On the Reusability of Ontologies in Knowledge-System Design

Pepijn Visser and Trevor Bench-Capon
Department of Computer Science, University of Liverpool
P.O. Box 147, L69 3BX Liverpool, United Kingdom
Phone: (+44) 151 - 794 3792
Email: {pepijn, tbc}@csc.liv.ac.uk

Abstract

In this paper we describe a case study which supports the claim that ontologies are reusable components in the design of knowledge systems. An ontology documents important domain assumptions which would otherwise remain implicit. Whereas a conceptual (or formal) system specification differs between different knowledge systems (even in the same domain), we show the underlying ontology to be invariant. This makes ontologies reusable for knowledge-system design. We illustrate this by discussing how a single legal ontology has been used for the construction of both a planning and an assessment system and argue that the same ontology can be reused for other knowledge systems as well.

1. Reusable components

There are important problems in reusing knowledge representations. Building knowledge systems usually entails constructing a new knowledge base from scratch (*e.g.*, [16]). One of the reasons why reusability is impeded is the difficulty in identifying and isolating the reusable components. High internal couplings in knowledge systems prevent the reusable system components to be isolated from the other components of the system [6]. As an instance of such couplings we mention the coupling between a knowledge representation and the purpose for which it is made, known as the *interaction problem* ([5]; [21]).

The purpose of making reusable components is to create libraries. By selecting an appropriate component from a library we can avoid building the component from scratch. The research conducted in this field covers various aspects of knowledge-system design. Besides reusing components of the executable code it also addresses the reuse of components from intermediate models (*viz.* models created while developing the executable model). Examples in this field are the reuse of conceptual task specifications (*e.g.*, [3], [4]), the reuse of conceptual specifications of problem-solving methods (*e.g.*, [14]), the reuse of conceptual domain specifications (*e.g.*, [12]), the reuse of formal task specifications (*e.g.*, [11]), the reuse of formal specifications of problem-solving methods (*e.g.*, [10]), the reuse of formal domain specifications (*e.g.*, [2], [13], [21]), and the reuse of software components (*e.g.*, [1], [19]). Recently, the reuse of ontologies (discussed below) has received a lot of attention (*e.g.*, [6], [7], [15], [24]).

In this paper we report on a case study on the feasibility of ontologies as reusable components. We show how a single legal ontology is used (and reused) for two prototype systems and illustrate the possibilities of using it for other systems as well. In section 2 we briefly elaborate on ontologies and their role in the design of knowledge systems. In section 3 we discuss the legal ontology used in this article (a partial ONTOLINGUA specification is given in an appendix). Then, in section 4 we show the applicability of the ontology for an assessment system, and, in section 5, for a planning system. Thereafter, in section 6, we argue that the ontology can be reused without modifications for other systems as well. Finally, in section 7 we draw conclusions.

2. Ontologies as reusable components

An ontology is an explicit conceptualisation of a domain, it describes the entities and relations taken to exist in the domain [7]. It is considered a meta-level description with respect to knowledge models in that it describes the building blocks of these models [9]. Thus, an ontology differs from these models because it only provides the elements with which the knowledge will be expressed in these models and not the knowledge itself.

Ontologies explicitly document assumptions about the domain being modelled. As such they allow us to communicate and discuss the assumptions about what does, and what does not exist in the domain as it is conceptualised in the model. Ontologies are useful in five areas:

(1) *Domain-theory development* Explicit documentation allows analysis and comparison of different domain theories;
(2) *Knowledge acquisition* Ontologies describe and structure the entities and relations that need to be acquired in the domain;
(3) *Knowledge-system design* Ontologies are reusable constructs in the design of knowledge systems;
(4) *System documentation* Ontologies provide a meta-level view (vocabulary and structure) on their domain which facilitates adequate system documentation;
(5) *knowledge exchange* Ontologies can be used to define assumptions that enable knowledge exchange between different agents.

In the next sections we illustrate the use of ontologies for knowledge-system design by describing one legal ontology that is used for different applications.

3. An ontology of the legal domain

Van Kralingen [12] and Visser [21] describe an ontology for the legal domain. In this section we provide a brief (and informal) discussion of this ontology. The ontology is divided into two separate ontologies: the *legal ontology* and the *statute-specific ontology*. We discuss both ontologies here (a fuller discussion can be found in [12] and [21]).

Legal ontology

The legal ontology, in contrast to the statute-specific ontology, is the generic and reusable part of the ontology (in the appendix we present a formal ONTOLINGUA specification of the legal ontology). It divides legal knowledge into three distinct structured entities: norms, acts and concept descriptions. Below, we briefly discuss these entities.

