Int. J. Human-Computer Studies (1999) **51,** 1127–1154
Article No. ijhc.1999.0300
Available online at http://www.idealibrary.com on IDEAL®

A principled approach to developing legal knowledge systems†

ROBERT W. VAN KRALINGEN
Center for Law, Public Administration and Informatization, Faculty of Law,
Tilburg University, P.O. Box 90153, 5000 LE, Tilburg, The Netherlands.
email: robert.w.van.kralingen@nl.arthurandersen.com

PEPIJN R. S. VISSER AND TREVOR J. M. BENCH-CAPON
LIAL-Legal Informatics at Liverpool, Department of Computer Science,
University of Liverpool, P.O. Box 147, Liverpool, L69 7ZF, UK.
email: pepijn.visser@sap.com, tbc@csc.liv.ac.uk

H. JAAP VAN DEN HERIK
Department of Law and Informatics, Faculty of Law, University of Leiden,
P.O. Box 9521, 2300 RA, Leiden, The Netherlands. email: herik@cs.unimaas.nl

In this article we present a principled, four-phased approach to the development of legal knowledge systems. We set out from the well-studied CommonKADS method for the development of knowledge systems and tailor this method to the legal domain. In particular, we propose a generic legal ontology, and describe the creation of statute-specific ontologies to adopt the method for building legal systems. In the construction of these ontologies, we start from a theoretical analysis of the legal domain. The well-known example of the Imperial College Library Regulations (ICLR) is used to illustrate the method. © 1999 Academic Press

1. Introduction

Several methods are available for the design of computer system. The essence of these methods is the division of the system-development process into a number of comprehensible phases. The result of each phase is a model of specific aspects of the system. Examples of such models are: organizational models, addressing the system in its organizational context, and functional models, specifying the tasks of the system.

Despite the attention system-development methods have received in the field of computer science and knowledge systems, they have not been widely applied in the domain of artificial intelligence and law. Hardly any research has been reported on the process of designing legal knowledge systems as such. In this article, we address the design of legal knowledge systems from a methodological point of view. In particular, we present a method for the stepwise construction of legal knowledge systems, showing

† This article is an extended version of Visser, Van Kralingen and Bench-Capon (1997) which was presented at the sixth International Conference on Artificial Intelligence and Law in Melbourne, Australia.

1071-5819/99/121127 + 28 $30.00/0

four major design phases: analysis, conceptual modelling, formal modelling and implementation. Our point of departure is the CommonKADS method for knowledge-system development (e.g. Breuker & Van de Velde, 1994). The method is tailored to the legal domain by adding domain-specific elements. Moreover, we focus primarily on the conceptual and formal models of the system. Also, we present heuristics for assembling these models. Our method is illustrated by discussing the creation of a small knowledge system that operates on an extended version of the well-known Imperial College Library Regulations (Jones & Sergot, 1992).

The outline of the article is as follows. In Section 2, we begin with a short description of the legal-theoretical background. Next, an overview of the method is provided in Section 3. In Section 4, the conceptual frame-based ontology is introduced after which we elaborate on three of the four phases. In Section 5 we address the analysis phase, in Section 6 the conceptual modelling phase and in Section 7 the formal modelling phase (the implementation phase is not described in this article). Finally, we conclude by presenting our main findings in Section 8.

2. Legal-theoretical background

Stepwise construction of knowledge systems by developing intermediate models greatly facilitates bridging the gap between knowledge and knowledge system. The models consecutively developed during the construction process can be viewed as intermediate representations of the system. Bench Capon, Robinson, Routen and Sergot (1987) list three advantages of such intermediate models. First, they impose structure on knowledge acquisition and knowledge modelling. Second, they make the interpretation of the knowledge to be contained in the knowledge system better intelligible. Third, they allow for a representation of knowledge that does not have to commit to the quirks of the implementation language.

In order for an intermediate representation to fulfil its function, it must meet certain requirements. In our opinion, the most important requirement is that the representation must comply with ideas domain experts have on the structure of the domain to be represented. Since we take the legal domain as our research domain, we have taken legal theory as the point of departure for the outline of our conceptual and formal models. We start from institutional theories of law, such as the ones proposed by MacCormick and Weinberger (1986), and Ruiter (1993). Both MacCormick and Weinberger, and Ruiter turn to the theory of speech acts (Searle, 1969) in their analysis of the legal domain. In speech-act theory, the concept "institutional fact" plays an important role. An institutional fact can be seen as "an abstract, socially defined entity or event". The legal domain contains many of these entities and events. We name some instances: legal institutions, legal definitions, legal performatives, juridical acts and legal norms all qualify as institutional facts. Since the norm is the most salient construct of the legal domain, we have selected it as the point of departure for our intermediate representations.

Norms are the most important elements of legal systems (see, for instance, Hart, Kelsen, Von Wright, Ross, etc.). An adequate definition of a norm is "a statement to the effect that something ought to, ought not to, may or can be done" (cf. Von Wright, 1963, 1983). Norms come in many shapes and forms. In most classifications of norms, a difference is observed between norms laying down standards of behaviour individuals and institutions must observe in going about their business and norms regulating the modification and application of existing norms and the creation of new norms. These types of norms can be coined "norms of conduct" and "norms of competence", respectively. Though there are numerous differences between norms of conduct and norms of competence, there are also numerous points of resemblance. For instance, in both cases actions may be criticized or assessed by reference to norms as legally the "right" or "wrong" thing to do. Norms of conduct and norms of competence introduce standards by which particular actions may be critically appraised.

Both main types of norms can be divided into several sub-types. Norms of conduct can be differentiated as duty-imposing and permissive norms,† as sein-sollen and tun-sollen norms,‡ as individual and general norms§ and as hypothetical and categorical norms.¶ Norms of competence can be divided into power-conferring norms†† as opposed to duty-imposing and permissive norms,‡‡ into individual and general norms, and into hypothetical and categorical norms. We do not elaborate on the peculiarities of the main or subtypes, but refer to Van Kralingen (1995).

Norms of conduct and norms of competence are by no means the only elements of legal systems. We have mentioned some other elements, such as legal institutions (e.g. marriage, ownership and personality) and legal definitions (e.g. a definitions of unemployment). We have stated previously, that in legal theory these phenomena have been approached from the angle of speech-act theory, and, more precisely, that they have been defined as institutional facts (e.g. MacCormick & Weinberger, 1986; Ruiter, 1993).

An institutional fact is a fact whose interpretation not merely depends upon the occurrence of acts or events in the world, but also on the application of rules to such acts and events. An example can help to clarify this definition: if we take the fact "I am not working", an example of the same fact seen from an institutional point of view could be "I am unemployed". The state of being unemployed, for instance in the context of the

† According to Hart, a norm is of the duty-imposing kind if it is "... conceived and spoken of as imposing obligations, when the general demand for conformity is insistent and the social pressure brought to bear upon those who deviate or threaten to deviate is great." (Hart, 1961, p. 84). In legal theory, permissive norms are sometimes not considered an independent type ('the mere absence of obligations'). However, there are persuasive arguments to maintain permissive norms as an independent norm type (see Van Kralingen, 1995, pp. 21–22).

‡ The distinction between *sin-sollen* and *tun-sollen* norms regards the object of the norm; it is possible that the object of a norm is neither an act nor an activity, but the mere existence of a certain state of affairs; these norms are not concerned with action, but envisage what may not or must not be. They are referred to as *sein-sollen* norms.

§ A norm can be individual or general, both with regard to the description of its object (an act, an activity or a state of affairs) and with regard to the description of its subject (the person or persons addressed by the norm). The standard form of, for instance, a criminal statute is general in two ways: it indicates a type of conduct and it applies to a class of persons who are expected to perceive that it applies to them and to comply with it (Hart, 1961, p. 21). However, a norm need not be general. For instance, if a norm prohibits a specified person to conduct a specified activity, it is an individual norm.

¶ The difference between categorical and hypothetical norms is the difference between norms which decree that a certain behaviour is obligatory *unconditionally* and those which decree that a certain behaviour is obligatory only *under certain conditions*.

†† Law has a peculiar property: law regulates its own modification, creation and application. Power-conferring norms install power on persons or institutions to fulfil this functions. Hart (1961, p. 33), for instance, typifies power-conferring norms as recipes for creating duties.

‡‡ Powers are seldom absolute. Duty-imposing and permissive norms of competence regulate application of powers conferred upon individuals or institutions through power-conferring norms.

Dutch unemployment benefits act, is a state that is defined by a rule (the rule providing a definition of unemployment). Since the interpretation of norms also relies heavily on rules, the definition applies to norms as well. This brings us to a classification of elements of legal systems with at the top institutional facts, and at lower levels norms, definitions, sub-types of norms and definitions, etc.

