Improving Legal Quality - an application report

Prof. dr. Tom M. van Engers
Belastingdienst/
Centrum voor Proces en Productontwikkeling
Herman Gorterstraat 5
Postbox 18280
3501 CG Utrecht
t.m.van.engers@acm.org

Drs. Margherita R. Boekenoogen
Belastingdienst/
Centrum voor Proces en Productontwikkeling
m.boekenoogen@planet.nl

ABSTRACT

Problems with legal quality will not only increase effort and costs of the law enforcement organisations, but also undermines the regulating power of the legislator. Unintended use or even abuse of the law may be the result. Governments therefore should improve their legal quality. The complexity of legislation however makes this task a hard one. The Dutch Tax and Customs Administration (DTCA in Dutch: Belastingdienst) has developed a method and supporting tools that support a systematic translation of (new) legislation into the DTCA's processes. This POWER-method and tools help to improve the quality of (new) legislation and codify the knowledge used in the translation processes in which legislation and regulations are transformed into procedures, computer programs and other designs. Thereby the time-to-market of the implementation of legislation will be reduced. In this article we explain some knowledge representation techniques that we use to improve legal quality. We will also show its application and give real-life examples of anomalies detected. In contrast to other knowledge modelling approaches the POWER-approach is focused on modelling legal sources rather than expert knowledge. Expert knowledge however is still needed to find the correct interpretations but also for efficiency reasons. Starting with representing the (legal) experts' knowledge (using scenarios) helps us to find the adequate scope (the legal sources to be analysed). Confronting the expert with differences between the model build out of the experts' knowledge and the ones we make out of the other knowledge sources (specifically the law) causes the legal experts to see things in a different light and has often led to changes in the law.

Categories and Subject Descriptors

Improving Legal Quality*; verification, formal modeling, UML/OCL, Knowledge-engineering, AI and Law.*

General Terms

Design, Reliability, Standardization, Languages, Theory, Legal Aspects, Verification.

ICAIL '03, June 24-28, 2003, Edinburgh, Scotland, UK.

Keywords

Legal Quality, Knowledge modeling, Knowledge engineering, Knowledge representation, Verification and validation, AI and Law.

1. INTRODUCTION

Getting the right knowledge at the right place at the right time is a critical success factor for the ability to effectuate the legislative power to regulate and control. In the POWER research program (Program for an Ontology-based Working Environment for Rules and regulations) run by the Dutch Tax and Customs Administration (DTCA) a method and supporting tools for the whole chain of processes from legislation drafting to executing the law by government employees and supporting citizens (see e.g. Van Engers et al., 1999, 2000, 2001) has been developed. Parts of this program, the E-POWER project, is subsidized by the European Commission through the IST 5th framework program.

The motive behind running the POWER program is that drafting and implementing new legislation is a rather time, energy and money consuming process consisting of many inter-connected processes. A large number of people is involved in these processes and the complexity of these processes make them very vulnerable to errors. Varying interests have to be aligned and communication difficulties due to differences in technical jargon have to be overcome in both drafting and implementing changes to legislation or even completely new legislation. The knowledge and experience needed to create new laws, specify, design and implement procedures and systems in legislative domains is very scarce. The law-enforcement organization not only has to adapt its procedures, processes and information systems because a new law has to be implemented. Also risk diagnosis, assessment procedures and audit measurements have to be designed and implemented as well. Needless to say that next to this, political and social-environmental requirements have to be taken into account. One of these requirements is the need for diminishing the administrative costs of business.

Between drafting new legislation and enforcement of this legislation a chain of processes has to be managed and aligned. Preventing errors as early as possible in this chain can save a lot of time and money. Not only at the design stage but even more during the law-enforcement stage. Unintended use or even worse abuse is often due to anomalies in the law. Also, the position of the government is much stronger when involved in a dispute if the law is very clear with respect to the object of disagreement.

Improving legal quality is one of the three main goals of the POWER research program. The other two goals are reduction of total cost of ownership (TTO) of the (knowledge-based) systems intended for the support of civil servants or of citizens and secondly, reduction of time to market (TTM) i.e. the speed with which these (knowledge-based) systems can be created.

The POWER-approach supports the finding of anomalies in legal sources. This article describes the different knowledge representation formats used in the POWER-approach. We will show how these knowledge representations are used and how they contribute to improving legal quality.

We will first discuss verification and validation in general. Next, we will present the two knowledge representation forms scenarios and Power conceptual models. Finally we present some results from a project that was aimed a improving the quality of legislation and the investigation of consequence of implementing a new law. We illustrate the results with scenario's and parts of POWER-conceptual models.

