

Modelling the draughtman's craft: the LEDA-project Legimatics and legimatica-projects in the Netherlands

WIM VOERMANS*

1. Introduction: the legimatic interest

Especially in Europe interest and research into legimatics (i.e., the field that concerns the study of and the research into the possibilities of informatics for legislative drafting¹) are booming. After, among others, researchers like Layman E. Allen,² and Mark Sergot³ proved that computerscience and, more in particular, AI-techniques, bore significant relevance for legislative drafting, different projects throughout the world were initiated. Especially the researchers at the Italian 'Istituto per la Documentazione Giuridica del C.N.R.' have pioneered in this field since 1986.⁴ The Florence-group, working at this Institute, even succeeded in developing prototypes for computersystems which could be used to assist

* Assistant professor legal informatics & constitutional law - Faculty of Law - Tilburg University - P.O. Box 90153 - 5000 LE Tilburg - The Netherlands - e-mail: Voermans@kub.nl - tel: +31-13-668123 - fax: +31-13-663143

¹ The concept and name 'Legimatics' or 'Legimatica' was introduced by C. Biagoli, P. Mercatali and G. Sartor in their book 'Elementi di Legimatica', published by CEDAM in Milan 1993.

² See Laymen E. Allen, Symbolic Logic: a razor-edged tool for drafting and interpreting legal documents, in: *Yale Law Journal*, vol. 66, 1956/1957, p. 833-879; Laymen E. Allen, Language, Law and Logic: Plain Drafting for the Electronic Age, in: B. Niblett (ed.), *Computer Science and the Law: An Advanced Course*, Cambridge, 1980, p. 75-100 and L.E. Allen and C.S. Saxon, Computer-aided normalizing and unpacking: some interesting machine-processable transformations of legal rules, in: C. Walter (ed.), *Computing Power and Legal Reasoning*, New York, Westport Connecticut 1985, p. 243-316.

³ See M. Sergot, F. Sadri, R.A. Kowalski, F. Kriwaczek, P. Hammond en H.T. Cory, The British Nationality Act as a logic program, in: *Communications of the ACM*, 29, p. 370-383; M. Sergot, Representing legislation as logic programs, in: J.E. Hayes, D. Michie and J. Richards (eds.), *Machine Intelligence 11*, Oxford 1988, p. 209-260.

⁴ See P. Mercatali, Strumenti automatici per il controllo della leggibilità di documenti giuridici, in: *Linguaggi*, 3 (3), p. 64-66. See also C. Biagoli, Elementi per la definizione di un linguaggio per la rappresentazione di testi normativi giuridici, in: *Informatica e diritto*, 14 (2), p. 33-51.

legislators (i.e. the professionals tasked with the drafting of bills) ‘intelligently’ and informationwise, in drafting bills. These prototypes, named Lexedit and Lexedit 2, meant the kickoff for other legimatic projects in Italy that resulted in other systems that on this moment are actually being used by different (regional) governmental bodies and agencies throughout Italy. Parallel to the Italian initiative a lot of different practically geared projects, especially in Europe were started. The Civil Law tradition in a lot of European countries, where the content of law relies strongly on written legislation, explains the strong – mostly very practical – European interest into this area over the last ten years. In Civil Law countries the quality and applicability of legislation as a means to govern an ever rapidly changing society are major issues. In a lot of European countries it was felt that state-of-the-art computer-assistance for legislative drafting could well mean a beneficial form of support for legislators in the field of systematic quality-control of bills. The span of this paper does not allow a comprehensive discussion of a great deal of legimatic projects. I will restrict myself to a brief discussion of some interesting legimatic projects in the Netherlands, which are, as I see it, exemplary for thoughts and (research)projects in other European countries, and indeed, projects throughout the world.

2. *Legimatic approaches*

How can legislators benefit from computerscience in general and AI-science more in particular? In the Netherlands various approaches to these questions were adopted over the last ten years. Two approaches are however predominant. First of all there are those who address the aforementioned question departing from a primarily *AI-oriented* point of view. In this AI-oriented approach the principal question is how AI-concepts and -techniques can be put to use in building systems that will benefit legislators. (Deontic) Logic calculation and (deonto)logical representation of the substance of drafts are key issues in this strategy. In this AI-line of thinking lot of attention is focused on revealing and representing the knowledge that is expressed in a draft itself. Once the knowledge within an existing draft is represented and formalized in (deonto)logical entities, computer systems can be conditioned to reason with it.⁵

⁵ See for examples of the AI-oriented approach the contributions of Nienke den Haan, Jörgen Svensson, Cees Groenendijk/Henning Herrestad, Patries Kordelaar and T.