Each elements fulfils its own function within a legal system. However, the mere fact that different elements fulfil different functions by no means implies that they should be modelled differently. Rather than just linking a function to a modelling method or a modelling primitive, the composition of the different elements should be analysed in order to provide the modelling primitives. How this analysis helps us in defining an ontology of the domain is described in Section 4 (the conceptual frame-based ontology).

3. An overview of the method

Although there are several design methods of knowledge systems, the application of these methods to the legal domain is not yet widespread. Often, the design of a legal knowledge system (henceforth LKS) is a rather *ad-hoc* and ill-documented process. Our primary research aim has been to tailor an existing knowledge-system development technique to the legal domain, thus creating a dedicated method for the development of LKS. Ideally, such a method would provide guidance for all the steps in the design of an LKS, providing better support for the designers of LKSs than the more general methods.

The method presented here largely adopts the CommonKADS framework (Breuker & Van de Velde, 1994). An important feature of this method is the division of the design process into separate phases. In the spirit of this method we distinguish: (1) an *analysis phase*, (2) a *conceptual modelling phase*, (3) a *formal modelling phase* and (4) an *implementation phase*. As in CommonKADS, we specify an informal and a formal *expertise model* — in phases 2 and 3, respectively. In the model, we separate *domain knowledge* (specifying the static knowledge in the domain), and *control knowledge* (specifying how the domain knowledge is applied to achieve a goal). Control knowledge consists of specifications of *inferences* (primitive reasoning steps) and *tasks* (a control structure over tasks and inferences).

The CommonKADS method does not commit any domain in particular and does not give much guidance in the specification of legal domain knowledge (e.g. Gardner & Spelman, 1993; Visser, 1995). For our purpose, we propose to supplement CommonKADS with a *legal ontology*, as developed by Van Kralingen (1995) and Visser (1995). The most important feature of this ontology is the distinction between *norms*, *acts* and *concept descriptions*, each of which has an associated modelling template (viz. the *norm frame*, the *act frame* and the *concept-description frame*). We elaborate on this ontology in Section 4 and in Sections 6 and 7. In addition to CommonKADS and these frame structures, we use the domain-analysis method KANT (Bench-Capon, 1991; Bench-Capon & Coenen, 1992; Visser, Bench-Capon & Van den Herik, 1997). This method is used to define a *statute-specific ontology*, which lists the vocabulary (viz. predicates) with which to instantiate the frame structures. Below, we describe the four design phases in some detail [based on Van Kralingen (1995) and Visser (1995)].†

† Although the phases are largely executed in the order listed here the creation of an LKS will involve several iterations through all phases.

1. Analysis phase
   (a) *Domain identification*: identify the legal knowledge that is to be contained in the LKS in terms of references to legal sources (e.g. set of legal cases, articles in statutes, assumptions, heuristics).
   (b) *Task identification*: identify the task(s) that the LKS has to perform using the domain knowledge. In particular, this should result in a description of the input, the output, and the problem-solving goals of the LKS. Together, 1(a) and (b) are meant to determine the competence of the LKS.
2. *Conceptual modelling phase*
   (a) *Method description*:† provide an informal description of *how* the system will perform the task. Otherwise stated, describe the method used to achieve the problem-solving goals by transforming the input into the output (use, for instance, the CommonKADS library of tasks). The method specification provides guidance in the acquisition of the relevant domain knowledge [see step 2(c)]. The result of this step is a hierarchical decomposition of the main task into a series of sub-tasks. Also, the various tasks are allocated either to the system or to the user.
   (b) *Domain ontology selection and adaptation*: select an appropriate legal ontology and tailor the ontology—if necessary—to support the tasks and methods described. As stated before, we select the frame-based ontology of Van Kralingen and Visser, summarized in Section 4 (in the remaining steps of the method we take this ontology as the standard) but other ontologies can be chosen as well. For an overview of legal ontologies, we refer to Visser and Winkels (1997).
   (c) *Knowledge acquisition and modelling*: model the domain knowledge in accordance with the ontology. Here, this involves identifying the norms, acts and concept descriptions in the domain knowledge, and gathering the necessary information that is needed to instantiate the frame structures (viz. the domain ontology). The result of this step are instantiated frame structures, each with their contents described in (structured) natural language (shortly: the conceptual domain specification).
3. *Formal modelling phase*
   (a) *Determine boundaries of control and domain knowledge*: identify procedural knowledge embedded in the conceptual frame structures (viz. meta-level procedural norms of competence and conflict-resolution knowledge) and decide whether to model this knowledge in the expertise model as domain knowledge or as control knowledge. Conflict-resolution knowledge, for instance, can be modelled as control knowledge (e.g. in case we want to conduct explicit meta-level reasoning about conflicts), or as domain knowledge (e.g. in which case we will "compile out" conflicts). More details about this step can be found in Visser (1995).
   (b) *Define control knowledge*: create a formal description of the tasks, identified in the steps 1(a) and 2(a). This description should specify the hierarchical

† The use of the word "method" here concerns the method of the LKS and should not be confused with the system-design method presented in this article.

decomposition of tasks, and the information that is passed between tasks in a formal language. In the spirit of CommonKADS, we refer to the tasks at the lowest level of the hierarchy as inferences. How they apply knowledge contained in the frame structures will be specified in step 3(e).

(c) *Create statute-specific ontology*: this step is aimed at determining and defining the predicate relations that are used to express the domain knowledge in a formal language. It involves the application of the KANT method on the legal texts identified in step 1(a). In particular, this step involves: (c1) the creation of a *TOO* (*Test-On-Objects*) *structure* (identifying entities and the tests applied to them), (c2) the creation of a *EAW* (*Entity-Attribute-Value*) *structure* (by extracting the attributes of concern to the domain and identifying the possible values for these attributes from the TOO structure), (c3) the creation of a class hierarchy (grouping the entities in a class hierarchy and using inheritance to rationalize the attribution of attributes and values to the entities) and (c4) the selection of predicate names to model the class hierarchy.

(d) *Formalize domain knowledge*: model the knowledge described in the informal conceptual domain model by bringing together the formal ontology and the statute-specific ontology. This step results in the formal domain specification.

(e) *Define inferences*: define the inferences (primitive tasks) that link the control knowledge and the domain specification.

4. *Implementation phase*

(a) *Select language and platform*: select an appropriate language and platform to implement the formal descriptions of the tasks and inferences, and the domain specification.

(b) *Implementation*: implement the formal model in the chosen language on the chosen platform.

In the remainder of this article, we elaborate on phases 1–3. The scope of this article does not allow us to extensively discuss all aspects of the process. Our primary focus is on the modelling of domain knowledge in phases 2 and 3 (also we briefly address the specification of control knowledge in phase 3). In Sections 5–7 we illustrate the method by applying it to an extended version of the Imperial College Library Regulations (henceforth ICLR) example. However, before we apply our method, we introduce the conceptual frame-based ontology and the analysis underlying its form.

4. The conceptual frame-based ontology

As stated above, the backbone of the conceptual frame-based ontology is constituted by three frame structures: a norm frame, an act frame and a concept-description frame. The theoretical foundation laid down in Section 2 constitutes the point of departure for the analysis that has yielded the three frame structures and the elements (slots) distinguished within the structures.

In addition to the norm, act and concept-description frames the ontology comprises a vocabulary needed to instantiate the frame structures. This section ends with a concise description of the vocabulary.

4.1. THE NORM FRAME

A norm must convey information to fulfil its function of communicating standards of behaviour. The way in which one is expected to behave must be clear from the norm. Thus, a norm serves as a scheme of interpretation. A person's behaviour is measured against abstract descriptions of behaviour included in the norm. By this process, someone's behaviour acquires legal meaning (cf. Kelsen, 1978, pp. 4, 5; Hage, Leenes & Lodder, 1994).

It can be assumed that legal subjects attune their behaviour to the standards laid down in norms. A legal subject attempts to stay within the law and attempts to avoid the breach of norms. The reason for this behaviour, whether it is social pressure or threat of sanctions, is not of importance to our discussion of norms. What is important is the assumption that legal subjects can learn the behaviour expected of them by examining norms. Consequently, norms must answer questions about the behaviour expected of legal subjects. Brouwer (1990, p. 62) has formulated five of these questions.

(1) Who is obligated or permitted to do something?
(2) Is there an obligation or a permission to do something or to leave something undone?
(3) What must be done or forborne?
(4) Where must something be done or forborne?
(5) When must something be done or forborne?

If a norm provides us with an answer to all of these questions, it can be coined a complete norm. Consequently, a complete norm is defined as a norm that supplies information about how to answer each of these questions.

