Legal Engineering: A Knowledge Engineering Approach To Improving Legal Quality

Prof. dr. Tom M. van Engers

University of Amsterdam/
Leibniz Center for Law
www.lri.jur.uva.nl
t.m.van.engers@acm.org

Abstract. Knowledge engineers have been working in the legal domain since the rise of their discipline in the mid-eighties of the last century. More and more knowledge engineering is not just seen as a way of capturing and distributing (legal) knowledge by means of the knowledge-based systems created, but as an analytical approach that helps to improve legal quality. Improving legal access of course remains equally important. This insight has lead to the application of knowledge engineering methods at a much earlier stage in the development of normative systems (including legal systems), preferably when the (new) norms are created or adapted (legislation drafting). A good example of this approach is the POWER-approach developed in the Netherlands by the Dutch Tax and Customs Administration (DTCA in Dutch: Belastingdienst) and some partners (see e.g. Van Engers et al., 1999, 2000, 2001, 2003). This POWER-approach offers both a method and supporting tools that support a systematic translation of (new) legislation into the administrations' processes. This paper describes how this systematic approach that has its origins in knowledge engineering can help to improve legal quality. The POWER-method not only helps to improve the quality of (new) legislation. It also supports codification of the legal knowledge into procedures, computer programs and other designs. One of the advantages thereof is the reduction of the time-to-market of the implementation of legislation and its increased transparency (which will lead to reduced maintenance costs. In this chapter we will focus on legal quality improvement and explain some knowledge representation techniques that we use to enable this. In contrast to other knowledge modelling approaches the POWER-approach is focused on modelling legal sources rather than expert knowledge. Expert knowledge however is needed to find the correct interpretations but also for efficiency reasons. Starting with representing the (legal) experts' knowledge (using scenarios) helps 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 knowledge that can be distilled 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.

J. Padget, R. Neira, J. Díaz de León (Eds.). e-Government and e-Democracy: Progress and Challenges
© IPN, Mexico 2004.

1 Introduction

Getting the right knowledge at the right place at the right time has always been a great challenge for governments since this inflicts the ability to effectuate the legislative power to regulate and control. The Dutch Tax and Customs Administration (ĐTCA) has developed a method and supporting tools supporting the whole chain of processes from legislation drafting to executing the law by government employees and citizens (see e.g. Van Engers et al., 1999, 2000, 2001, 2003). These method and tools resulted from the POWER research program (Program for an Ontology-based Working Environment for Rules and regulations), a research program that was partly sponsored (the E-POWER project) 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. These processes are very vulnerable to errors. Not only because of the intrinsic complexity of the law, but also because mostly a large number of people is involved in these processes as wells as of the complexity of these processes themselves. 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. The same holds when completely new legislation has to be implemented.

The knowledge and experience needed to create new laws or adapt existing ones, specify, design and implement procedures and systems in legislative domains is very scarce. A (piece of) law should reflect the intentions of the political responsible minister and should also meet some quality criteria such as clarity and consistency from the perspective of the law-enforcement organization. This is the responsibility for the legislation drafters that are responsible for drafting the new law.

The people responsible for implementing the law (i.e. the administration) have to adapt the procedures, processes and information systems to the new law. 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 for the citizens.

Between drafting new legislation and enforcement thereof 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.

Many legislation drafting departments at the different ministries already have their own quality insurance techniques. Furthermore in many cases the ministry of Justice has a special role because they are usually responsible for the overall legal quality of a country. Despite all the effort that's been spend on improving legal quality using traditional measurements, such as co-reading (peer reviewing etc.) many anomalies can still be found in recently drafted legislation. The situation is even worse in situations when existing legislation is adapted.

Quality insurance measurements also exist for the other processes in the chain mentioned. Most attempts to achieve quality improvements however focus at just one of the processes involved. In the approach developed in the POWER-research program we consider each of these processes as equally important. We furthermore stress the importance of managing the chain rather then the distinctive processes themselves.

Improving legal quality is just 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. Central in the approach is the central role for formal (and semi formal) knowledge representations. In the POWER-approach different knowledge representation formats are used. This paper describes how these knowledge representations are used and how they contribute to improving legal quality.

The knowledge representation formats discussed in this paper are procedural description called 'scenarios' (which are more or less comparable to UML action diagrams) and POWER-conceptual models (expressed in UML/OCL). Although scenarios (see section 3) lack the benefits of a strict formal model expressed in UML/ OCL (van Engers et al., 2001 [3] and [4]) they are useful to provide both analysts and experts 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 (see Van Engers et al. 2002 [7]).

This chapter that is based on previous work (see Van Engers and Boekenoogen 2003) we present some results form 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) or improving the quality of knowledge bases used in knowledge-based systems (see e.g. Preece 1994 and Vanthienen 1997).