In this AI-, or knowledge based, approach the drafting of a bill is treated much in the same way as the problem of law-application. Conceptualizations of that what legislative draughtsmen do during the drafting stage, and the information they use, are scarce in this line of thinking. Characteristic for this approach is the technology pull: AI-concepts and -techniques define the (im)possibilities of computerized (intelligent) drafting assistance.

A distinctly different approach is, what we can call, the *information-oriented* approach. In this research strategy the focus is on the information-needs of legislative drafters during the drafting process. Conceptualization and representation of drafting activities are the principle issues within this approach. Key questions according to this strategy therefore are: what is it that draughtsmen do during the drafting process, and, what kind of information, or what kind of knowledge do they use during the drafting process, and, how do they use it? In this strategy conceptualizations of drafting activities are the basis for possible system development. Because different drafting activities generate different information-needs, or knowledge-requirements, different techniques or tools (sometimes even AI-techniques) are used to meet these needs in the drafting support systems built according to this method. Characteristic for this approach is the demand pull: the different information-needs during the drafting process define what is technically desirable for the development of computerized drafting support systems.

Both the AI- and the information-oriented strategy have resulted in realization of (four) computerized drafting support systems in the Netherlands, though not all of these systems are actually being used in practice. According to the strategy used to build the systems and the main functionalities of these systems, they can be divided into two major categories:

- a) *legislative analysis and review systems* (using the AI-oriented method);
- b) *semi-intelligent drafting support systems* (using a more information-oriented method);

In the next paragraphs I will discuss these categories and systems briefly, and elaborate on one system – the LEDA-system – more in particular.

3. *Legislative analysis and review systems*

The existing legislative analysis and review systems in the Netherlands were all built using the AI-approach. Legislative analysis and review systems assist legislators in determining the consistency or the consequences of already *existing* (draft) regulations.⁶ What these systems are able to do, in fact, is making (deonto) logical inferences, i.e. logical calculations, using the normlogical substance of an existing draft. To be able to perform this functionality, natural language (draft) regulations have to be translated or modelled in terms of knowledge representation formalisms, such as deonto(logic) formalisms, in order to allow the system to reason with it. For this AI-concepts and AI-techniques are used. Although the need for a formal translation can pose a serious drawback in the time-pressed legislative drafting process, these systems have obvious advantages for complex legislative drafting projects, especially when draft-regulations have considerable quantitative (e.g. financial) aspects, or when numerous behavioral possibilities and situations have to be normalized in a consistent manner (e.g. traffic regulation). An additional benefit of draft-analysis and -review systems is that these systems force legislators to think more fundamentally about the deontological structure of their drafts. This confrontation may invite them to come up with logical equivalent alternatives for certain solutions. The necessary formalization and representation of drafts can also result in blueprints for knowledge based administrative (handling)systems. This latter pungent, but in most cases still latent, feature is hardly ever discussed in legal computer science literature however.⁷ In the Netherlands, two of these legislative analysis and review systems have been developed by the Ministry of Social Services and Employment (ExpertisZe)⁸ and by the University of

⁶ See for instance Laymen E. Allen, *Language, Law and Logic: Plain Drafting for the Electronic Age*, in: B. Niblett (ed.), *Computer Science and the Law: An Advanced Course*, Cambridge, 1980, p. 75-100; N. den Haan en J. Breuker, *A Tractable Juridical KBS for applying and teaching Traffic Regulations*, in: J.A. Breuker, R.V. de Mulder, J.C. Hage (eds.) *Legal Knowledge Based Systems; Model-based Legal Reasoning*, Lelystad 1991, p. 5-16 and numerous other authors etc.

⁷ See R.W. Overhoff en L.J. Molenaar, *In de regel beslist*, diss. RUL, Den Haag 1991 and T. van Buggenhout c.s., *The decision table technique as a part of a computer supported procedure of legal drafting*, in: J.S. Svensson, J.G.J. Wassink, B. van Buggenhout (ed.), *Legal Knowledge Based Systems: Intelligent Tools for Drafting Legislation - Computer Supported Comparison of Law*, Lelystad 1993, p. 71-80.