(a) *Norms* are the general rules, standards and principles of behaviour that subjects of law are enjoined to comply with. In the ontology a norm comprises the following eight elements: (1) a norm identifier, (2) a norm type, (3) a promulgation, (4) a scope, (5) conditions of application, (6) a norm subject, (7) a legal modality, and (8) an act identifier.

(b) *Acts* represent the dynamic aspects which effect changes in the state of the world. Within the category of acts we make two distinctions. The first distinction is between *events* and *processes*. Events represent an instantaneous change between two states, while processes have duration. The second distinction is between *institutional acts* and *physical acts*. The former type of acts are considered legal (institutional) interpretations of the (physical) acts that occur in the real world (more precisely: an institutional act is a legal qualification of a physical act). For example, the physical act of homicide may be any of the institutional acts of murder, manslaughter, or justifiable homicide. We note that these two distinctions result in four different types of acts. All acts are assumed to have the following fourteen elements: (1) the act identifier, (2) a promulgation, (3) the scope, (4) the agent, (5) the act type, (6) the modality of means, (7) the modality of manner, (8) the temporal aspects, (9) the spatial aspects, (10) the circumstantial aspects, (11) the cause of the action, (12) the aim of the action, (13) the intentionality of an action, and (14) the final state.

(c) *Concept descriptions* deal with the meanings of the concepts found in the domain. Concepts may be described by definitions or deeming provisions; in either of which case can their application can be definitively determined. In the case of definitions the description provides necessary and sufficient conditions. In deeming provisions the description establishes a legal fiction. Finally, there are concepts described by factors, which either establish a sufficient condition or indicate some contribution to the applicability of the concept (to be considered in relation to other factors). Concept descriptions comprise the following seven elements: (1) the concept to be described, (2) the concept type, (3) the priority, (4) the promulgation, (5) the scope, (6) the conditions under which a concept is applicable, and (7) an enumeration of instances of the concept.

Statute-specific ontology

The legal ontology contains constructs that are thought to be generic for the legal domain. That is, norms, acts and concept descriptions are considered to be present in any legal domain. However, modelling a particular legal subdomain also involves deciding upon numerous ontological questions. For instance, is it necessary to distinguish between male and female employers in the Unemployment Benefits Act? This motivates the distinction between the legal and the statute-specific ontology. Basically, the statute-specific ontology provides the vocabulary for describing the knowledge of the domain. We note that the statute-specific ontology cannot be reused for other domains, and should always be created for each legal sub-domain under consideration (though it should support various tasks in that sub-domain). Because in this article, we are mainly interested in the reusable (part of the) ontology we will not elaborate on this ontology any further.

4. An assessment system

The ontology describes how the domain is carved up into representational primitives. Using these primitives the actual domain knowledge can be expressed. In our case study we have used the ontology to formalise a substantial part of the Dutch Unemployment Benefits Act (DUBA). The ontology has been used to create a conceptual model of the domain knowledge in the DUBA [12]. This conceptual domain model has been used in the construction of a formal assessment model and an implementation in PROLOG [22]. In this section we confine ourselves to an informal description of the assessment model (an extensive discussion of the assessment model can be found in [21]. Our description of the model is structured according to the CommonKADS expertise model, *viz.* addressing domain knowledge, task knowledge and inference knowledge, respectively [3].

Domain knowledge The domain knowledge consists of a set of five domain models. Four of these domain models are derived directly from the structured entities identified in the legal ontology; the domain models of norms, events, processes and concept descriptions. The last domain model contains integrity constraints (used to keep the conclusion sets consistent). Note, that there are two different domain models for the representation of acts, namely one for events (which occur instantaneously), and one for processes (which have a duration). Each of these structured entities has a frame structure which contains slots for all elements prescribed by the ontology (see previous section). The language used to fill the slots in these frames is basically a reified version of the statute-specific ontology (this will also be used for the planning system described in section 5).

Task knowledge Assessment is a task in which it is determined whether a problem case can be classified as an instance of a given category. In our legal assessment expertise model the task attempts to find an institutional interpretation of a problem case stated in physical (*viz.* non-institutional) terms while the category is defined as a subset of the institutional interpretation. The problem case is expressed as a sequence of states and state-changes (either: the occurrence of an event, the start of a process, or the end of a process), expressed in physical terms only. Assessment is performed by trying to find institutional concepts, institutional events and institutional processes that match the physical concepts, physical events and physical processes in the problem case. This restating of the problem case also enables us to determine whether norms have been breached.