2. VERIFICATION AND VALIDATION

The quality of the law enforcement depends on the quality of the legislation itself and on the quality of the knowledge-based systems that are actually used in the client handling processes as well. Many approaches have been described that aim at improving the quality of legislation (see e.g. Voermans 2000 [22]) or improving the quality of knowledge bases used in knowledge-based systems (see e.g. Preece 1994 and Vanthienen 1997). The verification process as applied in the POWER-approach serves both perspectives. Spreeuwenberg et al. (Spreeuwenberg 2001) show how automated supported verification processes can be applied on formal legal models. VALENS for example is a verification tool that operates on a rule-based system (that could be an implementation of a formal legal model). The problem with verification procedures like the one supported by VALENS is that these procedures can only be applied after most of the hard work has already been done, i.e. a rule-based system or at least a knowledge component has been constructed. Especially if we want to give feedback to legislation drafters or policy makers, it is essential that before constructing a rule-based implementation of some regulations (or a piece of law) we could already find anomalies. Also, there may be no need at all to design a (rule-based) system. Once a conceptual model has been made, one can generate a knowledge component from it, see e.g. van Engers et al., 2001 [5].

We will show in this article that the POWER-approach is a powerful aid that helps to detect anomalies in legislation. During the modeling process, more defects are found than would have been found when applying automated verification (e.g. with Valens). Automated verification, combined with conceptual modeling, is still valuable because it might detect defects that have not been noticed during the modeling of the legislation. Automated verification is especially useful when applied to a knowledge-base that has received maintenance (Gerrits et al., 2000 [9]). In the same way, it could be applied to changed conceptual models due to changes in the legislation.

In the project we describe in this article, automated verification was not used, but as an illustration, we will indicate which of the examples of errors found are errors that can be found with a automated verification tool (like Valens for example) in a rule-based implementation.

3. SCENARIOS

Before making a formal representation of a certain domain it proves to be helpful to first understand a bit about the legal domain. This is best obtained by looking at how some (prototypical) cases are solved that correspond to a certain target group. The reasoning strategies of legal experts used for the solution of these (hypothetical or real) cases can be represented in a kind of procedural representation like a decision tree which can be expressed in e.g. UML Activity Diagrams. Within the POWER research program we use a special form of such activity diagrams which we call ‘scenarios’ because they represent the possible scenarios of solving cases (see also Van Engers et al. 2002 [7]).

The process of creating a scenario starts by asking the experts to explain how he or she applies the legislation (within the domain of interest) to a certain prototypical case. This reasoning strategy is then mapped onto a decision tree. The decision tree is subsequently elaborated until all cases within the range of interest are described by the decision tree. The join of all scenarios corresponding to solving a case using the legal source(s) forms the final scenario: a map of the legal domain expressed in a decision tree. The nodes of the map correspond to questions or decisions that follow from applying the legislation: a node typically contains a reference to a part of the legal source it is based on. A decision needs to be taken by a (certain) yes or no. Traversing the scenario a result or conclusion is reached. Tasks are represented by grouping questions concerning one issue on one diagram (or more if necessary) and naming the diagram accordingly to the issue at hand. Figure 1. presents an example of a part of a scenario that was made of a bill concerning subsidies for children’s day nursery.

Research shows that the scenarios are a rather effective representation if we want to communicate between knowledge modelers and legal experts (see Van Engers et al 2002 [7]). In our projects ran at the DTCA we experienced that developing such scenarios at the start of the knowledge modeling process helped both knowledge analysts and legal experts, especially in case of modeling new or complex legislation. Not only the scenario's create a quick and global overview of the legal domain at hand for analyst and expert, but also they can be used by different parties involves in designing and implementing legislation.

One of the things law enforcement agencies face when designing an implementation strategy is their risk assessment process. The diamonds in the scenarios represent a question or decision that has to be made when making a legal inference. The legal experts and the risk management analysts use these diamonds in the scenario diagrams to ask themselves what kind of data elements will be needed when taking such decisions and what alternatives exist for acquiring these data elements. Although many details still are lacking, one can for instance already use this information to start thinking about what data elements will have to be on the documents that need to be designed. One can also use it to check if the law to be implemented will or will not increase or diminish the administrative burden of the citizens involved. The questions or decisions in scenarios are kept as global as possible because scenarios are just intended to provide us with a quick overview of how a certain legal domain functions. If everyone involved in the