4.1.1. The elements of a norm

Before the essence of the concept "norm" can be captured in a structure, it must be established what the elements of a norm are. If we consider the five questions, a link with the classification of norms as given in the Section 2 can be found (below the link with the classification is placed in brackets). The first question, "who is obligated or permitted to do something?", is a question about the norm subject (individual and general norms). The second question, 'is there an obligation or a permission to do something or to leave something undone?', inquires after the legal modality of the norm (duty-imposing, permissive or power-conferring norms). The third question, "what must be done or forborne?', addresses the description of the act (individual and general norms, and sein-sollen and tun-sollen norms). The fourth and the fifth question, 'where must something be done or forborne?" and "when must something be done or forborne?", are also questions important to the description of the act (individual and general norms). The five questions form a stepping stone in the determination of the elements of a norm. The questions could be said to inquire after the various elements of a norm: the first three questions address the norm subject, the legal modality and the act description (norm object), respectively. The relation between these elements is as follows: the norm subject is commanded, prohibited, permitted or empowered (legal modality) to perform an act (act description).

The importance of the norm subject, legal modality and act description is acknowledged by most authors. However, there is no consistent terminology to denote the elements. The norm subject is sometimes denoted by the term “addressee” (e.g. Kelsen 1991). The element of a norm that determines whether we are dealing with a duty-imposing, permissive or power-conferring norm is referred to as the “deontic modality” (Brouwer, 1990), the “deontic operator” (Aarnio, 1987), the “norm character” (Von Wright, 1963), the ‘function of a norm” (Kelsen, 1991), the “directive operator” (Ross, 1968) or the legal modality (Ross, 1968). We will use the term “legal modality”. The description of the act (Brouwer, 1990) is also known as the “content of a norm” (Von Wright, 1963), the “object of a norm” (Kelsen, 1991), or the “theme of a norm” (Ross, 1968). In addition to the aforementioned elements of a norm, most authors name the “conditions of application” or “norm conditions” as a fourth norm element (hypothetical and categorical norms in the classification of Section 2). This element of a norm is only relevant when dealing with a hypothetical norm.

The four norm elements mentioned above constitute what has been called the norm kernel (Von Wright, 1963, p. 70). Every norm must comprise a norm subject, a legal modality and an act description. If one of these elements is missing (i.e. the elements is neither explicitly nor implicitly present), we are not dealing with a complete norm; the norm does not answer the previously formulated questions. Dependent on the type of norm, categorical or hypothetical, these three elements can be supplemented with conditions of application.

To be able to refer to a norm as a separate entity and to link a norm to its origin, some additional norm elements are introduced: the norm identifier, the norm promulgation and the scope. The norm identifier is used as a point of reference for a norm, the norm promulgation links a norm to its source (e.g. an article in a statute), and the scope limits the range of application of a norm (e.g. the range of application of a norm can be limited to the context of one statute). To distinguish these norm elements from the four primary elements, they are denoted as auxiliary elements.

Now that the primary and auxiliary elements of a norm have been introduced, they can be placed in a structure: a norm frame. A norm frame can be seen as a template to represent norms. In Table I, we present a norm frame. The frame is used to represent the elements of a norm coherently. Each row of the table is referred to as a slot of the norm frame. The second column of the table contains the elements of a norm, the third column contains a typification of the element, the fourth column refers to the status of the element (i.e. is it a primary or an auxiliary element and is it obligatory or optional to have the element present in a norm frame).

The first, third and fourth slots are reserved for the auxiliary elements of a norm. They contain the norm identifier, the promulgation and the scope, respectively. The elements contained in these slots do not form an essential part of the norm; the norm identifier, for instance, is used as a point of reference for the norm. The second, fifth, sixth, seventh and eighth slots hold the primary elements of a norm. In the second slot, the norm type, we specify whether a norm is a norm of conduct or a norm of competence. The fifth slot facilitates the representation of the conditions of application of a norm. The slot is only instantiated when we deal with a hypothetical norm. The sixth slot, the norm subject, contains the persons or institutions to whom the norm is addressed. The seventh slot accommodates the legal modality. The legal modality determines the function of a norm; a norm is either an obligation (a command or a prohibition; ought and ought not, respectively), a permission (may), or a power-conferring norm (can). The combination of

TABLE 1
A norm frame

	Element	Typification	Station
1	Norm identifier	The norm identifier (used as a point of reference for the norm)	Auxiliary, obligatory
2	Norm type	The norm type (norm of conduct of norm of competence)	Primary, obligatory
3	Promulgation	The promulgation (the source of the norm)	Auxiliary, obligatory
4	Scope	The scope (the range of application of the norm)	Auxiliary, obligatory
5	Conditions of application	The conditions of application (the circumstances under which a norm is applicable)	Primary, optional
6	Subject	The norm subject (the person or persons to whom the norm is addressed)	Primary, obligatory
7	Legal modality	The legal modality (ought, ought not, may, or can)	Primary, obligatory
8	Act identifier	The act identifier (used as a reference to a separate act description)	Primary, obligatory

the second and the seventh slots provides us with a complete picture of the legal modality. The legal modality extends over the description of the act. The length slot holds a reference to a separate act description. The act description is the topic of the next section.

4.2. THE ACT FRAME

Every action has many different aspects. To mention the most obvious ones: every action is performed by someone, at a certain time and at a certain place. Since we deem an easily accessible representation of actions desirable, we have opted for the use of a separate structure for the representation of actions: the act frame (instead of using a single predicate).

Rescher (1967 1970; cf. Loth, 1988) has provided a description of aspects of action in which the miscellaneous aspects of action are taken into consideration. According to Rescher, an act has the following aspects (the sixth aspect has been added to Rescher's five aspects.

(1) An agent.
(2) An act type.
(3) A modality, divided into
  - a modality of means; and
  - a modality of manner.
(4) A setting, divided into
  - a temporal aspect;
  - a spatial aspect; and
  - a circumstantial aspect.

(5) A rationale, divided into
- a cause;
- an aim; and
- an intentionality.

(6) A final state.

The interpretation of the (primary) aspects of action is provided below, when discussing the act frame. To be able to refer to act descriptions, again some auxiliary elements are introduced: the act identifier, the promulgation and the scope. What has been said above about the auxiliary elements of a norm applies equally here. The six aspects of an action (and their subdivisions) can be grouped in an act frame (Table 2).

In the frame, the division between primary and auxiliary aspects is maintained. The first three slots of the act frame contain the auxiliary aspects of action "act identifier", "promulgation" and "scope", respectively. The fourth to the 14th slots contain the primary aspects of action. The fourth slot is reserved for the agent who performs the act (the agent and the norm subject are the same person). The fifth slot contains a (general) typification of the act. The sixth and the seventh slots are reserved for a description of the modality of action. The modality of means can be specified in the sixth slot, the modality of manner in the seventh slot. The eighth, ninth and tenth slots facilitate a specification of the setting or context of an act. They are used to specify the time, location and circumstances of an act respectively. The rationale of an act is specified in the 12th and 13th slots. The 11th slot can contain information about the cause of an act, the 12th slot contains the aim of an act and the 13th slot comprises the state of mind of the agent. Finally, the results and consequences of an act are represented in the 14th slot.

The 14 slots of the act frame provide an overview of the aspects actions can have. Not every aspect proves relevant in each case. If an aspect proves irrelevant, the slot can remain uninstantiated. However, instantiation of the agent slot and the act type slot is essential to the description of an action.

4.3. THE CONCEPT-DESCRIPTION FRAME

The description of acts and act frames has already brought us into the realm of concept descriptions: act descriptions are a specific form of concept descriptions. A concept description determines the meaning of a concept. In the legal domain, the concepts that need description are usually specific legal terms. The best known type of concept description has already been introduced in Section 2. This type is referred to as the (legal) definition.