Inference knowledge The primitive inference steps in the assessment expertise model consist of the determination whether certain conditions are satisfied. The most important inferences determine whether (a) the conditions of a concept frame are satisfied, (b) the initial conditions of an event or process are satisfied, (c) the final conditions of an event or process are satisfied (d) a norm is applicable, (e) an agent is capable of performing an event or process, and (f) a norm has been breached.

5. A planning system

The same ontology that has been used in the creation of an assessment expertise model is also used in the creation of a planning expertise model (for a full description, see [21]).

Domain knowledge The domain knowledge used in the planning expertise model is largely the same as the domain knowledge in the assessment model (and hence, grounded in the ontology described in section 3). However, there are some differences in the way the domain knowledge is used. In the assessment system events and processes are used to determine whether a certain state-change could be classified as an institutional act. In the planning system events and processes are used to bring about new states. Whereas the assessment expertise model was shown to use the institutional events and processes, the planning expertise model mainly uses the physical events and processes (institutional events and processes are used in the planning expertise model, but only to determine whether the goal state has been reached). Because institutional events and processes are only used to classify physical acts they cannot be used as operators that change the world itself. Hence, the planning system has to use physical events and processes to propose a plan in the world.

Task knowledge Planning is finding a sequence of actions that transforms an initial state into a goal state. Our planning expertise model implements a simple non-hierarchical planning method. That is, planning proceeds by stringing together primitive (*viz.* neither composed nor abstract) acts from the initial state and determining whether the goal state has been reached. This proceeds until a goal state or a given search depth has been reached.

Inference knowledge The primitive reasoning steps in planning mainly concern the determination of new (physical) states in the planning search space. Hereto, the physical events and processes are examined for the results they bring about in the state under consideration. Other inferences concern the determination whether processes, that have been started in the past, will terminate in the state under consideration.

6. Using the ontology for other systems

In the previous sections we have illustrated how the ontology is used to create two expertise models. Both models have been implemented in PROLOG, resulting in a prototype system FRAMER (the name FRAMER is used to denote both the assessment and the planning system) [22]. This illustrates the reusability of the ontology for an assessment and a planning system.

However, we have not yet shown that the ontology is the only reusable component. In fact, it is not only the ontology that is reused for FRAMER. Although we illustrated the expertise models of the two systems to be different, FRAMER uses only one (conceptual and formal) domain specification (in fact, this has been an aim of the research in [21]). Accordingly, one could argue that it is not only the ontology that is the reusable but also the conceptual and formal domain specifications. With respect to the expertise models described here this is true. However, we believe that the ontology has a greater reusability than the conceptual and formal domain specification (*cf.* [12]). Below, we argue that the ontology has a higher reusability than the (conceptual and formal) domain specification.

An ontology is a meta-level description of a domain and is thus necessarily an abstraction of the domain specification. Creating a domain specification, either conceptual or formal, implies that design commitments have to be made. Usually, the nearer a system model gets to its implementation, the more design commitments are made for the description. Naturally, an abstract description, like an ontology has fewer details (and thus fewer design decisions), than the less abstract conceptual and formal models. For this reason we claim that the ontology is more likely to be a reusable component when a different system is to be developed. We illustrate this claim with three examples (*cf.* [12]).

The first example concerns the method chosen for our planning system. The domain specification described in sections 4 and 5 is intended to make as few commitments to tasks and methods as possible. That is, it is created with the intention to support as many tasks and methods as possible. Despite this intention, the specification is still tailored to the methods chosen. If other methods had been chosen, it could have been necessary to modify the domain specification [3]. This holds in particular for the (non-hierarchical) planning method. The non-hierarchical planning method implemented in the system requires atomic planning operators. If we had implemented a hierarchical planner, then there would have been a need for abstract planning operators (*e.g.*, [18]). Similarly, if we had implemented a script-based planner we would have needed to represent scripts or skeletons of plans (*e.g.*, [8]). We note, that although the domain specification would have changed for a hierarchical planner and the script-based planner, the ontology would not have to be revised. Both the abstract plans and the scripts can be considered instances of events and processes as defined in the ontology.

The second example concerns the use of the ontology in structure preserving representation of law. Peek [17] has used the theory of Van Kralingen for the representation of law using so-called 'feature structures' (*e.g.,* [20]). Although he does not use the ontology explicitly he implements the same theoretical distinctions made in the ontology. For this reason we consider the work of Peek as an example of the use of the ontology.