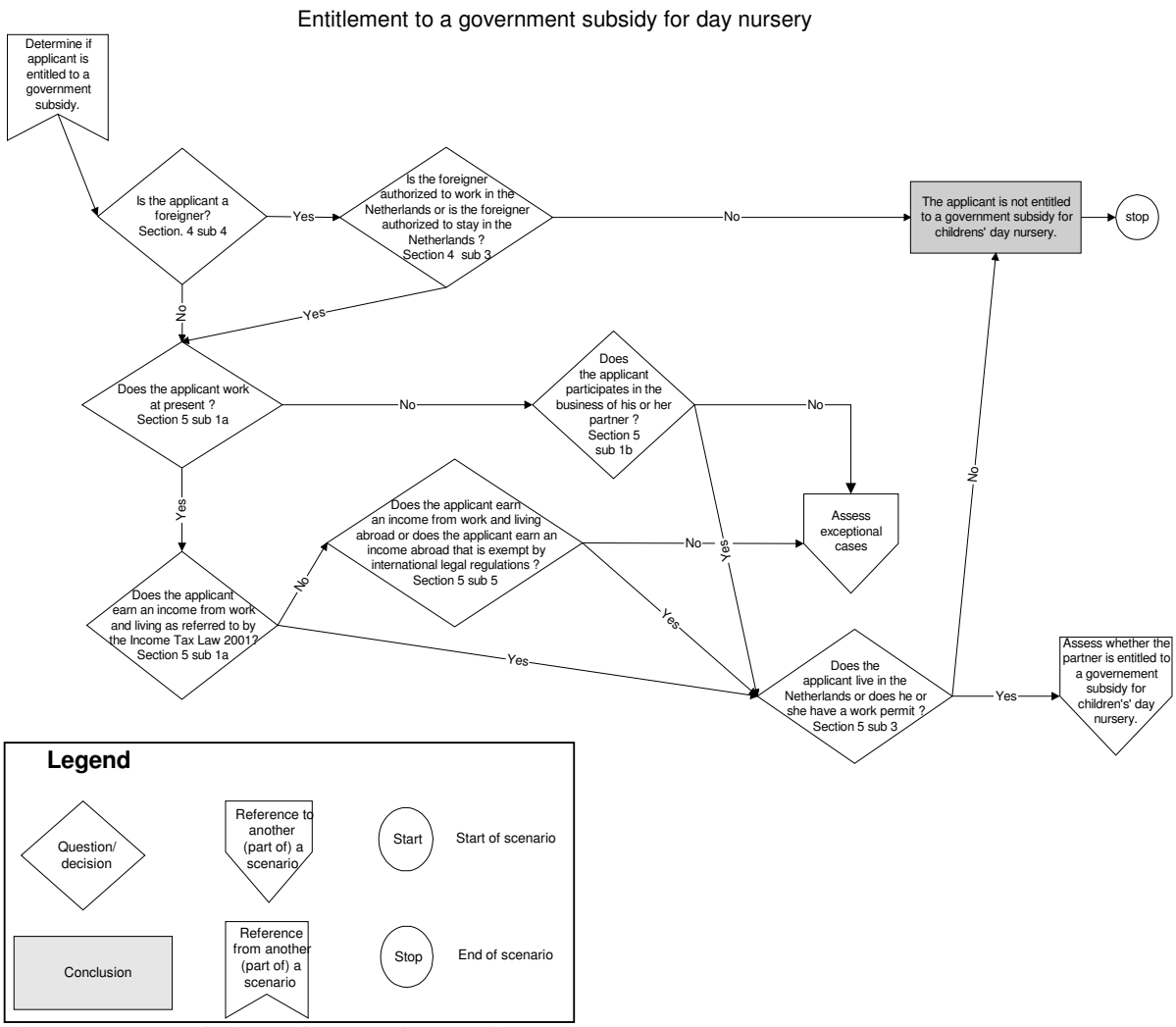

Figure 1: Example of a (part of) a scenario called "determine entitlement to a government subsidy for day nursery"

modeling process shares a global understanding of the domain, we might for example ask whether a certain section applies instead of posing several separate questions, each spelling out the exact conditions of the specific section. We leave the details to POWER UML models of legal sources (van Engers et al., 2001).

POWER scenarios are distinct from UML Scenarios in that they provide a pure functional idea of how legal experts use legislation to solve certain cases whereas UML Scenarios provide a "system view" on how a decision support system would function (D'Souza et al., 1999 [17]). UML scenarios are used to define the boundary of a system, whereas POWER scenarios can be considered as a (global) specification of the knowledge intensive process (which could be supported by a system). UML scenarios and POWER scenarios match when it comes to capturing the task flow. UML scenarios are used to capture the ideal task flow as perceived by end users. POWER scenarios provide us with the means for discovering the implicit tasks and task flow within legal domains. Legislation typically is declarative in nature. Tasks and task flow are revealed when asking the expert to apply the legislation to solve cases. These task may later appear in the UML activity diagrams that we make when modeling the task flow accompanying the POWER conceptual models (see next section). During the conceptual modeling phase the initial sketch of the task flow in the scenarios may alter as a result of gaining insight during the rigorous and precise analysis of the legal sources.

The idea of using these scenarios as the basis for implementing knowledge-based systems design may be tempting, but essential to the POWER approach is that we base our knowledge models on the legal sources rather than on the experts' interpretation of these sources. As we found out the experts' interpretations may be incomplete or even conflict with these knowledge sources (i.e. the law!). Furthermore a serious handicap of procedural representations is in the limitations they put on the implementation. The order of the different reasoning steps represented in the scenarios (remember it represents the experts' interpretation of solving a problem) may very well be not the most efficient one, or may not be the most effective one from a communication point of view. When designing an user dialogue for a knowledge-based system one may want to choose a different order for posing questions then you would derive from such scenarios. In the POWER method of modeling legislation, the (declarative) conceptual models are combined with separate task models, which we model using UML activity diagrams. The conceptual models are described in the next section.