A concept description comprises seven elements. The first element is the concept: every concept description frame describes one concept or term. It does so by either stating the *conditions* under which a concept is applicable or by naming some *instances* of a concept (combinations are also possible). An important element of the concept description is the concept type. Four concept types can be distinguished: legal definitions, deeming provisions, factors and meta-concepts. Deeming provisions are used to introduce legal fictions. Deeming provisions allow things which are not true to be treated as if they were, and things which are true to be treated as if they were not. The concept type

Table 2
An act frame

	Aspect	Typification	Status
1	Act identifier	The act identifier (used as a point of reference for the act)	Auxiliary, obligatory
2	Promulgation	The promulgation (the source of the description)	Auxiliary, obligatory
3	Scope	The scope (the range of application of the act description)	Auxiliary, obligatory
4	Agent	The agent (an individual, a set of individuals, an aggregate or a conglomerate)	Primary, obligatory
5	Act type	The act type (both basic acts and specified elsewhere can be used)	Primary, obligatory
6	Means	The modality of means (material objects used in the act or more specific descriptions of the act)	Primary, optional
7	Manner	The modality of manner (the way in which the act has been performed)	Primary, optional
8	Temporal aspects	The temporal aspects (an absolute time specification)	Primary, optional
9	Spatial aspects	The spatial aspects (a specification of the location where the act takes place)	Primary, optional
10	Circumstances	The circumstantial aspects (a description of the circumstances under which the act takes place)	Primary, optional
11	Cause	The cause for the action (a specification of the reason(s) to perform an action)	Primary, optional
12	Aim	The aim of an action (the goal visualized by the agent)	Primary, optional
13	Intentionality	The intentionality of an action (the state of mind of the agent)	Primary, optional
14	Final state	The final state (the results and consequence of an action)	Primary, optional

'factor' is used when we are not dealing with necessary and sufficient conditions. A factor merely contributes to the applicability of a concept; it assigns a weight factor to properties that play a role in the determination of the meaning of a concept. Factors can be positive or negative. A positive factor increases the likelihood of applicability of a concept, a negative factor decreases the likelihood of applicability. Finally, meta-concepts are used to deal with textual constructions such as application provisions and (some types of) exceptions. Application provisions make provisions operative, or render others inoperative. Again, promulgation and scope are introduced as auxiliary elements (no identifier is needed, since the concept itself serves as an identifier). The elements of a concept description can be moulded into a concept frame. The result is displayed in Table 3.

TABLE 3
A concept frame

	Element	Typification	Status
1	Concept	The concept to be described	Auxiliary, obligatory
2	Concept type	The concept type (definitions, deeming provisions, factor or meta)	Primary, obligatory
3	Priority	The weight assigned to a factor (only relevant when we deal with the concept type "factor")	Primary, optional
4	Promulgation	The promulgation (the source of the concept description)	Auxiliary, obligatory
5	Scope	The scope (the range of application of the concept description)	Auxiliary, obligatory
6	Conditions	The conditions under which a concept is applicable	Primary, it is obligatory to instantiate at least slot 6 or slot 7
7	Instances	An enumeration of instances of the concept	

The first slot of a concept frame is reserved for the concept that is described. The second slot contains the concept type or, in other words, the type of description that is provided of the concept. The third slot is only relevant when we are dealing with a concept description of the factor type. This slot holds the appropriate weight of the properties (conditions) contained in the sixth slot. The fourth and fifth slots accommodate the auxiliary elements promulgation and scope, respectively. The sixth slot contains the conditions under which a concept is applicable. Finally, the seventh slot can be used to represent instances of the concept.

4.4. THE VOCABULARY

Of course, the three frame structures need to be instantiated in order to create a model. To this end a vocabulary is needed. Within the context of the frame-based ontology, different vocabularies, ranging from natural language to formal notation methods, can be used to instantiate the frame structures. The selection of a vocabulary depends on the aim of the modelling exercise; an analysis of a domain poses different demands than a formal model that is used as the point of departure for the implementation of a knowledge system. Note that the frame-based ontology does not commit to a particular modelling language; different languages can be selected and used within the context of our approach. For the conceptual modelling phase, we have opted for (structured) natural language (in the formal modelling phase we opt for PROLOG, identifying the predicate relations using the KANT method). We distinguish 11 categories within the natural language vocabulary: (1) words denoting actions; (2) words denoting agents; (3) words denoting objects; (4) words expressing relations; (5) words assigning properties to other entities; (6) words indicating time; (7) words indicating place; (8) words indicating source; (9) words marking textual constructions; (10) words marking arithmetical operations; and (11) words indicating the legal modality.

The division into categories is partly based on the structure of the three frame types. The categories acts, agents, time, place, source and legal modality are directly traced back to the frames. To give an example: the act-type slot of an act frame can only contain expression from the acts category of the vocabulary. However, some slots (e.g. the circumstance slot of the act frame) can contain entries from nearly all categories of the vocabulary. The distinction between objects, relations and properties is partially based on the need to distinguish the six aforementioned categories; these categories are negatively defined as the categories that contain entities unsuited for any of the other categories. The objects category comprises material entities (as opposed to human entities; agents). The relations category contains words that express the status of objects or agents in their relation to other objects or agents. The properties category is constituted by words that ascribe certain qualities to objects or agents. The categories textual constructions and arithmetics are reserved for words expressing textual constructions (e.g. exceptions) and mathematical operations operations (e.g. adding or subtracting), respectively. Note that categories such as acts and agents also contain relations or properties. The act of "hitting someone" could be represented as "person *p1* hits person *p2*", and this clearly expresses a relation. Another example: if we state that a person *p* is a fisherman, this qualifies both for the properties and the agents category (the agents category is selected). The time category also contains relations (e.g. "later" expresses a relation between time points). In fact, it is true that every category comprises relations in the broadest sense of the word. The content or meaning of the relation determines its category.

5. Analysis phase

The analysis phase is meant to outline the competence of the LKS. In our example, the domain-identification step [step 1(a)] yields the articles and allowances of the ICLR (Jones & Sergot, 1992).

```
art 1. A separate form must be completed by the borrower for each volume
       borrowed.
art 2. Books should be returned by the date due.
art 3. Borrowers must not exceed their allowances of books on loan at any one
       time.
art 4. No book will be issued for borrowers who have books overdue for return
       to the library.
art 5. Article 3 and article 4 do not apply if written permission has been
       obtained from the Dean.

Book allowance: undergraduates:  6
                post graduates: 10
                academic staff: 20
```

We have selected the ICLR as an example because of its status as a benchmark problem. We have extended the original ICLR example by adding a fifth article to illustrate our approach given an applicability restriction.

In the ICLR example many different tasks can be performed. It is important to make the task explicit since this greatly determines how the domain has to be modelled. For instance, the task of listing all norms in the ICLR does not require a very sophisticated domain model but if we are to determine who should undertake what series of action to satisfy all ICLR norms, we obviously need a more elaborated domain model. We here choose the task of assessing whether in a given case description any norm of the ICLR are breached (and by whom). The case description, being the input of the task, is assumed to be expressed in phrases such as: "university status S of person P", "person P has borrowed book/volume B", "person P has book/volume B overdue;† "person P has completed a form for book/volume B", and "the librarian lends a book/volume to P" (later on, these phrases will have to be stated more formally — see Section 6). We assume that there is no distinction between books and volumes. The output of the task is a list of tuples of norms and agents specifying the norms that are breached and by which agents.

6. Conceptual modelling phase

In the first step of the conceptual modelling phase, we provide an informal description of the method with which the system will perform its task [step 2(a)]. Our task "assessment of breach" evaluates a case in retrospect. We here confine ourselves by stating that the control of the task is an iteration of the following three steps: (1) give an "institutional view" on the case description by applying the information contained in concept-description frames, (2) determine in this institutional view which "institutional acts" have been performed and (3) determine whether the performance of these acts results in the breach of a norm. These three steps are performed for each time interval in the case of description (see Section 7) and repeated until no more conclusions can be reached. The task returns a list of breached norms. More details on the description of the method can be found in Visser (1995).

The second step in the conceptual modelling phase involves the selection (and adaptation) of a domain ontology [step 2(b)]. As stated previously, we have selected the frame-based ontology described in Section 4. We will use a natural-language vocabulary to instantiate the (conceptual) frame structures.

The third step in the conceptual modelling phase is the acquisition and modelling of domain knowledge. [step 2(c)]. In essence, this step involves the creation of the (conceptual) frame-based model by filling in the frame structures. The language that is used to fill the structures can be characterized as 'structured English'. It contains means to represent textual constructions (e.g. references, rule-exception structures and application provisions), means to represent the norm promulgation, means to typify the legal modality, etc. In this article, we do not elaborate on the conceptual language (see Van Kralingen, 1995).

We have developed seven heuristics to guide the process of assembling a conceptual frame-based model. We briefly discuss the two core heuristics. The first core heuristic reads: start at the core of a norm, act or concept description. This heuristic aims at finding an appropriate starting point for the modelling process. The second heuristic

† Note that the ICLR does not contain information on loan periods, so we cannot *calculate* which books are overdue.

governs the extension of the model. It reads: a new provision should be added an existing frame if and only if adding the provision does not result in changes to more than one slot of the frame to which the provision is added (for the application of this heuristic the norm-identifier slot and the norm-promulgation slot are not taken into account since they are merely used as a means of referring to a norm frame and a means of representing the norm's promulgation, respectively). The rationale behind the heuristic is a representation in a minimal number of frames while preserving the original meaning of the regulation represented. The other five heuristics facilitate the separation of different norm and concept types and control the complexity of the model (e.g. by avoiding complex nested structures). An extensive description of the heuristics is found in Van Kralingen (1995).