Legal sources (including law texts) suffer from different quality problem, some of them are fundamental ones. First of all the legal sources are expressions in 'natural' language ('natural' is between quotes because many citizens might consider a piece of law beyond the language they are familiar with). Natural language often is ambiguous and this feature can not always be avoided even when it is used to specify a normative system (e.g. a specific law). Besides this feature of the natural language that is used to carry the desired behaviour of the normative system described by the legislator other sources of vagueness exist. Legal sources often contain so-called open evaluative terms. These terms are intentionally used to achieve a certain amount of sustainability (i.e. making the regulation robust to a certain level of change in the world that is subject to the regulation). Another reason for vagueness is due to the fact that legislation often results from a political debate and is as such a compromise of different interests.

Two techniques, verification and validation (V&V), can be used to test the quality of the knowledge representations. Validation deals with testing the knowledge representation against the legislation drafters' intentions. Verification deals with the consistency of the formal representation, while validation deals with the intentional aspects of the representation (does the model represent what it should). 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 (which is a specific representation form of a formal model). VALENS can be used by a developer after or during construction of a rule-base or can be integrated in a tool that allows users to write their own business rules. The output of the tool is a document in which all invalid rules (combinations) detected are reported. Each fault is classified and explained.

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. constructing a rule-base. Since this process of formalizing the legislation into a formal representation generating a rule-base out of it (van Engers et al., 2001 [5]) and applying a verification process to it, usually takes some time even when it is supported by tools such as the Power-workbench, a less subtle and profound approach is needed to satisfy the practical needs of legislation drafters and policy makers need feedback .Especially if in the drafting process, where these drafters deal with the politicians and other influential stakeholders feedback is needed in a much earlier stage. Furthermore it is not always necessary to design a (rule-based) system at all. Therefore a less labour and time intensive method derived from the original Power-method that helps to find anomalies has been developed. That method is called the Power-light method. The fact that no formal model actually results from that Power-light approach is a small price to pay, given the demands mentioned earlier.

In this chapter we will show how the POWER-approach can be used to detect anomalies in legislation. The Power-light method works quite similar, except for the fact that formalization only takes place in the modellers' mind. During the modelling process using the POWER-method, more defects are found than would have been found when applying automated verification (e.g. with VALENS). Automated verification, combined with conceptual modelling, is still very valuable because it might detect defects that have not been noticed during the modelling of the legislation. Automated verification has also proved its value when applied to a knowledge-base that has received maintenance (Gerrits et al., 2000). 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 chapter we didn't apply VALENS, nevertheless as an illustration, we will indicate which of the examples of errors found are errors that could have been found with VALENS if we would had applied it to the corresponding rule-based implementation.

3 Scenarios

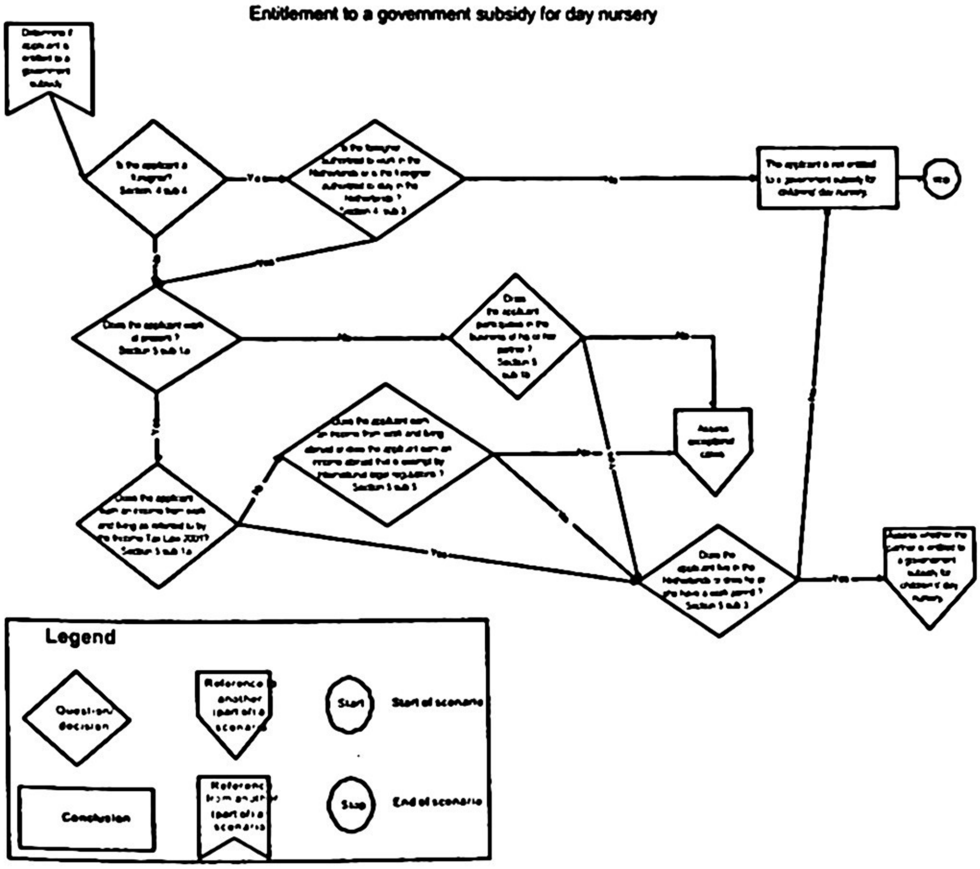
Entitlement to a government subsidy for day nursery
Yes
No
Legend
Question/decision
Reference to another part of a scenario
Start
Start of scenario
Conclusion
Reference from another part of a scenario
Stop
End of scenario