4. THE POWER-CONCEPTUAL MODEL

Although the way conceptual models are represented was already published in previous publications (e.g. Van Engers et al 2001 [5]) for readers yet unfamiliar with this approach a short introduction is given in this section. The POWER-conceptual model is represented in a notation called Unified Modeling Language (UML see D'Souza and Wills 1999). This notation has become the standard notation for representing models in the domain of information technology, but there are many ways to use the notation. The usage defined in the POWER-method, starts by dividing the model in UML packages. The structure of packages within the translated conceptual model is identical to the hierarchy in the legislation (i.e. chapters, sections, articles, members etc.), which allows tracing all conceptual models, and products that will derive from them, to the original legislation. The structure of packages within the integrated conceptual model represents the definition of concepts found in the legislation, and the relationships between these definitions.

Within each UML package, the important concepts found in the legislation are modeled as types and attributes. As opposed to the more often used classification of concepts in classes, the use of types allows introducing redundant concepts, and is independent of the way the data will be structured in later applications. Examples of concepts that demonstrate potential redundancy are e.g. "Natural Person" and "Tax Payer". Attributes can be simple properties of existing concepts, such as the age of a natural person, or can be relationships between concepts, such as the children of a natural person. The references found in the legislation are modeled as an extension to the UML, which we called "Package Reference". A package reference is modeled as a classifier, which represents some not-yet-identified other packages. Finally, the norms within the legislation are modeled in a formal language, named Object Constraint Language (OCL), which is a part of the UML. The Object Constraint Language can for instance determine under which conditions a "Natural Person" becomes a "Tax Payer". This is written down in an invariant about "Natural Person", which is a statement in the OCL that uses all the concepts modeled about "Natural Person". One can use OCL in a similar way as one would use a reified first order predicate calculus to express a legal norm.

The translation from legal text into a POWER-conceptual model (expressed in UML/OCL) is a two-phase process: translate and integrate.

4.1 Translate

After deciding on the (restricted) scope of legislation (using the scenarios as described before) that piece of legal text is analyzed. First we analyze the hierarchical structure of legislation and within each chapter, article, section, the text is analyzed for references. This view of legislation contains sufficient detail for detecting structural defects that can be reported as attention points.

Then, concept extraction (supported by a natural language parser) is used to identify the concepts used in each chapter, article and section that are consequently put into a conceptual model.

Finally the norms within each block are written down as (OCL-) constraints (expressions over the concepts). The result of this step is a conceptual model that represents the unique interpretation of a single article of legislation, which does not depend on any other articles.

4.2 Integrate

Still within the scope of legislation, we can combine the articles that use identical concepts to create an integrated conceptual model. During this process, synonyms (different words, same meaning) are discovered to be identical concepts and homonyms (same word, different meanings) are distinguished as separate concepts. The structure of exceptions and extensions to the general rule is unraveled for each concept. At this point, a conceptual model is produced that represents the unique interpretation, but also the reasoning, involving specific interdependent concepts. This integrated conceptual model can be used to reason on a specific scope of cases, although some reasoning knowledge for rare cases may still be missing and may have to be added in future iterations or put out of scope for the project. The (integrated) conceptual models produced this way (the POWER-model) contain the legal knowledge. When this knowledge is combined with the process and task knowledge, we have a specification for a supporting knowledge-based component.

5. LEGAL QUALITY CONTROL OF THE BASIC FACILITY NURSERY'S ACT

5.1 Context and assignment

The Dutch Ministry for Social Affairs and Employment has written a bill regarding the Basic Facility Nursery's Act[1]. At the time of writing this article, the bill had to be passed by the Dutch Lower Chamber. The Basic Facility Nursery's Act (BFNA) aims at guaranteeing the quality and accessibility of day nursery and at creating possibilities for parents to combine jobs and child care. The Ministry of Finance is involved because the Dutch Tax Administration is assigned to be the executive organization. Implementation of the BFNA by the Dutch Tax Administration seems logical because of its acquaintance with similar business processes, however there is one difference: as a tax administration it executes fiscal processes, not subsidiary processes like the Basic Facility Nursery's Act. This new dimension places even greater demands on aspects such as validating legal quality and risk assessment.

The POWER-project team received the assignment of making a conceptual model of the Basic Facility Nursery's Act. The main reason for making a conceptual model of the Basic Facility Day Nursery's Act was to perform a quality check of the new legislation so possible defects could be repaired before the law would come into operation. Secondly, it enabled the Dutch Tax Administration to obtain a good insight into the content of the bill and consequences for the processes at the DTCA that were designed for implementation of the bill.

5.2 The procedure

First, the POWER knowledge analysts developed scenarios of the relevant part[2] of the BFNA. These scenarios were created in

[1] Wet Basisvoorziening Kinderopvang.

[2] The part of the BFNA that regarded the task of the Dutch Tax Administration.

cooperation with and validated by the only BNFA expert available at that time at the DTCA. Because of the scarce expertise and the new kind of legislation, the scenarios were immediately used by the different disciplines involved in the implementation design of the BNFA at the DTCA: particularly process and organizational design and compliance and risk assessment. The scenarios were also handed over to the Ministry of Social Affairs and Employment.

Next, the conceptual model of the relevant part of the BFNA was developed. The POWER-team has already developed some tools that support the modeling process: JEWEL for structure parsing of the legislation and OPAL (see e.g. Van Gog and Van Engers 2001) for the translation of the legislation into a formal model in UML. These UML-model are then exported to a case tool (e.g. Rational Rose or MEGA). In this project MEGA was used. The conceptual modeling took place in two phases: first translation of the legislation into partial models closely corresponding to the legislation text. Next, the integration and re-factoring of the partial conceptual models into complete and coherent conceptual models of the main concepts defined in the legislation.