Applying the first heuristic to the ICLR yields the following (conceptual) norm frame.

```
(1) norm identifier        ‘‘norm-1’’
    norm type:             Norm of conduct
    promulgation           ICLR article 1
    scope                  ICLR
    conditions of ap.      Subject wants to borrow a book.
    subject                Borrower
    legal modality         Ought to
    act identifier         ‘‘complete-form’’
```

If we consider the second article of the ICLR we find that the second core heuristic prevents the article from being added to norm frame (1) since adding the article would result in changes to more than one slot, namely the conditions-of-application slot and the act-identifier slot. Consequently, a second frame is created:

```
(2) norm identifier        ‘‘norm-2’’
    norm type:             Norm of conduct
    promulgation           ICLR article 2
    scope                  ICLR
    conditions of ap.      Subject has borrowed a book.
    subject                Borrower
    legal modality         Ought to
    act identifier         ‘‘return-book-by-date-due’’
```

The act-identifier slot makes reference to the act "return-book-by-date-due". This act can be described in an act frame. Note that not all different aspects are relevant to the situation at hand, and, hence, they are not addressed in the act frame.

```
    act identifier         ‘‘return-book-by-date-due’’
    promulgation:          ICLR article 2
    scope:                 ICLR
    agent:                 Borrower
    act type:              Return
    temporal aspects:      Book should be returned by the date due.
    circumstances:         A book has been borrowed.
```

The third article presents us with an interesting interpretation issue. It can be argued that two norms can be read from the article: one norm forbidding a borrower to exceed his allowance, and one norm forbidding the librarian to issue a book if a borrower has reached his allowance. We can represent both interpretations in separate norm frames.

```
(3a) norm identifier         ''norm-3''
     norm type:              Norm of conduct
     promulgation            ICLR article 3
     scope                   ICLR
     subject                 Borrower
     legal modality          Ought to
     act identifier          ''complete-form''

(3b) norm identifier         ''norm-3b''
     norm type:              Norm of conduct
     promulgation            ICLR article 3
     scope                   ICLR
     conditions of ap.:      Borrower has reached book allowance.
     subject                 Librarian
     legal modality          Ought not
     act identifier          ''issue-book''
```

In fact, the phenomenon that one article comprises more than one norm is not uncommon (e.g. Hart, 1961; Kelsen, 1991). For instance, in penal law, we often find provisions stating that a person will be punished if he performs a certain action. Such a provision can be interpreted as both a norm of conduct (a prohibition to perform a certain action) and a norm of competence (conferring a power onto an official to administer a certain sanction). Note that, while norm (3a) does not have any conditions of application, norm (3b) does.

Modelling article 4 is straightforward, and for the resulting norm and act frames, we refer to the appendix. Article 5 of the ICLR, on the other hand, provides us with an interesting problem. The article restricts application of the articles 3 and 4 of the ICLR under the condition that written permission has been obtained from the Dean. It can be modelled as two concept descriptions of the "meta type". Below, we provide one of the concept-description frames.

```
     concept                 ''not-applicable-art3''
     concept type:           meta
     promulgation            ICLR article 5
     scope                   ICLR
     conditions:             written permission has been obtained from the
                             Dean
```

A complete conceptual model of the ICLR is presented in the appendix. Below, we elaborate further on the model, as we discuss the next phase, the formal modelling phase.

7. Formal modelling phase

The first step in the formal modelling phase concerns defining more precisely the boundaries of control knowledge and domain knowledge, the two major types of knowledge in the expertise model [step 3(a)]. This step is necessary because some legal sources contain procedural aspects. Intuitively, legal sources are modelled as domain knowledge, but if they contain procedural aspects it is not clear whether they should be modelled as CommonKADS domain or control knowledge. In such cases, an explicit decision has to be made as to how such sources will be modelled. Visser (1995) distinguishes two forms of procedural knowledge in legal sources: meta-level procedural norms of competence, and conflict-resolution knowledge.

In the ICLR there are no meta-level procedural norms of competence. That is, there are no procedural norms of competence that express how other norms should be applied. Consequently, we do not have to decide how to model this form of control knowledge for our example system. The situation is different with conflict-resolution knowledge. The added fifth article of the ICLR states applicability restrictions and resulted in two concept-description frames of type meta. There are three ways to deal with the applicability restrictions (cf. Visser, 1995, p. 89).

(1) Compile out the applicability restrictions by adding extra conditions to the frames representing articles 3 and 4.
(2) Let the inference engine automatically withhold conclusions based on articles 3 and 4 in case article 5 applies.
(3) Perform explicit meta-level reasoning about whether articles 3 and 4 should be applied.

Which choice is made depends on the requirements of the LKS under development (e.g. explanation and tracing facilities, isomorphic modelling). Here, we will choose for the first option; we modify the norm frames (3a) (3b) and (4) by adding extra conditions making the norms inapplicable in case of a written permission by the Dean. Apart from the applicability restrictions, there are no other occurrences of conflict-resolution knowledge in the ICLR (a librarian who issues a book where this is not allowed is considered to breach a norm rather than cause a conflict).

The second step in the formal modelling phase [step 3(b)] involves defining the control knowledge. In Section 6 we stated that the control knowledge of our task consists of three successive steps that are performed in each time interval of the case at hand. Before we address the control knowledge in more detail, we elaborate on the case description and its time intervals. A case description constitutes the input for the LKS and is structured as a series of time intervals. Each interval has an associated set of facts (called *physical facts*) that are assumed to be true during that interval. Together the intervals describe what we refer to as the *physical world* (or, indeed, a relevant portion of it). The task of the LKS is to create an *institutional view* on this description of the physical world. In solving the task, the LKS draws (institutional) conclusions from the (physical) facts in the interval (possibly using facts from previous intervals as well). The LKS thus creates a network of conclusion sets—called an *argument network*—for each interval. Conclusions in the nodes of each argument network describe what institutional concepts are applicable in the interval (based on concept description frames); what institutional acts are performed (based on event and process frames) and what norms are breached (based on norm frames). The latter type of conclusions form the output of our task. In Figure 1 we present an example of a physical case description (here consisting of four intervals) and its argument networks. The thick arrows in the figure indicate succession of intervals, the thin arrows indicate institutional conclusions and the dotted arrows indicate institutional processes that are found to take place during successive intervals (e.g. the dotted line between intervals [$t3$, $t4$] and [$t4$, $t5$] could mean that an institutional process is taking place from $t3$ until $t5$).

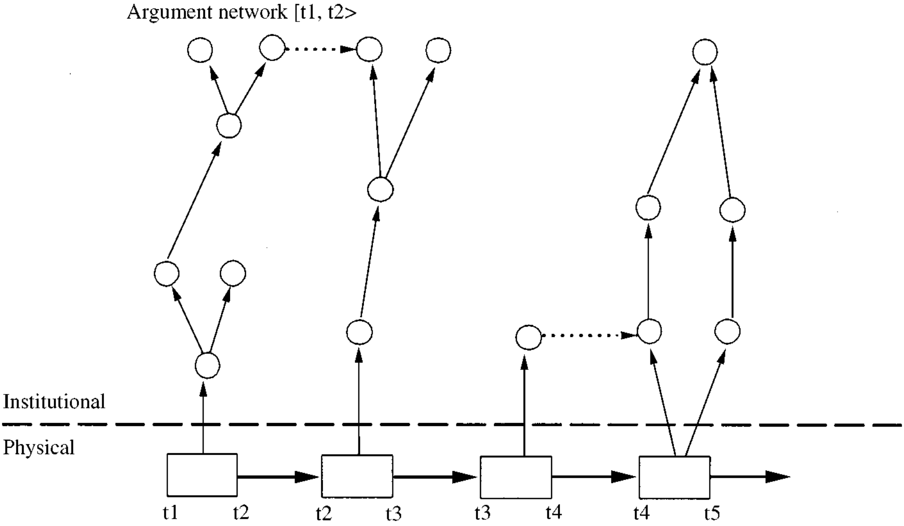

FIGURE 1. Physical case description and Institutional conclusions.