The third example concerns the use of the ontology in the process of drafting regulations. Voermans [23] has used the ontology for his LEDA system. LEDA, which is considered to be an information system rather than a knowledge system, implements guidelines of the Dutch Ministry of Justice for drafting regulations. The ontology, that is, the distinctions made in the ontology, are used to guide knowledge acquisition.

7. Conclusions

In this paper we have reported on a case study concerning the feasibility of ontologies as reusable knowledge-system components. We have shown that one ontology of the legal domain is (re)usable for two different expertise models (and thus for two different systems). Moreover, we have argued that the ontology can be used for other systems as well. The main conclusions drawn from this case study are:

- ontologies are useful to reveal domain theoretical design decisions underlying knowledge systems;
- ontologies are reusable components of knowledge systems, and in the design of a knowledge system it is useful to have a library of ontologies.
- the creation of an ontology (or the selection of an ontology from a library) should be a separate design phase in the creation of any substantial knowledge system. This design phase should precede the creation of the conceptual model.

Appendix: ONTOLINGUA specification of the legal ontology (partial)

```
(In-Package "ONTOLINGUA-USER")

;;; Written by user P.R.S.Visser owned by group
;;; JUST-ME, Date: Dec 21, 1995  14:17

(Define-Ontology
    Legal-Ontology
    (Frame-Ontology)
    "This is the ONTOLINGUA specification of the
    legal ontology described by Van Kralingen
    (1995) and Visser (1995a)."
    :Io-Package
    "ONTOLINGUA-USER"
    :Intern-In
    ((Frame-Ontology Arity Thing Subclass-Of Class
    Instance-Of Documentation Slot-Cardinality)
     (Kif-Numbers Number) (Kif-Sets Member)))

(In-Ontology (Quote Legal-Ontology))

;;; ------------------ Classes --------------
;;; Act

(Define-Frame Act
    :Own-Slots
    ((Arity 1)
     (Documentation "The class of acts contains
    all occurences that are initiated by human
    beings.")
     (Instance-Of Class)
     (Subclass-Of Legal-Knowledge))
    :Template-Slots
     ((Act-Identifier (Slot-Cardinality 1))
      (Promulation (Slot-Cardinality 1))
      (Scope (Slot-Cardinality 1))
      (Agent (Slot-Cardinality 1))
      (Act-Type (Slot-Cardinality 1))
      (Modality-Of-Means (Slot-Cardinality 1))
      (Modality-Of-Manner (Slot-Cardinality 1))
      (Temporal-Aspects (Slot-Cardinality 1))
      (Spatial-Aspects (Slot-Cardinality 1))
      (Circumst-Aspects (Slot-Cardinality 1))
      (Cause (Slot-Cardinality 1))
      (Aim (Slot-Cardinality 1))
      (Intentionality (Slot-Cardinality 1))
      (Final-Sate (Slot-Cardinality 1))))

;;; Concept-Description

(Define-Frame Concept-Description
    :Own-Slots
    ((Arity 1)
     (Documentation "A concept description lays
    down the meaning of a legal term.")
     (Instance-Of Class)
     (Subclass-Of Legal-Knowledge))
    :Template-Slots
     ((Concept (Slot-Cardinality 1))
      (Concept-Type (Slot-Cardinality 1))
      (Priority (Slot-Cardinality 1))
      (Promulgation (Slot-Cardinality 1))
      (Scope (Slot-Cardinality 1))
      (Conditions (Slot-Cardinality 1))
      (Instances (Slot-Cardinality 1))))

;;; Concept-Description-Type

(Define-Class Concept-Description-Type
    (?Type)
```

```
    "A concept-description type is an element from
    the set {Definition-Type, Deeming-Provision-
    Type, Factor-Type, Meta-Type}."
    :Iff-Def
    (member ?Type
       (Definition-Type, Deeming-Provision-Type,