Possible defects were found during the analysis of the legislation. They were reported to the experts at the DTCA and the experts at the Ministry of Finance. If possible defects indeed seemed defects, the experts passed the findings to the legislation drafters of the Ministry of Social Affairs and Employment. In the next paragraph, we will present some examples of the defects that were found while making a conceptual model.

When the conceptual model of the BNFA was completed, we derived a data model from the conceptual model. This (conceptual) data model - however incomplete at that time due to a missing Order of Council for the BNFA - proved very important for the DTCA in being able to estimate whether they could implement the BNFA, particularly with respect to back-office information systems, risk assessment and requirements for form design.

5.3 EXAMPLES OF DETECTED DEFECTS

The sections of the bill of the Basic Facility Nursery's Act[3] used in the examples are unofficial translations from Dutch.

5.3.1 Defect: no reference found & concept confusion

Basic Facility Nursery's Act

Section 22

2. If a parent is a person as referred to in section 5, second subsection, only the costs of nursery as referred to in the first and second subsection of that section are rated among costs that are associated with nursery as defined in that subsection.

The first step in the integration process is resolving all package references. Package reference "section 5 sub 2" that refers to "person" is easily solved as can be seen in the text of section 5 sub 2: the person is underlined in the text. Package reference "Section 5 subsection 1" proved impossible to solve: first it is an ambiguous reference as it can be read as a reference to "nursery" or to "costs of nursery". However, section 5 sub 1 does not contain the concept of "nursery" at all, nor of "costs of nursery".

Figure 2 shows the partial conceptual model of subsection 2 of section 22.

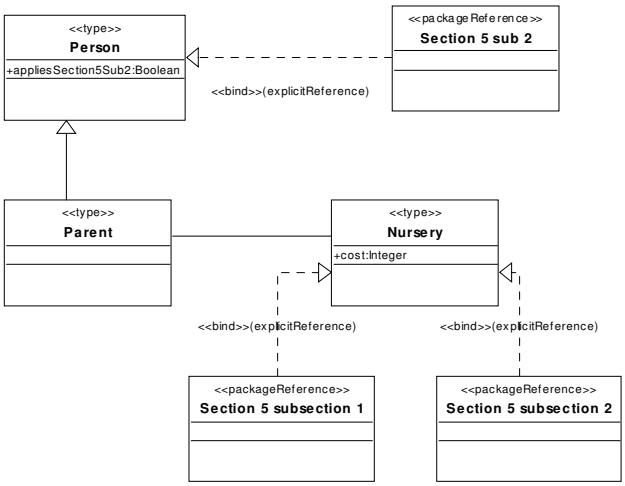

Figure 2. Partial conceptual model of section 22 subsection2.

The third package reference "Section 5 subsection 2" is the same ambiguous reference ("nursery"/"costs of nursery") as described before, but now it refers to subsection 2. In section 5 sub 2 the concept of "nursery" is indeed mentioned but the concept is certainly not defined there, as the text of section 22 sub 2 suggests: "costs that are associated with nursery as defined in that subsection [section 5 sub 2]". On a more close reading of the text of section 22 sub 2 the part of the sentence "the costs of nursery as referred to in the first and second subsection of that section are rated among costs that are associated with nursery as defined in that subsection" seems almost circular. These findings were reported to the experts.

Basic Facility Nursery's Act

Section 5

1. A parent is entitled to a government subsidy for a contribution year, if the parent in that year:

   a) works at present from which an income from work and living as referred to in the Income Tax Law 2001 is earned,

   b) [],

2. Also entitled to a government subsidy is a parent, insofar this is not a person as referred to in the first subsection, who:

   a) is handicapped or is a chronically sick patient, of which has been laid down by order, as referred to in section 20, that this is a necessary condition for nursery , or

   b) has a child with respect to whom, by order as referred to in section 21, has been laid down that nursery in the interest of a good and healthy development of that child is necessary.

[3] Version as presented to the Dutch Lower Chamber.

They agreed with the findings and wrote the following amendment:

> Section 22 *(amendment, concept version)*
>
> 2. If a parent is a person as referred to in section 5, second subsection, only the costs of nursery which are connected with the circumstances as defined in section 5, second subsection are rated among costs that are associated with nursery as defined in the first subsection.

The former package reference "nursery as referred to in section 5 second subsection" was replaces by "circumstances as defined in section 5, second subsection" and "nursery as defined in that subsection" was substituted by "nursery as defined in the first subsection [of section 22]". Repairing this deficit is beneficial to the Ministry for Social Affairs and Employment, the enforcement agency and the civilian in that it is clear now which costs of nursery are subsidizable.

5.3.2 Ambiguous reference & missing concept

In subsection 3 of section 2, a reference is made to *this law* for the concept of "being equated with having been married to one another". At first sight we modeled it as a package reference to the Basic Facility Nursery's Act. Then, when we tried to resolve the package reference, we could not find the concept of "being equated with being married to one another" at all in the BFNA.

