat}(c_1),\text{sec}(c_2),\text{sub}(c_3),\text{par}(c_4),\text{sit}(c_5)] \equiv source\text{-}reference\begin{bmatrix} \text{STATUTE} & c_1 \\ \text{SECTION} & c_2 \\ \text{SUB} & c_3 \\ \text{PAR} & c_4 \\ \text{SIT} & c_5 \end{bmatrix}$$

4.2 Acts

We describe *acts* by specifying the name of ACTION itself, and a variable number of *roles*, the first of which is the AGENT performing the act. When necessary, additional circumstances (CIRC) can be specified:

(M7)
$$act\begin{bmatrix} \text{ACTION} & \\ \text{AGENT} (= ROLE_1) & \\ ROLE_2 & \\ \vdots & \\ ROLE_n & \\ \text{CIRC} & \end{bmatrix}$$

The number and type of roles is dependent on the act at hand. In the example mentioned in Section 2, for instance, the action 'import' has three attributes: the AGENT, the OBJECT being imported and the AREA in which this object is imported.

4.3 Concepts

In the definition of *concepts*, we take full advantage of the extension/subsumption relation as described in section 3. We define a concept by extending the feature-structure

[4] The situation number is used to distinguish multiple units contained in a single paragraph.

describing the concept (being in fact the *definiendum* of the definition) with three attributes SCOPE, variable (VAR), and *definiens* (DEF), as shown in M8. The SCOPE, being a source reference, limits the validity of the concept. We refer to the *definiendum* and the SCOPE together as the *head* of the definition. The variable is a tag which can be used in the structured value for *definiens* to refer to the concept as a whole. The value for *definiens* contains an embedded feature structure with attributes SOURCE, TYPE, and conditions (COND). Here, the SOURCE refers to the provision in which the definition was promulgated, the TYPE concerns the type of definition at hand (e.g., 'legal definition' or 'deeming provision'), and the conditions determine the applicability of the concept. The value for the conditions feature will in turn be a complex feature structure, in which the variable mentioned above can be used to refer to the defined concept.

(M8)
$$concept\begin{bmatrix} Attribute_1 & Value_1 \\ \vdots & \vdots \\ Attribute_n & Value_n \\ \text{SCOPE} & source\text{-}reference[\,] \\ \text{VAR} & definiendum \text{ (structural description of the concept under consideration)} \\ \text{DEF} & \begin{bmatrix} \text{SOURCE} & source\text{-}reference[\,] \\ \text{TYPE} & \\ \text{COND} & \end{bmatrix} \end{bmatrix}$$

Since the feature structure describing a concept definition extends the feature structure describing the concept under consideration (the *definiendum*), it matches with occurrences of the concept in norms or other definitions. Any matching occurrence will be called a *reference* to the concept. Furthermore, no formal restrictions apply to the defined concept: any possible feature structure may fill this part of matrix M8. This allows us to describe legal definitions and deeming provisions pertaining to composite concepts, while preserving the structural representation of these concepts. As a result, references to such concepts will be structured too. This is illustrated in the next section.

5 Representing complex references using feature structures

We now return to the legislative example of section 2 of this article. We begin by making a translation of the discussed prohibition of harddrugs import, P1, and then we turn to the translation of the deeming provision concerning the composite concept 'importing the drugs referred to in Section 2 of the DOA', P2. The structure of this composite concept is preserved in the representation, which solves the problem of complex references.

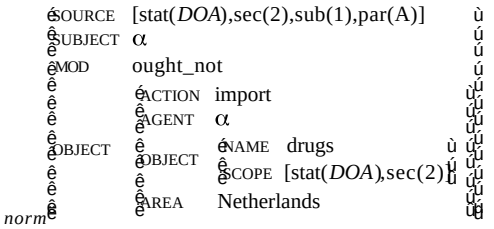
5.1 Isomorphic representation of norms and definitions

We recall that Section 2, Subsection 1 of the DOA prohibits the import of harddrugs. In this provision, we recognize a norm (prohibition), and a description of what count as harddrugs, formally termed 'drugs referred to in Section 2 of the DOA'. The feature structure describing this concept is depicted by matrix M9.

(M9)
$$concept\begin{bmatrix} \text{NAME} & \text{drugs} \\ \text{SCOPE} & [\text{stat}(DOA),\text{sec}(2)] \end{bmatrix}$$

We first give the translation of paragraph A of P1, the norm that makes use of this concept. It states that one ought not import into the Netherlands this type of drugs; the result of the translation is shown in M10.

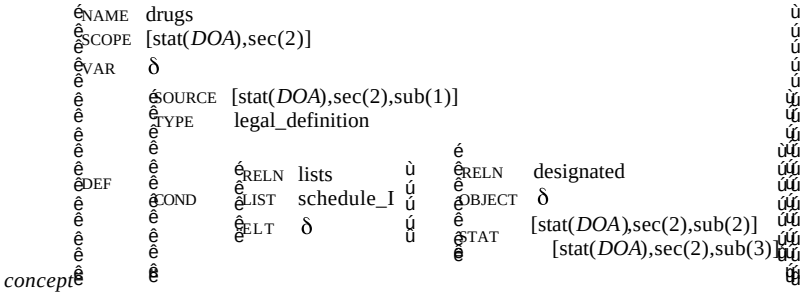
(M10)



We note that no restrictions hold for the SUBJECT of the norm, except that this must be the same person as the AGENT of the act that is prohibited; we can therefore confine ourselves to simply filling in the same tag, α, for both values.