⁸ See for a detailed discussion J.G.J. Wassink, *Kennistechnologie en het ontwerpen van regelgeving*, Den Haag 1992. See for an English description of (the basis fo) this system P. Kordelaar, *Supporting the drafting of a new dutch national assistance act with*

Amsterdam (TRACS).⁹ ExpertiSZe is a system which allows legislative draughtsmen to consider the (logical) consequences of changes in social security legislation. In a largely similar way the TRACS system makes it possible to determine and analyze the consequences of Dutch traffic regulations.

Characteristic for legislative analysis and review systems is that the model or conceptualization underlying these systems is – in most cases – a model or a concept of law (or draft) application. These systems apply drafts in order to seek out the consequences and confront draughtsmen with the implications. Determining and analyzing the consequences of drafts are (however important) only minor aspects of legislative drafting. Legislative analysis and review systems therefore only support some aspects of legislative drafting. What they actually do is largely treating legislative drafting like legal problemsolving.

The difference between legislative and legal problemsolving

There is however a big difference between legislative drafting and legal problemsolving by way of (forms of) law-application. The legislative drafting process or the legislative decision-making process is, for instance, only partly dependent on legal problemsolving, legal knowledge and legal reasoning. In comparison with other forms of legal problem-solving (like application of the law), legislative problemsolving, i.e. the decision-making process aimed at the enactment of legislation, is much more dependent on world knowledge (common sense), and it equally involves, throughout the different stages, substantial political, economic and social-scientific reasoning.¹⁰ Furthermore, the legislative process does not primarily result in legally (in)valid conclusions, but rather in 'relatively appropriate' solutions, or in convincing arguments.¹¹ Whether a bill is an

ExpertiSZe, in: J.S. Svensson, J.G.J. Wassink en B. van Buggenhout, *Legal Knowledge Based Systems; Intelligent Tools for Drafting Legislation, Computer-Supported Comparison of Law*, Lelystad 1993, p. 95-105.

⁹ See for instance N. den Haan en J. Breuker, A Tractable Juridical KBS for applying and teaching Traffic Regulations, in: J.A. Breuker, R.V. de Mulder, J.C. Hage (eds.) *Legal Knowledge Based Systems; Model-based Legal Reasoning*, Lelystad 1991, p. 5-16 and N. den Haan, Towards support tools for drafting legislation, in: J.S. Svensson c.s., *Legal knowledge based systems, Intelligent Tools for Drafting Legislation, Computer-Supported Comparison of Law*, Lelystad 1993, p. 23-30.

¹⁰ See J. Habermas, *Faktizität und Geltung*, Frankfurt am Main, 1992, especially p. 236, 360 and 430-435.

¹¹ See R. Hotz, *Strukturierung des Vorverfahrens der Gesetzgebung - Erste Schritte*

appropriate answer to a legislative problem does only partly depend on its legal quality, and, vice versa, the correct application of legal requirements does not automatically procure good or appropriate bills.¹² These differences between the legislative process (and its components) and the process of legal problemsolving amount to the conclusion that comprehensive automation of legislative reasoning, using AI-methods and AI-techniques, is (still) not possible, due to the complexity of reasoning and the structure of the knowledge involved. Existing legislative analysis and review systems illustrate that the use of the AI-approach can only meet with success in small areas of legislative drafting, like deontological consistency-checking, or determination and analysis of consequences of a draft. This conclusion does not rule out the relevance of legal computer science and AI-techniques for certain legislative activities however. It does mean, though, that in efforts to build (intelligent) tools and systems to support legislative activities, the standard approaches of legal AI- or KBS-development will not always fit the problem. Legislative support systems will therefore, I believe, have to be developed in accordance with the specific characteristics of legislative activities, using information-oriented development strategies. Treating legislative drafting in the same way as law-application (like in the case of the legislative analysis and review systems) is, as I see it, not the adequate strategy.

4. *Semi-intelligent drafting-support systems*

Legislative drafting-support systems, like the ones developed in the Netherlands, are built using more information-oriented development methods. During the development of these systems the focus is on the drafting process and drafting activities themselves, rather than on the application of a draft itself. Where legislative analysis and review systems come in when (draft) regulations have already been made, drafting-support systems function in situations where there does *not yet exist a draft*, but – for instance – only a relatively vague notion that legislation can procure the answer to a certain (social or policy) problem. Legisla-

zu einem allfälligen Einsatz von Computern bei der Schweizerischen Gesetzgebung, in: Theo Öhlinger (Hrsg.), *Gesetzgebung und Computer*, München 1984.