> Basic Facility Nursery's Act
>
> **Section 2**
>
> 3. Anyhow there is report of having a joint household, as referred to in section 1.2, first subsection, beneath part b, 1°, of the *Income Tax Law 2001*, if the parent and a third person reside in the same house and
>
> a) they have been married to one another or earlier for the enforcement of this law have been equated with it, […].

On a more closer reading, the reference to *this* law could also be interpreted as a reference to the *Income Tax Law 2001*. If we used this interpretation, the package reference could indeed be solved, because the Income Tax Law 2001 does contain a definition of the concept of "the equation with marriage" in section 1.2 subsection 5:

> Income Tax Law 2001
>
> **Section 1.2**
>
> 5 For the enforcement of this law and the regulations that rest with it will for the determination of relationship the classification as partner be equated with marriage.

The equation with marriage from the Income Tax Law did not seem to correspond to what one would expect from the reference in the BFNA, so we asked the experts to what law the reference *this law* referred and what equation with marriage the legislator alluded to. The answer came that the reference could not be to the Income Tax Law, as the fiscal motives for equating a fiscal partner with marriage did not have any meaning in the context of the BFNA. The reference was indeed to the BFNA itself. The experts confirmed that the concept of the equation was empty or at least not sufficiently specified. This defect is a fine example of a treacherous reference which possible defects can only be found when the partial conceptual models are integrated and an attempt is made to resolve the reference. Because the POWER-method of modeling legislation works in such a structured way, these defects are always filtered out.

5.3.3 Gap in the law

The classification of the partner of a parent who is entitled to the subsidy for nursery, is an important concept defined in the BFNA. The subsidy is income-related and is related to the collective income of both parent and partner. From this perspective it is beneficial for the applicant not mention a partner when applying for this subsidy. When people are married, they are automatically partner for the BFNA, but if they are not, there are a number of regulations specified for classifying a housemate as a partner. We will not go through all of the regulations for partner, but there is one aspect that is equal for all of them: the regulation only applies if both parent (applicant) and its housemate (potential partner) are older than 18 years. This age limit comes from the definition of the classification as partner in the *Income Tax Law 2001*. As can be seen in the legal text of section 1, part b, the BNFA refers to the Income Tax Law for the concept of making the choice for classification as partner.

> Basic Facility Nursery's Act
>
> **Section 1**
>
> In this law and the regulations that rest with it is meant by partner:
>
> a) []
>
> b) the one who is not the partner of the parent for the enforcement of the *Income Tax Law 2001*, but *pursuant to article 1.2 of that law* together with the parent may make the choice for classification as partner.

We will illustrate the definition of classification as partner form the Income Tax Law 2001 with the following OCL constraint from the conceptual model of article 1.2 of the Income Tax Law 2001:

Constraint name	Income Tax Law 2001:: article 1.2 1b
Context	Person
Constraint text:	
self <> Parent AND NOT(Parent.isMarried) AND Parent.beOfAge AND NOT(self.isMarried) AND self.beOfAge AND sharesAHouseholdForMoreThanSixMonthsContinuouslyWith(Parent, ContributionYear) AND isRegisteredAtTheSameAdressDuringThePeriodOfTheJointHouseholdWith(Parent, ContributionYear) ==> self: Partner	

Example of an OCL constraint from the conceptual model of article 1.2 of the Income Tax Law 2001

From the constraint it is clear that both parent and person that may classify as a partner must be of age, that is, having attained the age of 18 years. The BNFA, however, has a special regulation (see section 3) that enables minors (e.g., teen-mothers) to apply for the subsidy for nursery.

> Basic Facility Nursery's Act
>
> Section 3
>
> A minor is competent to execute the legal transactions that are necessary to receive a subsidy pursuant to this law. […].

The combination of section 3 and the rules for classification as a partner reveal that for a teenager that has become a parent, the partner cannot be classified for the enforcement of the BFNA, even though they are living together. The result is that the income of the partner (not in the legal sense but in real life) is not taken account of, as it would be if the parent had been of age.

We submitted this issue to the experts, who told that the legislators had recognized this deficit and had thought that this situation would be so very rare that an amendment for this type of exception was not necessary.

5.3.4 Non applicable regulation

> Basic Facility Nursery's Act
>
> **Section 5**
>
> 1. A parent is entitled to a government subsidy for a contribution year, if the parent in that year:
>
> e) has not yet attained the age of 18 years, receives education and […],
>
> 4 A parent with a partner is only entitled to a claim, if the partner is also a person as referred to in the first or second subsection. […]

Section 5 subsection 1, combined with section 5 sub 4, shows a piece of legislation that for logical reasons can never apply: subsection 4 states that a parent (applicant for the subsidy) and its partner must both be persons that have right to a subsidy. The idea behind this is that the subsidy is only granted to families where both parents work or are returning to work; also special target groups can make a claim to the subsidy. Now, part *e* of subsection 1 [section 5] defines a target group with the property: "has not yet attained the age of 18 years". We just saw that this can never apply to a unmarried person who is a partner in the sense of the BFNA: he or she must have attained the age of 18 year, as this requirement is part of the definition of the concept of "partner".