    Factor-Type)))

;;; Deeming-Provision

(Define-Frame Deeming-Provision
    :Own-Slots
    ((Arity 1)
     (Documentation "Deeming provisions lay down
    the meaning of a concept by stating sufficient
    conditions for the concept to be classified
    under the heading. They differ from the
    definitions in that they establish a legal
    fiction.")
     (Instance-Of Class)
     (Subclass-Of Concept-Description))
    :Template-Slots
     ((Concept-Type Deeming-Provision-Type)))

;;; Definition (left out)
;;; Factor (left out)
;;; Institutional-Act

(Define-Frame Institutional-Act
    :Own-Slots
    ((Arity 1)
     (Documentation "An institutional act is an
    act as denoted in a legal source.")
     (Instance-Of Class)
     (Subclass-Of Act)))

;;; Institutional-Event (left out)
;;; Institutional-Process

(Define-Frame Institutional-Process
    :Own-Slots
    ((Arity 1)
     (Documentation "An institutional process is
    an intitutional act that has a duration.")
     (Instance-Of Class)
     (Subclass-Of Institutional-Act)))

;;; Legal-Knowledge

(Define-Class Legal-Knowledge
    (?X)
    "The class of legal knowledge is the root of
    all other classes."
    :Def
    (And (Class ?X)))

;;; Legal-Modality

(Define-Class Legal-Modality
    (?Modality)
    "The legal modality of (a norm) is an element
    from the set {Can, Ought, Ought-not, May}."
    :Iff-Def
    (Member ?Modality (Can, Ought, Ought-not,
    May)))

;;; Norm

(Define-Frame Norm
    :Own-Slots
    ((Arity 1)
     (Documentation "A norm is a statement to the
    effect that something ought (not) be done.")
     (Instance-Of Class)
     (Subclass-Of Legal-Knowledge ?X))
    :Template-Slots
    ((Identifier (Slot-Cardinality 1))
     (Norm-Type (Slot-Cardinality 1))
     (Promulgation (Slot-Cardinality 1))
     (Scope (Slot-Cardinality 1))
     (Conditions (Slot-Cardinality 1))
     (Subject (Slot-Cardinality 1))
     (Legal-Modality (Slot-Cardinality 1))
     (Act-Identifier (Slot-Cardinality 1))))

;;; Norm-Of-Competence (left out)
;;; Norm-Of-Conduct

(Define-Frame Norm-Of-Conduct
    :Own-Slots
    ((Arity 1)
     (Documentation "The norm of conduct is a norm
    that imposes duties to people in society.")
     (Instance-Of Class)
     (Subclass-Of Norm))
    :Template-Slots
    ((Norm-Type Conduct)))

;;; Norm-Types

(Define-Class Norm-Type
    (?Type)
    "The norm type of a norm is an element from
    the set {Conduct, Competence}."
    :Iff-Def
    (Member ?Type (Conduct, Competence)))

;;; Physical-Act

(Define-Frame Physical-Act
    :Own-Slots
    ((Arity 1)
     (Documentation "A physical act is an act that
    is assumed to occur in the (external) world.
    In contrast to institutional acts, physical
    acts can be performed in the world.")
     (Instance-Of Class)
     (Subclass-Of Act)))

;;; Physical-Event (left out)
;;; Physical-Process (left out)

;;; ------------------ Relations --------------
;;; Event-Qualification

(Define-Relation Event-Qualification
    (?Physical-Event ?Institutional-Event)
    "An Event-Qualification is a qualification of
    a physical event as an institutional event."
    :Def
    (And
      (Physical-Event ?Physical-Event)
     (Institutional-Event ?Institutional-Event)))

;;; Process-Qualification (left out)

;;; ------------------ Functions --------------
;;; Normative-Status

(Define-Function Normative-Status
    (?Norm)
    :->
    ?Status
    "The normative status of a norm is a function
    from a norm (instance) onto an element from
    the set {Breached, Not-breached}."
    :Iff-Def
    (And
      (Norm ?Norm)
      (Member ?Status (Breached Not-Breached))))
```