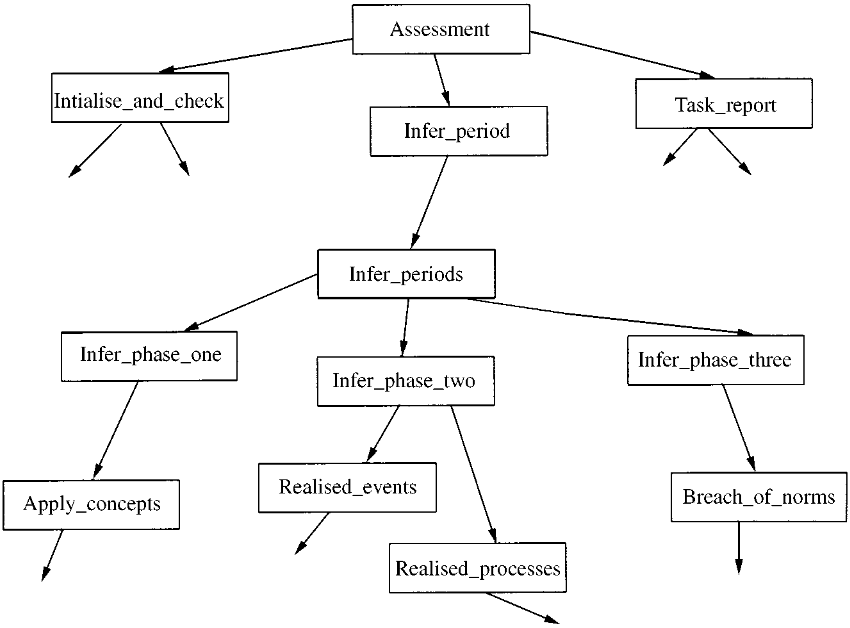

FIGURE 2. Partial task structure for the assessment task.

CommonKADS provides a library of generic problem-solving tasks which can be selected according to the type of problems that need to be solved (Breuker & Van de Velde, 1994). Such a generic task description specifies a decomposition of the main task in a set of more primitive tasks. Here, we do not address the selection process of such a generic task, nor do we discuss the task decomposition of our task in detail. We confine ourselves to presenting a fragment of the task structure in Figure 2.

The formal specification of each task consists of a task definition and a task body (viz. the CommonKADS method). The task definition states the goal of the task and the input and output. The task body states how the task is achieved. Its most important part is the task control structure, which specifies how the output is derived from input. Below we give an example of a task specification for the assessment task under consideration. The task (apply_concepts; see above) controls the application of successive concept frames while extending the *argument network*. In the description of the input and output (roles), we list the names of the variables used in the control structure. The language used to describe the control structure of the task is PROLOG.

```
task: apply_concepts
task definition
  goal:         Extending an argument network in one node with all
                conclusions that can be derived given a list of concept
                frames.
  input roles:Argument Network: $\Omega$,
                List of concept frames: $\{c_1, \ldots, c_n\} \ll$[Frame|RestFrame].
                Period: $[t_1, t_2 > \ll$Period,
                Scenario: $sn \ll$Scenario,
                Open node: $v \ll$OpenNode,
                Initial number of successfully applied concept frames:
                $n \ll$ln, $N$.
  output
  roles:        Extended argument network: $\Omega$,
                New open nodes: $\{v_1, \ldots, v_n\}$,
                Total number of successfully applied concept frames:
                $n \ll$Out, $N$.
task body
  task type:    composite
  decomposi-
  tion:         (apply_concepts/6), apply_concept_frame/4.
task control structure:
                apply_concepts([ ], _, _, _, N, N).
                apply_concepts([Frame|RestFrames], Period, Scenario,
                OpenNode, In, Out):-
                       (
                                 (
                                       apply_concept_frame(Period,
                                       Scenario, OpenNode, Frame),
                                       NewIn is In + 1
                                 ):    NewIn is In
                       ),
                apply_concepts(RestFrames, Period, Scenario, OpenNode,
                In, NewIn).
```

A detailed description of the assessment task control knowledge as well as the precise meaning of the task-specification slots can be found in Visser (1995).

The third step in the formal modelling phase [(step 3(c)] is the creation of the statute-specific ontology. This is done by applying the KANT method, the first step of which is the creation of a TOO structure. For the ICLR, the TOO structure reads:

borrower completes form (for volume)
borrower borrows *volume*
book has a *date due*
borrower has *allowance*
  (of books on loan)
borrower has *book overdue*
  (for return to the library
borrower has *status*

dean gave written permission
librarian issues *book*
undergraduate has *allowance*
postgraduate has *allowance*
academic staff has *allowance*

In our domain ontology, all acts are assumed to be performed by a human actor. We have introduced the notion of a librarian to be able to represent that a book can be issued to a borrower.

The next step in KANT method is the creation of the EAV structure, in which the entities are given attributes and the potential values of these attributes are identified. The EAV structure reads (values marked with an asterisk may have multiple instantiations):

entity	*attribute*	*value(s)*
borrower	completed-form	book*
borrower	borrowed	book*
book	date-due	date
borrower	allowance	integer
borrower	has-written-permission	yes/no
borrower	has-book-overdue	book*
borrower	has-status	{undergraduate, postgraduate, academic staff}
librarian	issues	{book, borrower}*
dean	gave-permission	borrower
undergraduate	has-allowance	integer
postgraduate	has-allowance	integer
academic staff	has-allowance	integer

In the creation of the EAV structure, we have changed the tense of some of the attributes to obtain a more uniform terminology (e.g. completed-form). Note again that we interpret the ICLR such that books and volumes are the same.

For the creation of a class hierarchy, we regroup the entities in the EAV structure and introduce some abstract entities. We decide that we do not need separate entries for both "borrower has written permission from Dean" and "Dean gave permission to borrower". Maintaining the entry for the borrower suffices (and eliminates a separate entity 'Dean', which can be seen as outside the system). The class hierarchy for the ICLR is given in Figure 3. Classes lower in the hierarchy inherit the attributes of their parents. Note that we introduced the top-level class *Thing* and the class *Person* (which is given a name and address as its attributes).

The class hierarchy is assumed to distinguish all relevant entities in the domain. For this reason, we use it as the basis for choosing predicate names. This is a process guided by heuristics. Briefly stated, a predicate *attribute-id*(*Class-id*, *Attribute-value*) is defined for each entry *Class-id*(*attribute-id*) in the class hierarchy. For instance, the predicate *name*(*Person*, *Name*) corresponds to the entry *Person*(*name*) in the class hierarchy. Below, we list the predicate list.

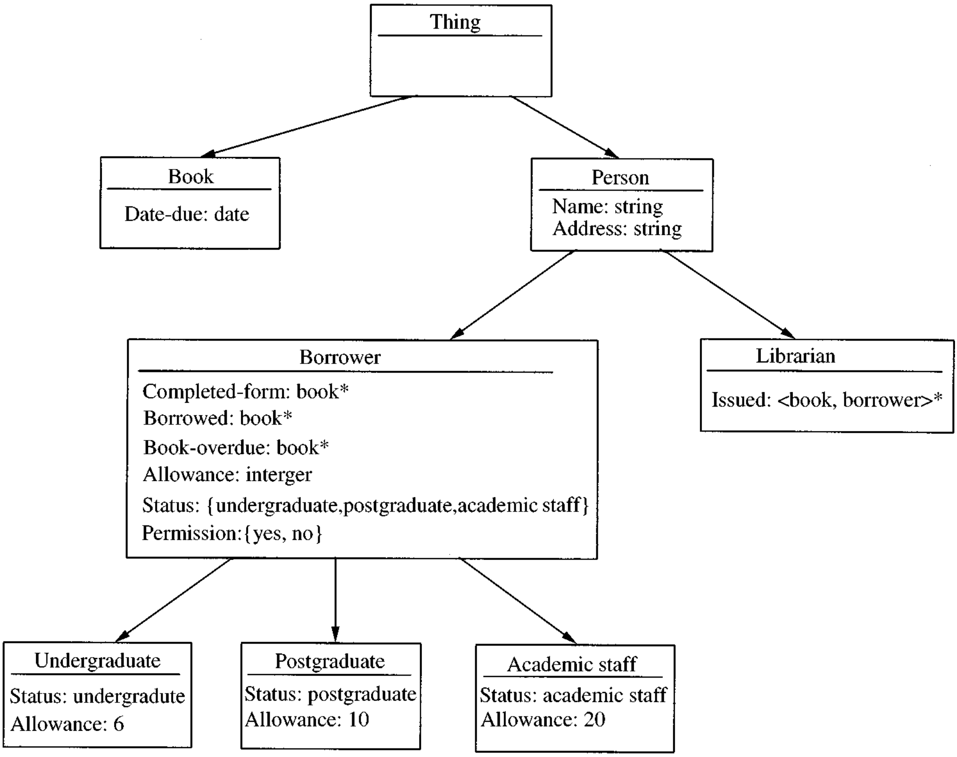

FIGURE 3. Class hierarchy for the ICLR.

```
date_due(Book, Date)              status(Borrower, Status)
name(Person, Name)                written_permission(Borrower)
address(Person, Address)          allowance(Status, Allowance)
issued(Librarian, Book, Borrower) undergraduate(Person)
completed_form(Borrower, Book)    postgraduate(Person)
borrowed(Borrower, Book)          academic_staff(Person)
allowance(Borrower, Allowance)    book_overdue(Borrower, Book)
```

Predicates such as *borrowed*(*Academic-Staff, Book*) are left out since such predicates are effectively subsumed under the predicate *borrowed*(*Borrower*, *Book*). For the same reason, we have left out predicates such as *name*(*Postgraduate*, *Name*).