¹² See W. Voermans, E. Verharen, M. Fridael, Experimenteel model LEDA: kennisgebaseerd gebruik van de hypertexttechniek in een wetgevingsontwerp- en -adviesstelsel, in B.R. van der Spek, R. de Hoog (eds.), *Proceeding Kennistechnologie conferentie oktober 1992*, Den Haag 1992, p. 329-340.

tive drafting-support systems support the drafting process – and the different drafting activities therein – itself. In order to be able to perform this functionality a conceptualization of the drafting process, and drafting activities precedes the development of such systems.

The drafting process

Like in many other countries all over the world, in the Netherlands, drafting regulations is not just a matter of putting down policy choices into words. Drafting regulations involves a complex decision-making process in which many substantive choices regarding content, structure, structure elements and – eventually – phrasing and wording of a draft have to be made. Quite frequently legislative drafting even means that policy decisions have to be made or reviewed.¹³ While making these choices a lot of requirements have to be met. These requirements are not only of a homogeneous nature, comprising legal standards (e.g. constitutional standards) and aspects of legislative policy and technique, but also of a heterogeneous nature resulting from various factual conditions related to particular subject matter, or from existing policies regarding the field of the projected draft. The drafting process is a complex decision-making process which requires great skill and knowledge. In the Netherlands most of the legislative drafting is therefore carried out by specialists within the ministerial departments. To ensure the quality of their drafts, these legislative specialists – in most cases – approach the drafting process methodically. Although these approaches vary between the different departments, some general characteristics of these approaches to legislative design can be discerned. Generally speaking, these approaches consist of the following (iterative and interdependent) steps:

- 1) problem definition (including the determination of the policy and of the legislative goals of the draft solution);
- 2) problem analysis (including the determination of the relevant legal and factual contexts);
- 3) generating of alternative solutions;
- 4) analysis of the different solutions (in the light of the goals, context and effects);
- 5) selection of a solution (in the light of the goals, contexts and effects);
- 6) implementation of a solution in a legislative text;
- 7) evaluation.

¹³ See for instance the directives 6-18 of the Dutch *Recommendations for regulations*.

This model of the legislative design process (the design-step-model) does not always concur with the actual designing procedure. According to the nature of a specific project, sometimes only a few steps within this process are deemed necessary. Sometimes steps in this iterative cycle are repeated. Analytically speaking, however, this process model is empirically¹⁴ and prescriptively¹⁵ substantiated.

This analysis model also constitutes the pretext and the knowledge-backbone of two design support systems that have recently been developed for the Dutch Ministry for Education and Science (OBW¹⁶) and The Ministry of Justice (LEDA¹⁷).

The search for authoritative arguments

If we examine the legislative decision making process more closely, we see that legislative draughtsmen do not only use legislative methods to tackle legislative problems. During this process they constantly make all kinds of legislative decisions. These decisions can, as we have seen in the former paragraph, never claim to be perfect, of legally valid decisions. Legislative decisions or solutions can only claim to be 'relatively appropriate' solutions¹⁸ in view of all the (factual, societal, political, legal, and socio-economical) circumstances involved. Legislative decisionmaking is therefore not a process of application of fixed standards, but an open process in which a legislative draughtsman weighs different possible solutions in view of their relative appropriateness. The relatively best solution is the solution which is substantiated with the most convincing arguments. The most convincing arguments will be the arguments which rate very very high in the *legislative discourse* in which legislative draughtsmen find themselves together with their departmental superiors, politicians, members of parliament, interested parties, lobby groups, etc. Very convincing arguments, or *authoritative arguments*, in this discourse

¹⁴ See the results of empirical research conducted within the Dutch Ministry of Education and Science. See for the results of this research J.G.J. Wassink, *Kennistechnologie en het ontwerpen van regelgeving*, Den Haag 1992.

¹⁵ See for instance the directives 6-18 of the Dutch *Recommendations for regulations*

¹⁶ OntwerpBank Wet- en regelgeving (Regulations Design Bench) developed by the Dutch Department of Education and Science and Bolesian.

¹⁷ Prototype of a Legislative Design and Advisory system developed at Tilburg University. This system is momentarily being used at the Dutch Ministry of Justice.