References

[1] Bakker, H. (1995). *Object-oriented Modelling of Information Systems; The INCA Conceptual Object Model*, Doctoral Thesis, University of Limburg, Maastricht, The Netherlands.

[2] Bench-Capon, T.J.M. (1991). *Knowledge-Based Systems and Legal Applications*, APIC series, No. 36, Academic Press, London, United Kingdom.

[3] Breuker, J.A., and W. van de Velde (1994). CommonKADS Library for Expertise Modelling, Reusable Problem Solving Components, J.A. Breuker, and W. van de Velde (eds.), IOS Press, Amsterdam, the Netherlands.

[4] Breuker, J.A., and B.J. Wielinga (1989). Models of Expertise in Knowledge Acquisition, Topics in Expert System Design, G. Guida, and C. Tasso (eds.), pp.265-295, Elsevier Science Publishers B.V. (North-Holland), Amsterdam, the Netherlands.

[5] Bylander, T., and B. Chandrasekaran (1987). Generic tasks for knowledge-based reasoning: the "right" level of abstraction for knowledge acquisition, International Journal of Man-Machine Studies, Vol. 26, pp.231- 243.

[6] Gruber, T.R. (1991). The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases, Principles of Knowledge Representation and Reasoning, KR'91, Proceedings of the Second International Conference, J. Allen, R. Fikes, and E. Sandewall (eds.), pp.601-602, Cambridge, Massachusetts, United States.

[7] Gruber, T.R. (1993). Toward Principles for the Design of Ontologies Used for Knowledge Sharing, technical report KSL-93-4, Knowledge Systems Laboratory, Stanford University, Stanford, United States.

[8] Hammond, K.J. (1986). CHEF: A Model of Case-based Planning, Proceedings AAAI-86, pp.267-271, Cambridge, Massachusets, AAAI Press/MIT Press, Cambridge, United States.

[9] Heijst, G. van (1995). *The Role of Ontologies in Knowledge Engineering*, Doctoral Thesis, University of Amsterdam, Amsterdam, the Netherlands.

[10] Karbach, W., M. Linster, and A. Voss (1990). Models, methods, roles and tasks: many labels - one idea?, Knowledge Acquisition, Vol. 2, pp.279-299.

[11] Kowalczyk, W. and J. Treur (1990). On the use of Formalised Generic Task Models in Knowledge Acquisition, *Current trends in Knowledge Acquisition*, IOS Press, Amsterdam, the Netherlands, pp.198-221.

[12] Kralingen, R.W. van (1995). Frame-based Conceptual Models of Statute Law, Computer/Law Series, Kluwer Law International, The Hague, The Netherlands.

[13] Lenat, D.B., and R.V. Guha (1990). Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project. Addison-Wesley, Reading, Massachusetts, United States.

[14] McDermott, J. (1988). Preliminary Steps Toward a Taxonomy of Problem-Solving Methods, Automating Knowledge Acquisition for Expert Systems, S. Marcus (ed.), pp.225-256, Kluwer Academic Publishers, Dordrecht, the Netherlands.

[15] Musen, M.A. (1992). Dimensions of Knowledge Sharing and Reuse, *Computers and Biomedical Research*, 25, pp.435-467.

[16] Patil, R.S., R.E. Fikes, P.F. Patel-Schneider, D. Mckay, T. Finin, T.R. Gruber, and R. Neches (1992). The DARPA Knowledge Sharing Effort: Progress Report, Prinicples of Knowledge Representation and Reasoning, Proceedings of the Third International Conference, B. Nebel, C. Rich, and W. Swartout (eds.), pp.777-788, Cambridge, Massachusetts, United States.

[17] Peek, N. (1995). Structure Preserving Representations of Complex References, *Legal Knowledge Based Systems, Telecommunications and AI & Law*, J.C. Hage, T.J.M. Bench-Capon, M.J. Cohen, and H.J. van den Herik (*eds*.), Koninklijke Vermande B.V. Lelystad, the Netherlands, pp.95-104.

[18] Sacerdoti, E.D. (1974). Planning in a Hierarchy of Abstraction Spaces, *Artificial Intelligence*, Vol. 5, pp.115-135.

[19] Schach, S.R. (1994). The Economic Impact of Software Reuse on Maintenance, Software Maintenance: Research and Practice, Vol. 6, pp.185-196.

[20] Shieber, S. (1986). An introduction to Unification-Based Approaches to Grammar, *CSLI Lecture Notes*, Vol. 4, Standord, United States.

[21] Visser, P.R.S. (1995a). *Knowledge Specification for Multiple Legal Tasks; A Case Study of the Interaction Problem in the Legal Domain*, Computer/Law Series, No. 17, Kluwer Law International, The Hague, the Netherlands.

[22] Visser, P.R.S. (1995b). *FRAMER: Source Code of a Legal Knowledge System that performs Assessment and Planning*, Reports on Technical Research in Law, University of Leiden, Leiden, the Netherlands, Vol. 2, No 1.

[23] Voermans, W. (1995). *Sturen in de mist ... maar dan met radar*; *de mogelijkheid van de toegepaste informatica bij het ontwerpen van regelgeving*. Doctoral thesis, Catholic University of Brabant, W.E.J. Tjeenk Willing, Zwolle, the Netherlands (in Dutch).

[24] Wielinga, B.J., G. Schreiber, W. Jansweijer, A. Anjewierden, F. van Harmelen (1994). *Framework and Formalism for Expressing Ontologies*, DO1b.1-Framework-1.1-UvA-BW+GS+WJ+AA, University of Amsterdam, the Netherlands.