The next step in the formal modelling phase [step 3(d)] is the formalization of the domain knowledge. Before we can formalize the domain knowledge, we need to define formal versions of the frame structures themselves. In fact, this can be done in many different ways, depending on the implementation language used, the domain model requirements and so on. In this article, we assume that we have formalized versions of the frame structures available (not of their contents). For a more detailed description of the formalization process, including a discussion of the differences between the conceptual and the formal frame structures, we refer to Visser (1995) and Visser and Bench-Capon (1996). The next step is to express the knowledge in the conceptual frame structures in terms (a) of the general legal ontology and (b) the statue-specific ontology. The latter ontology comprises the predicate relations that we just determined by applying the KANT method. The former ontology consists of the frame structures and a limited set of generic meta predicates, the latter which will be detailed below.

One of the differences between the conceptual and the formal frame structures is that in the formal norm frame, the conditions of applications slot has been split up into an object(-level) conditions slot (used to state conditions about the outside world) and a meta-level conditions slot [used to state (meta-level) conditions about the conclusions drawn from other frame structures]. The reason for this split is that the two types of conditions are used differently in the reasoning process. For the condition slots, a special set of reserved (meta) predicates is defined, the most important of which are: *breached*(*Person*, *Norm*) to state that a norm has been breached, *arithmetic*(*Expression*) to express necessary calculations, *true_from* (*T*. *Clause*) to state a clause is true at and after a certain point in time *realized*(*Agent*, *Event*) to state that an agent has realized an event, *function*(*FunctionCall*) to refer to an externally defined function, *occurs*(*Agent*, *Process*, *T*-*begin*, *T*-*end*) to state that an agent is involved in realizing a process between two points of time, *capable*(*Agent*, *Act*) to state that an agent is capable of performing an act, *effectuate*(*Person*, *Modality*, *Norm*, *Act*) to state that a person ought (not) to do an act according to a particular norm and the predicates *always_false* and *always_true* which effectively are a contradiction, and a tautology. Also, we use special predicates to refer to act frames (these predicates are not discussed here).

Below, we list the formal version of norms (2) and (3b). The time references are used to link predicates—and thus conditions–onto states (it is assumed that the case description consists of a chain of states and acts). Note that we have added an extra condition in the second norm frame.

```
norm identifier:      norm_2
norm type:            conduct
promulgation:         {iclr_art_2)}
scope:                {iclr}
time reference:       Today
object conditions:    true_from(Today, borrowed(Borrower, Book)) and
                      true_from(Today, date_due(Book, Date_due)) and
                      arithmetic(Today > = Date_Due)
meta conditions:      always_true
subject:              Borrower
legal modality:       ought_to
act reference         return_book(Borrower, Book, Date_due)

norm identifier       norm_3b
norm type:            conduct
promulgation:         {(iclr_art_3)}
scope:                {iclr}
time reference:       Today
object conditions:    true_from(Today, completed_form(Borrow, Book))
                      and true_from(Today, status(Borrower, Status))
                      and true_from(Today, allowance(Status, Allow-
                      ance)) and function(number_of_books_borrowed
                      (Borrower, Today, Number)) and arithmetic-
                      (Number < Allowance) and not(true_from(Today,
                      written_permission(Borrower)))
```

```
meta conditions:        always_true
subject:                Librarian
legal modality:         ought_not
art reference:          issue_book(Librarian, Borrower, Book)
```

In contrast to the conceptual model, the formal model has separate frames for *events* (acts that occur instantaneously and *processes* (acts that have a known and fixed duration). Also, a distinction is made between acts that occur in the world (e.g. *a* kills *b*), referred to as *physical acts*, and acts that are legal interpretations of acts that occur in the world (e.g. *a* murders *b*, or *a* manslaughters *b*), referred to as *institutional acts*. Below, we present the physical event of issuing a book.

```
event identifier:         issue_book
act:                      issue_book(Librarian, Borrower, Book)
promulgation:             {(iclr_art_4)}
scope:                    {iclr}
agent:                    Librarian
act type:                 physical
temporal setting:         always_true
spatial setting:          always_true
circumstantial setting:   true_from(Before, not(borrowed(Borrower,
                          Book)))
time reference:           Before, After
Initial state:            {completed_form(Borrower, Book)
                          not(borrowed(Borrower, Book))}
Final state:              {not(completed_form(Borrower, Book))
                          borrowed(Borrower, Book)}
```

Note that part of the event specification is an initial state and a final state. This provision has been introduced to accommodate planning tasks (see Visser, 1995). The set of clauses in the initial state are true in the state before the event takes place (tagged *Before* in the event frame) and the set of clauses in the final state are true in the state after the event (tagged *After* in the event frame). This idea allows us to use the slots as so-called add and delete lists in STRIPS-style planning systems (Fikes & Nilsson, 1971). By including the clauses "completed_form (Borrower, Book)" and "not(completed_form(Borrower, Book))" in the initial and final state, respectively, we are able to withdraw form so that the book can be issued again (of course, the book has to be returned before it can be reissued).

In the ICLR, there are no definitions as there are no (institutional) concepts introduced. For that reason, we give an example of an imaginary formal concept(-description) frame. In the frame, the concept "borrowed" is defined as "registered possession of library book". Note that since the terms possession and registered are not mentioned in the ICLR, our KANT method has not given us predicate names for these terms. Here, we simply assume that the predicates "registered (Person, Book)" and "possession(Person, Book)" are available. The promulgation of the definition here is not the ICLR but the knowledge engineer that specified the rule.

```
concept identifier:       borrowed
concept:                  borrowed (Person, Book)
concept type:             definition
priority:                 { }
```

```
time reference:          T
promulgation:            {(knowledge-engineer)}
scope:                   {iclr}
conditions:              true_from(T, registered(Person, Book)) and
                         true_from(T, possession(Person, Book))
instances:               always_false
```

The last step in the formal modelling phase is the definition of inferences (not shown here). Inferences link the tasks knowledge onto the domain knowledge (viz. the filled-in frame structures). They define how, for instance, the object conditions and the meta-level conditions slots are evaluated (in case of norm frames) and how initial state is transformed into the final state (in case of the act frames).

The formal model can be used as the basis for an implementation. In this article, we have chosen to describe the formal model in a PROLOG-style language (which eases the implementation of the formal model in PROLOG), but other languages could have been chosen as well.

8. Conclusion

In this article, we have illustrated the applicability of the method with the help of a small benchmark problem. In Van Kralingen (1995) and Visser (1995) several steps from the method presented here have been applied to a substantial fragment of the Dutch Unemployment Benefits Act. This has resulted in a prototype system called FRAMER (which has been implemented in PROLOG). In several smaller research projects, the conceptual ontology has been used in diverse domains, such as penal law, administrative law and civil law. Its applicability has also been shown by Voermans (1995). In 1999, the Dutch Internal Revenue Service has adopted the frame-based ontology in the POWER project; a project that investigates the drafting and implementation of legislation in legal knowledge systems.” We summarize our main findings.

- Legal knowledge systems often have an implicit conceptualization. The use of ontologies to make conceptualizations allows us to compare and analyse—and thus to assess the merits—of different conceptualization.
- Ontologies are a useful instrument during the construction of a legal knowledge system, in particular, for knowledge acquisition.
- Without extension, CommonKADS provides little support for the specification of *legal* domain knowledge
- Extending CommonKADS with ontologies of the legal domain makes the method more suitable for building LKS.
- The distinction between a statue-specific ontology and a generic legal ontology proves useful. Both ontologies are necessary but only the generic ontology is reusable.
- The method presented here provides a guided way of bridging the gap between legal knowledge and an operational prototype of an LKS.
- The method presented here can be used to create libraries of reusable problem-solving methods, domain ontologies and domain models.
- To be an adequate benchmark problem the ICLR should (at least) be extended with an applicability restriction and a term definition.

The research presented in this article has partly been carried out at the Department of Law and Computer Science of the University of Leiden (The Netherlands). The study at the University of Leiden was supported by a grant from the Foundation for Law and Public Administration (REOB) which is part of the Netherlands Oganzation for Scientific Research (NWO).

References

AARNIO, A. (1987). *The Rational as Reasonable*; *A Treatise on Legal Justification*. Dordrecht, The Netherlands: D. Reidel.

BENCH-CAPON, T. J. M. (1991). *Knowledge-based Systems and Legal Applications*, APIC Series No. 36. London,, UK: Academic Press.

BENCH-CAPON, T. J. M. & COENEN, F. P. (1992). Isomorphism and legal knowledge based systems. *Artificial Intelligence and Law*, **1,** 65–86.