We will now translate the definition of the concept 'drugs referred to in Section 2 of the DOA'. The resulting feature structure, M11, has structure M9, describing this concept, as its head.

(M11)

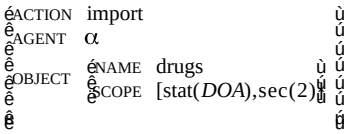


This is to be read as: an object, denoted by the variable δ, can be described by the concept name 'drugs' within the scope of Section 2 of the DOA, under the condition that this object is either listed by schedule I or designated by Section 2, Subsection 2 or Subsection 3 of the DOA.

5.2 Structure-preserving representation of composite concepts

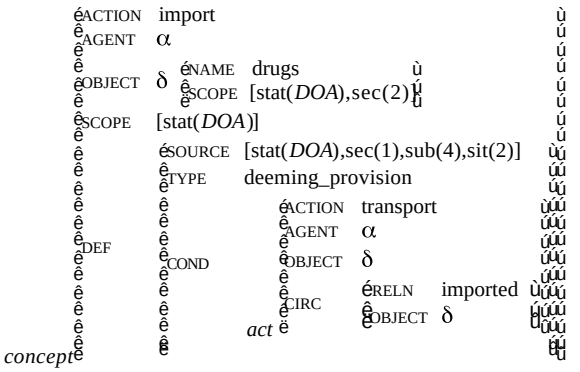
As we saw in section 2, the interpretation of the composite concept 'importing the drugs referred to in Section 2 of the DOA' is extended by the deeming provision P2. We represent this concept by feature structure M12.

(M12)



The deeming provision is translated into structure M13, which extends M12. We can read from the value for *definiens* (DEF) that the composite concept applies when the drugs (OBJECT, tag δ) that have been imported (CIRC) are transported.

(M13)



By using this structure-preserving representation of the concept 'importing the drugs referred to in Section 2 of the DOA', we have avoided the use of a hyphenated predicate as in the formalization of Section 2 of this article; the relations between this composite concept and the primitive concepts occurring in it are preserved in the representation and do not need separate specification. Complex references (i.e., other occurrences of the composite concept in the representation), such as the reference to this concept in the prohibition discussed above, refer by their form both to the concept as a whole and to its component parts.

6 Conclusions and further research

Deeming provisions modify an existing interpretation of a concept, by explicitly excluding parts of this interpretation, or by including other interpretations. When this concept is composite, occurrences of the concept elsewhere in legislation refer both to the concept as a whole, and to its component parts. These *complex references* pose problems for formalizations that do not preserve the internal structure of composite concepts in the representation.

In this paper, we have presented a new method for isomorphic representation of legal knowledge, in which the problems posed by complex references can be circumvented. The representation preserves the structural properties of composite concepts. The relations between a composite concept and its component parts can be derived from the syntax and do not need separate specification. Using an example from the Dutch Opium Act, we have shown that the representation of complex references is straightforward in our formalism.

It should be noted that, from a logical perspective, feature structures are not more expressive than first-order predicate logic. We think, however, that using feature structures as a representation formalism has clearly its merits from a conceptual point of view. Furthermore, the formalism can be extended in various ways, e.g., by including modal operators or default rules. Our main concern for the future will be to study the benefits of various extensions for the purpose of legal-knowledge representation.

References

Aït-Kaci, H. (1985). *A New Model of Computations Based on a Calculus of Type Subsumption.* Doctoral Thesis, University of Pennsylvania, Philadelphia, PA, USA.

Bench-Capon, T.J.M. and Coenen, F.P. (1992). Isomorphism and legal knowledge based systems. *Artificial Intelligence and Law*, Vol. 1, No. 1, pp. 65-86.

Bench-Capon, T.J.M., Robinson, G.O., Routen, T.W., and Sergot, M.J. (1987). Logic programming for large scale applications in law: a formalisation of the supplementary

benefit legislation. *Proceedings of the First International Conference on Artificial Intelligence and Law*, pp. 190-198.

Brachman, R. and Schmolze, J. (1985). An overview of the KL-ONE knowledge representation system, *Cognitive Science*, Vol. 9, pp. 171-216.

Karpf, J. (1989). Quality Assurance of Legal Expert Systems. *Jurimatics*, Vol. 8, Copenhagen Business School.

Kralingen, R.W. van (1995). *Frame-based Conceptual Models of Statute Law*, Ph.D. Thesis, University of Leiden. Kluwer Law International, The Hague, The Netherlands.

Pollard, C. and Sag, I.A. (1987). *Information-Based Syntax and Semantics. Volume 1: Fundamentals.* CSLI Lecture Notes, Vol. 13, Stanford, CA, USA.

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

Rounds, W.C. and Kaspar, R. (1986). A complete logical calculus for record structures representing linguistic information. *Proceedings of the Symposium on Logic in Computer Science*, pp. 34-43.

Sergot, M. (1991). The representation of law in computer programs. *Knowledge-based systems and legal applications* (ed. T. Bench-Capon), pp. 3-68. Academic Press Limited, London, England.

Shieber, S. (1986). *An Introduction to Unification-Based Approaches to Grammar.* CSLI Lectures Notes, Vol. 4, Stanford, CA, USA.

Valente, A. (1995). *Legal Knowledge Engineering; A Modelling Approach*, Ph.D. Thesis, University of Amsterdam. IOS Press, Amsterdam, The Netherlands.

Visser, P.R.S. (1995). *Knowledge Specification for Multiple Legal Tasks; A Case Study of the Interaction Problem in the Legal Domain*, Ph.D. Thesis, University of Leiden. Kluwer Law International, The Hague, The Netherlands.