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

Before making a formal representation of a certain domain it proved 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 action diagrams. Within the POWER research program we use a special form of such action diagrams which we call 'scenarios' because they represent the possible scenarios of solving cases (see also Van Engers et al. 2002 [7]).

We experienced in the projects ran at the DTCA that developing such scenarios at the start of the knowledge modelling process helped both knowledge analysts and legal experts, especially in case of modelling new or complex legislation.

The process of creating a scenario goes 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 can be "handled" 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 the form of a kind of 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. Figure 1. presents an example of a part of a scenario that was made of a bill concerning subsidies for children's day nursery.

These scenarios have showed to be a rather effective representation if we want to communicate between knowledge modellers and legal experts (see Van Engers et al 2002 [7]). Not only the scenarios create a quick and global overview of the legal domain at hand, but also they serve different purposes.

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 we aim at providing just a quick overview of how a certain legal domain functions. If everyone involved in the modelling 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. As with calculations, we leave the details to POWER UML models of legal sources (van Engers et al., 2001).

POWER scenarios provide a pure functional idea of how legal experts use legislation to solve certain cases: it does not provide a "system view" on how a decision support system would function. This is the distinction between a POWER scenario and a UML scenario, which is a story about how a system will be used. A UML scenario describes a prototypical sequence of interactions in a business-collaboration or the system context (D'Souza et al., 1999). The main difference between POWER scenarios and UML scenarios is that 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). However different, UML scenarios and POWER scenarios match when it comes to the goal of 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.

In POWER scenarios, 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. Note that such a diagram not necessarily contains an end point of the reasoning path: the decision tree may be connected to another issue ("task"). In the next sections scenarios refer to POWER scenarios, not UML scenarios (unless explicitly stated otherwise).

The idea of using these scenarios as the basis for 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 their limitations they put on the implementation. The order of the different reasoning steps represented in the scenarios may very well be not the most efficient one. Also 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 form such scenarios.

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 in this section a short introduction is given. The POWER-conceptual model is represented in a notation called Unified Modelling 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 modelled 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 modelled as an extension to the UML, which we called "Package Reference". A package reference is modelled as a classifier, which represents some not-yet-identified other packages. Finally, the norms within the legislation are modelled 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 modelled 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 analysed. First we analyse the hierarchical structure of legislation and within each chapter, artic le, section, the text is analysed 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 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 intended date of commencement is on the first of January 2004. 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.

In the middle of 2002, the POWER-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.

[1] Wet Basisvoorziening Kinderopvang.

5.2 The procedure

First, the POWER knowledge analysts developed scenarios of the relevant part[2] of the BFNA. These scenarios were created in cooperation with and validated by the only BNFA expert available at that time at the DTCA. Because of the scarce expertis e and the totally 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 had already developed some tools that support the modelling process: a structure parser for detecting the structure of a piec of law (e.g. the chapters, sections, articles, members, sentences etc.) and a natural language processing based parser for automated concept extraction (see e.g. Van Gog and Van Engers 2001) used for the translation of the legislation into a formal model in UML. These UML-models are then exported to a case tool (e.g. Rational Rose or MEGA). In this project MEGA was used. The conceptual modelling took place in two phases (see Van Engers et al. 2001 [5]): 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 of this piece of law.

As a result of completing the conceptual model of the BNFA, we derived a data model from the conceptual model. This data model, indicating the data necessary for applying knowledge-based components based on the conceptual 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 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 all unofficial translations from Dutch.

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

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

Defect: no reference found & concept confusion

Figure 2 shows the partial conceptual model of subsection 2 of section 22. The first step in the integration process is resolving all package references. Package reference "section 5 sub 2" that refers to the person is easily solved as can be seen in the legal text below: 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).