This conclusion will not have far reaching consequences for the enforcement of the BFNA, but it is again an illustration of something that can be easily overlooked because of the complicated definition of the concept of Partner (it is largely imported from another (type of) law) and the recursive use of the definition of persons who make a claim to the subsidy.

The error found is one that can be found by a automated verification tool like VALENS. The proof-by-processing algorithm (see Gerrits and Spreeuwenberg, 1999) would detect that there are no situations in which the rule that corresponds to section 5, subsection1, part *e* can ever apply to a person to which a partner rule applies. Also the example of the "gap in the law", which we discussed before, is a defect that can be detected by an automated verification tool.

6. CONCLUSIONS

Using a systematic method to find anomalies in legal sources or even trying to prevent is not new. Systems that restrict the freedom of the legislation drafters have been proposed, but never seemed to have really worked despite their theoretical advantages. Luccardi (1994) for example suggest the use of decision tables before writing the rules. The clear advantage of this approach is the fact that decision tables don't allow most forms of anomalies, but at the price of lack of expressive power (and frustration for the legislation drafters of course). Voermans (1995 [21]) has a somewhat less restrictive proposal. He implemented the guidelines of the Justice department (and some additional ones) that are aimed at improving legal quality (especially by avoiding confusing language constructions etc.) in a supporting system (a text editor meant to be used by the legislation drafters). Kordelaar (1996) shows an example of an approach typical for the early nineties (building upon the ideas of e.g Bench-Capon et al. (1987) and Coenen et al. (1993)), i.e. building a knowledge-based system using expert knowledge of a specific juridical domain an testing this system against cases to test if the system (and consequently the legal basis of that system according to the followers of that approach) has the intended outcomes (if not you could repair the system or the legal source).

In the POWER program we took a different approach. First of all we respected the legislation drafters. We take their output (the law) as it is. Secondly we don't model second-hand knowledge but keep as closely as we can to the original knowledge sources, i.e. the law itself. Last but not least we use an accepted ICT-standard for knowledge representation (UML/OCL). Not because no better representation formalisms exist, but because in the end we have to deliver regular information systems. And despite the

fact that these systems are knowledge-based, requirements that hold for any other information system should be met (for example maintainability, performance etc.).

The knowledge representation formats we presented in this article are scenarios and the POWER-conceptual models expressed in UML/OCL. Expressive power of a UML type model using OCL (van Engers et al., 2001 [3] and [4]) is better compared to scenarios (see section 3) which lack the benefits of a strict formal model. But nevertheless scenarios seem to provide us with a good insight in the legal domain represented, especially when the legislation involved is to be used in a categorization or assessment task. Scenarios also proved to be an excellent means of communication with experts and representatives of disciplines involved in the implementation of legislation.

The POWER-method has shown to be a very useful approach for modeling (pieces of) the law. Within the POWER-program we apply the POWER-method not only for designing (and even generating) knowledge-based systems, but an even greater benefit lies in its possibilities to detect anomalies in legislation in an early stage of design (hopefully even before the law becomes effective). With the processes created around the formal models that are the result of applying the POWER-method in which different legal experts are involved, a feedback loop is implemented that improves the legal quality significantly.

In this article we showed some examples of errors found in a bill. This article presents only a small portion of the anomalies that were found. The project has been very successful in making a contribution to the quality of legislation. We also analyzed a part of the Order in Council that belongs to the bill. We found ten deficits that were immediately reported to the legislators (analysis and report within one day). We also did this for the two following concept versions of the Order in Council, and will do this for the versions to come. The exact results can not be published because the Lower Chamber is discussing the Order at present. Because it concerns the part of the Order in Council that specifies the exact calculations of the amount of subsidy, the deficits we reported had rather large economic value for both DTCA and civilians. It is a perfect illustration of how knowledge analysts, experts and legislation drafters can interact with the aim of realizing a sound and enforceable piece of legislation. The representations used in the POWER-method have shown to be also very useful when designing law-enforcement strategies, design (E-)forms etc. By providing insights in the data-elements needed to enforce the law one can think about alternative process designs for the law-enforcement organization (in this case the DTCA). Furthermore, estimates can be made, based upon this information about the administrative costs that would result from effectuation of the law. Also, the inventory of data-elements needed for the law enforcement can be used to advise the legislation drafters if implementation problems are to be expected or not.

In a future project it would be interesting to see what errors would be detected by a automated verification tool in relation to the errors found during the modeling process. We know that during the "manual" modeling process we find many more errors than by applying a verification tool alone. This is logical because the analyst also finds the semantic errors and errors resulting for example from incorrect references within legislation. However it would be interesting to see if there is a category of errors that escape the attention of the analyst (and experts), but that can be found by an automated verification tool.

Many things still can be improved, like the natural language parsing components in the POWER-supporting tools (we are e.g. working on parsing deontic expressions in the law into OCL), but with the POWER-method the designers of new legislation and the designers of the administrations' processes and systems already have a very 'powerful' instrument at their disposal. In the near future we hope to further improve the POWER-method and its' supporting tools (especially the POWER-workbench).