BENCH-CAPON, T. J. M. ROBINSON, G. O., ROUTEN, T. W. & SERGOT, M. J. (1987). Logic programming for large scale applications in law: a formalisation of supplementary benefit legislation. *Proceedings of the 1st International Conference on Artificial Intelligence and Law*, pp. 190–198. Boston, MA, USA.

BREUKER, J. A. & VAN DE VELDE, W. (1994). In J. A. BREUKER & W. VAN DE VELDE, *CommonKADS Library for Expertise Modelling*, *Reusable Problem Solving Components*, Amsterdam, The Netherlands: IOS Press.

BROUWER, P. W. (1990). *Samenhang in recht*; *een analytische studie*. Rechtswetenschappelijke Reeks. Groningen, The Netherlands: Wolters-Noordhoff.

FIKES, R. E. & NILSSON, N. J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, **2,** 198–208.

GARDNER, K. M. & SPELMAN, K. C. (1993). The use of KADS to model an intellectual property legal practice: a case study. *Proceedings Kennistechnologie '93*, pp. 439–443. Amsterdam, The Netherlands.

HAGE, J. C., LEENES, R. & LODDER, A. R. (1994). Hard cases: a procedural approach. *Artificial Intelligence and Law*, **2,** 113–167.

HART, H. L. A. (1961). *The concept of Law*, Clarendon Law Series. Oxford, UK: Oxford University Press.

JONES, A. J. I. & SERGOT, M. J. (1992). Deontic logic in the representation of law: towards a methodology. *Artificial Intelligence and Law*, **1,** 45–64.

KELSEN, H. (1978). *Pure Theory of Law*. Translation of "Reine Rechtslehre" (2nd revised edition, 1960) by Max Knight, First published in 1934. Berkeley, USA: University of California Press.

KELSEN, H. (1991). General theory of norms. *Translation of "Allgemeine Theorie der Normen"* Translate by Hartney) Oxford, UK: Clarendon Press.

KRALINGEN, R. W. VAN (1995). *Frame-Based Conceptual Models of Statute Law*, Computer/Law Series Vol. **16**. The Hague, The Netherlands: Kluwer Law International.

LOTH, M. A. (1988). *Handeling en aansprakelijkheid in het recht*. Arnhem, the Netherlands: Gouda Quint.

MACCORMICK, N. & WEINBERGER, O. (1986). *An Institutional Theory of Law*: *New Approaches to Legal Positivism*. Dordrecht, The Netherlands: D. Reidel.

RESCHER, N. (1967). The logic of decision and action. In *Aspects of Action*. Appendix II, pp. 215–219.

RESCHER, N. (1970). On the characterization of actions. In M. Brand, (Ed.). *The Nature of Human Action*, pp. 247–254. Pittsburgh, USA: University of Pittsburgh, Scott Foresman and Company.

ROSS, A. (1968). *Directives and norms*. New York, USA: Humanities Press.

RUITER, D. W. P. (1993). *Institutional Legal Facts*: *Legal Powers and their Effects*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

SEARLE, J. R. (1969). *Speech Acts*. Cambridge: Cambridge University Press (Dutch translation by F.H. van Eemeren, 1977, Utrecht, The Netherlands: Het Spectrum.

VISSER, P. R. S. (1995). *Knowledge Specification for Multiple Legal Tasks*; *A Case Study of the Interaction Problem in the Legal Domain*, Computer/Law Series, Vol. 17. The Hague, The Netherlands: Kluwer Law International.

VISSER, P. R. S. & BENCH-CAPON, T. J. M. (1996). The formal specification of a legal ontology. In: Van Kralingen *et al*., (Eds.), The formal specification of a legal ontology. *Proceedings of 9th International Conference on Legal Knowledge-Based Systems* (*JURIX'96*), pp. 15–24. Tilburg, The Netherlands.

VISSER, P. R. S., BENCH-CAPON, T. J. M. & VAN DEN HERIK, H. J. (1997). A method for conceptual legal domains: an example from the Dutch unemployment benefits act. *Artificial Intelligence and Law* (to appear).

VISSER, P. R. S. & WINKELS, R. G. F. (1997). *Proceedings of the* 1*st International Workshop on Legal Ontologies* (LEGONT'97). Melbourne, Australia.

VISSER, P. R. S., VAN KRALINGEN, R. W. & BENCH-CAPON, T. J. M. (1997). A method for the development of legal knowledge systems. *Proceedings of the 6th International Conference on Artificial Intelligence and Law*. pp. 151–160. Melbourne, Australia.

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).

WRIGHT, G. H. von (1963). *Norm and Action*; *a Logical Enquiry*, International Library of Philosophy and Scientific Method. London, UK: Routledge & Kegan Paul.

WRIGHT, G. H. von (1983). *Practical Reason. Philosophical Papers*, Vol. 1. Oxford: Basil Blackwell.

Appendix: a conceptual frame-based model of the ICLR

ICLR: Imperial College Library Regulations

1. A separate form must be completed by the borrower for each volume borrowed
2. Books should be returned by the date due.
3. Borrowers must not exceed their allowances of books on loan at any one time.
4. No book will be issued for borrowers who have books overdue for return to the library.
5. Rule no. 3 and rule no. 4 do not apply if written permission has been obtained from the Dean.

```
Book allowances:  undergraduates:  6
                  post graduates: 10
                  academic staff: 20
```

Norm frames

```
(1)  norm identifier:     ''norm-1''
     norm type:           Norm of conduct
     promulgation:        ICLR article 1
     scope:               ICLR
     conditions of ap.:   Subject wants to borrow a book.
     subject:             person
     legal modality       Ought to
     act identifier       ''complete-a-form''

(2)  norm identifier:     ''norm-2''
     norm type:           Norm of conduct
     promulgation:        ICLR article 2
     scope:               ICLR
     conditions of ap.:   Subject has borrowed a book.
     subject:             Borrower
     legal modality       Ought to
     act identifier       ''return-book-by-date-due''
```

```
(3a) norm identifier:      ‘‘norm-3a’’
     norm type:            Norm of conduct
     promulgation:         ICLR article 3
     scope:                ICLR
     subject:              Borrower
     legal modality        Ought to
     act identifier        ‘‘exceed-allowance’’
(3b) norm identifier:      ‘‘norm-3b’’
     norm type:            Norm of conduct
     promulgation:         ICLR article 3
     scope:                ICLR
     conditions of ap.:    Borrower has been reached book allowance
     subject:              Librarian
     legal modality        Ought to
     act identifier        ‘‘issue book’’
(4)  norm identifier:      ‘‘norm-4’’
     norm type:            Norm of conduct
     promulgation:         ICLR article 4
     scope:                ICLR
     conditions of ap.:    Book is overdue.
     subject:              Librarian
     legal modality        Ought to
     act identifier        ‘‘issue-book’’
```

Act frames

```
(1)  act identifier:       ‘‘borrow-book’’
     promulgation:         ICLR article 1
     scope:                ICLR
     agent                 Borrower
     act type:             Borrow
     circumstances:        Borrow a book
(2)  act identifier:       ‘‘complete-form’’
     promulgation:         ICLR article 1
     scope:                ICLR
     agent                 Borrower
     act type:             Complete
     circumstances:        Borrow completes a form.
(3)  act identifier:       ‘‘return-book-by-date-due’’
     promulgation:         ICLR article 2
     scope:                ICLR
     agent                 Borrower
     act type:             Return
     temporal aspects:     Book should be returned by the date due.
     circumstances:        A book has been borrowed.
(4)  act identifier:       ‘‘exceed-allowance’’
     promulgation:         ICLR article 3
     scope:                ICLR
     agent                 Borrower
     act type:             Exceed
     circumstances:        Agent exceeds his book allowance
(5)  act identifier:       ‘‘issue-book’’
     promulgation:         ICLR article 4
     scope:                ICLR
     agent                 Librarian
```

```
    act type:            Issue
    circumstances:       Agent issues a book
```

Concept-description frames

```
(1) concept:             ‘‘book-allowance’’
    concept type:        Definition
    promulgation:        ICLR book allowances
    scope:               ICLR
    conditions:          The allowance for undergraduates is 6 books,
                         the allowance for post graduate is 10 books, and
                         the allowance for academic staff is 20 books.
(2) concept:             ‘‘exceed-the-allowance’’
    concept type:        Definition
    promulgation:        ICLR book allowances
    scope:               ICLR
    conditions:          The book allowance of a person exceeds his
                         ‘‘book-allowance’’.
(3) concept:             ‘‘not-applicable-art3’’
    concept type:        meta
    promulgation:        ICLR article 5
    scope:               ICLR
    conditions:          written permission has been obtained from
                         the Dean
(4) concept:             ‘‘not-applicable-art4’’
    concept type:        meta
    promulgation:        ICLR article 5
    scope:               ICLR
    conditions:          written permission has been obtained from
                         the Dean
```