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.

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.

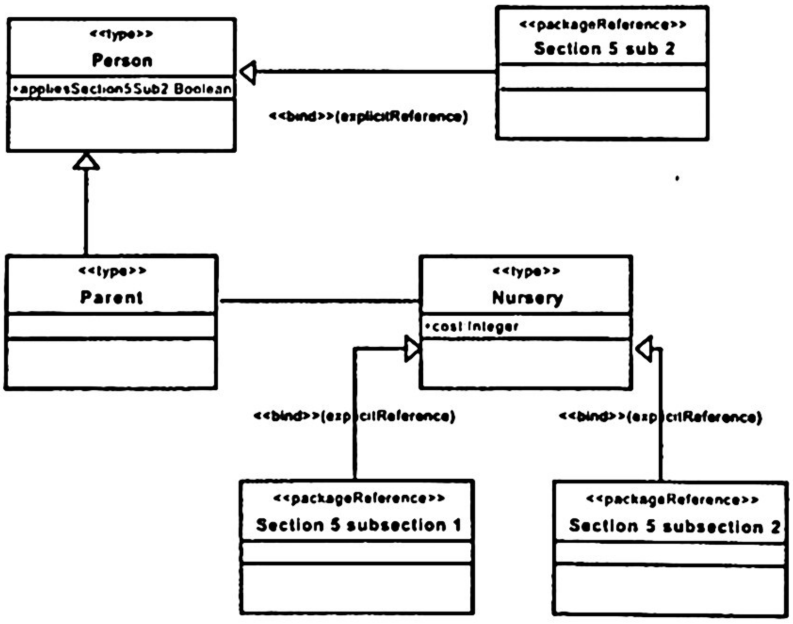

Figure 2. Partial conceptual model of section 22 subsection2.

The last 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 subsection 2 the concept of "nursery" is indeed mentioned but the concept is certainly not defined there, as the legal text suggests: "costs that are associated with nursery as defined in that subsection [section 2]". On a more close reading of the text of section 22 sub 2 it almost seems circular. These findings were reported to the experts. 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 defined in section 5, second subsection are rated among costs that are associated with nursery as defined in the first subsection.

Ambiguous reference & missing concept

> 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, [...].

> 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

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 modelled 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. On 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:

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 finally 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 is 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 modelling legislation works in such a structured way, these defects are always filtered out.

Gap in the law

The classification of the partner of a parent who has children that go to a nursery and who is entitled to the subsidy for nursery, is an important concept defined in he BFNA. This is because the subsidy is income-related and it is related to the collective income of both parent and partner. 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.

Section 3

A minor is competent to execute the legal transactions that are necessary to receive a subsidy pursuant to this law. [...].

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.isAdult AND
NOT(self.isMarried) AND self.isAdult AND
sharesAHouseholdForMoreThanSixMonthsContinuouslyWith (Parent, ContributionYear)
AND
isRegisteredAtTheSameAdressDuringThePeriodOfTheJointHouseholdWith(Parent, ContributionYear)
=> self: Partner
```

Figure 3. 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 adult, 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. 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 adult.

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.

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. [...]

In section 5 subsection 1, combined with section 5 sub 4, we have 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 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 a verification tool.

6 Conclusions

The POWER-method has shown to be a very useful approach for modelling normative systems. These systems are described in laws and other regulations including regulations that are used outside the government e.g. insurance policies. The POWER-method is not only suited for designing (and even generating) knowledge-based systems. One of its benefits lies in its possibilities to detect anomalies in legislation in an early stage of design (preferably 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 has proven its power to improve the legal quality significantly. This makes the Power-method a powerful tool for both legislation drafters, law-enforcement organizations and other organizations that are responsible for the design and/or execution of large bodies of regulations.

This paper shows some examples of errors found in real-life projects. In the study, both the original Basic Facility Nursery's Act and a part of the Order in Council that belongs to that act were analysed. The exact results can not be published yet because the Dutch House of Parliament is still discussing the Order at present. Ten deficits were found and immediately reported to the legislators (analysis and report within one day). The method was also applied to the two following concept versions of the Order in Council, and will be done for the versions to come. The project 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 which errors would be detected by a automated verification tool in relation to the errors found during the modelling process. We know that during the "manual" modelling 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 a 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 (the POWER-workbench). We thank the European Commission for taking the IST 5th framework program initiative.

References

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

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

[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] van Engers, T.M., Boekenoogen, M.R., 2003, Improving Legal Quality: an application report, in proceeding of ICAIL2003.

[9] 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

[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] 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

[12] 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.

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

[14] Kordelaar, P.J.M., Betere wetten met kennissystemen, Universiteit Twente, Enschede 1996 (diss.)

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

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