7. ACKNOWLEDGMENTS

We thank the European Commission for taking the IST 5th framework program initiative.

8. REFERENCES

[1] Bench-Capon, T.J.M., Robinson, G.O, Routen, T.W. and M.J. Sergot, , 1987, Logic Programming for Large Scale Applications in Law, First International Conference on AI and Law, Boston

[2] Coenen, F., Bench-Capon, T.J.M., 1993, Maintenance of Knowledge-Based Systems: Theory, Techniques and Tools (Apic Series, number 40), Academic Press.

[3] van Engers, T.M., Kordelaar, P., 1999, POWER: Programme for an Ontology based Working Environment for modeling and use of Regulations and legislation. Proceedings of the ISMICK'99.

[4] van Engers, T.M., Kordelaar, P.J.M., den Hartog, J., Glassée, E., 2000, POWER: Programme for an Ontology based Working Environment for modeling and use of Regulations and legislation. In Proceeding of the 11th workshop on Database and Expert Systems Applications (IEEE) Eds. Tjoa, Wagner and Al-Zobaidie, Greenwich London, ISBN 0-7695-0680-1, pp. 327-334.

[5] van Engers, T.M., Gerrits, G., Boekenoogen, M.R., Glassée E., Kordelaar, P., 2001, *POWER: Using UML/OCL for Modeling Legislation – an application report*. Proceedings of the ICAIL 2001, St. Louis.

[6] van Engers, T.M., Glassée, E., 2001, Facilitating the Legislation Process using a shared Conceptual Model. In: IEEE Intelligent Systems January 2001, p. 50- 57.

[7] van Engers, T.M., Van Driel, L., Boekenoogen, M., 2002, The effect of formal representations formats on the quality of legal decision-making, in Legal Knowledge and Information Systems (Bench-Capon, Daskalopulu, Winkels eds.), IOS Press Amsterdam, ISBN 1 58603 299 2, p 63-71.

[8] Gerrits, R., Spreeuwenberg, S., 1999, A Knowledge Based Tool to Validate and Verify an Aion Knowledge Base, Validation and Verification of Knowledge Based Systems, Theory, Tools and Practice, 67 – 78

[9] Gerrits,,R., Spreeuwenberg, S., Boekenoogen, M., 2000, VALENS: A Knowledge Based Tool to Validate and Verify an Aion Knowledge Base, Proceedings 14th European Conference on Artificial Intelligence, 731 – 735

[10] Glassée, E., Van Engers, T.M., Jacobs, A., 2003 POWER: An Integrated Method for Legislation and Regulations from their Design to their Use in E-government Services and Law Enforcement, in Digital Legislation (ed Moens), die Keure, Brugge, ISBN 90 5958 039 7, p175-204

[11] van Gog, R., Van Engers, T.M., 2001, Modeling legislation using natural language processing, in: 2001 IEEE International Conference on Systems, Man & Cybernetics, Tucson, Arizona, , ISBN 0-7803-7089-9.

[12] den Haan, N, Automated Legal Reasoning, University of Amsterdam, Amsterdam, 1996 (diss.)

[13] Kordelaar, P.J.M., 1996, Betere wetten met kennissystemen, Phd-Thesis Twente University, ISBN 90-365-0865-7.

[14] Luccardi, G.L., 1994, Functional Object-Types as a Foundation of Complex Knowledge-Based Systems, Phd-Thesis, Technical University Eindhoven, ISBN 90- 6743-350-0

[15] Nieuwenhuis, M.A., Tessec: een expertsysteem voor de Algemene Bijstandswet, Universiteit Twente, Enschede, 1989 (diss.)

[16] Preece, Shingal, 1994, Foundation and Application of Knowledge Base Verification, International Journal of Intelligent Systems, 9, 683 – 701

[17] D'Souza D.F. and Wills A.C., 1999, Objects, components and frameworks with UML: the Catalysis approach, Addison-Wesley, ISBN 0-201-31012-0.

[18] Spreeuwenberg, S., Van Engers, T.M., Gerrits, R., 2001, The Role of Verification in Improving the Quality of Legal Decision-Making, in Legal Knowledge and Information Systems, IOS press, ISSN 0922-6389.

[19] Svensson, J.S., Kennisgebaseerde micro simulatie, Universiteit Twente, Enschede, 1993 (diss.)

[20] Vanthienen, J., Mues C., Wets, G., 1997, Inter-Tabular Verification in an Interactive Environment, Proceedings Eurovav 97, 155 – 165

[21] Voermans, W., 1995 Sturen in de mist…, maar dan met radar; een onderzoek naar praktisch haalbare vormen van computerondersteuning bij het ontwerpen van regelingen, dissertatie KUB, Tjeenk Willink, ISBN 90 271 4269 6

[22] Voermans, W., 2000, Computer-assisted legislative drafting in the Netherlands: the LEDA-system, A National Conference on Legislative Drafting in the Global Village

[23] Waterman, D.A, 1986, A guide to Expert Systems, Addison-Wesley Publishing Company, Reading.