¹⁸ See R. Hotz, Strukturierung des Vorverfahrens der Gesetzgebung – Erste Schritte zu einem allfälligen Einsatz von Computern bei der Schweizerischen Gesetzgebung, in: Theo Öhlinger (Hrsg.), *Gesetzgebung und Computer*, München 1984.

will be the arguments upon which almost everyone agrees. In this sense legal (e.g. constitutional) arguments or generally accepted legislative methods and techniques¹⁹ constitute strong authoritative arguments to back up a solution, while mere personal or political opinions or beliefs have a much lower ranking status. The appropriateness of a draft is largely dependant on the quality and the status of arguments which sustain the solutions held in it. In the legislative decision process legislative draughtsmen will always try to find and use the most strong argument possible to substantiate a solution and in choosing between equivalent solutions he or she will choose the solution which is backed up by the most convincing arguments within the legislative discourse. This searching for and weighing of – especially – authoritative arguments is a process which can be conceptualized, modelled and formalized.²⁰ In the Dutch LEDA system, that I will discuss here below, a modelization of this 'argumentative strategy' is represented and implemented into the system. But first let us consider the more general features and functionalities of the Dutch drafting support systems.

Semi-intelligent drafting support systems

Although the open nature of legislative problemsolving and the substantive reliance on word knowledge resist comprehensive automation of legislative reasoning, AI-techniques, I already pointed out that, can be used for the development of drafting-support systems. For instance, the two Dutch systems, LEDA and OBW, use these techniques to represent methodological knowledge according to the above mentioned design-step-model (using the frames representation formalism). In both systems the various design-steps derived from the design-step-model constitute instances within a hierarchically ordered (hypertext) network. These instances, which are visually represented in the interface as different screens (OBW) or levels within a screen (LEDA), (can) possess various attributes and methods. Sometimes a level or a screen in the network will comprise (access to) textual information about the desired level- or screen-activity, and sometimes it will contain a procedural rule (or a hierarchical hypertext-link) regarding the hierarchical place and status of the level/screen and the permitted procedures between the various levels/screens.

¹⁹ See for for instance the Dutch *Recommendations for Regulations*. These Recommendations constitute a normative corpus of guidelines and prescriptions for legislative drafting. The Recommendations are enacted in a sort of code issued by the prime minister.

²⁰ See S. Toulmin, *The Uses of Argument*, Cambridge University Press 1958.

Both systems support users by pre-structuring the drafting process and offering knowledge-based access to relevant information. They do this by using knowledge-based drafting-templates (LEDA) combined with hypertext-based information access and document-assembly (LEDA and OBW).

5. Motivation for the development of leda

Over the past ten years the Dutch government has – due to serious problems regarding the quality and effectiveness of legislation – become increasingly concerned with the quality of legislation. To improve the overall legislative quality, different policies were pursued and enacted.²¹ One of the main results of these governmental efforts and policies was the adoption of a general legislative policy, which consists of a set of measures aimed at the lasting improvement of legislative quality by setting quality criteria. A substantial part of these measures concerns the fundamental drafting stage.

The Recommendations for regulations

To guarantee attention for the legislative quality more effectively during the drafting stage, new *Recommendations for regulations* were drawn up in 1993.²² These Recommendations consist of 346 directives and guidelines regarding important drafting issues and activities. Aside from legislative technique issues, like terminology and model clauses, they also deal with policy aspects, methodological issues, procedures, structural design etc. Although they closely resemble ordinary legal rules, they are of a different nature, however. They are not always generally binding, like legal rules, but directives that can, in certain cases, be ignored if there is a good reason to do so. They constitute a mix of legal (constitutional) rules and a guidelines concerning ‘best practices and solutions’, derived from legislative experience. Besides legal rules, best practices, and legislative quality criteria, a large amount of quality safeguards are incor-

²¹ See the policy memorandum by the Dutch Ministry of Justice, *Legislation in perspective* (a policy plan for the further development of the general legislative policy, aimed at improving the constitutional and administrative quality of government policy), The Hague 1991.

²² *Recommendations for Regulations*; regulations for legislative drafting issued by the prime-minister 1992.

porated throughout the 346 Recommendations. The Recommendations therefore can be considered a voluminous 'Draughtman's handbook' dealing with every important activity within the drafting process (see the design-step-model in § 4). Related to the activities in the drafting process, the Recommendations can be categorized into the following groups:

- a) Recommendations concerning methodological and substantive issues (preparatory activities);²³
- b) Recommendations concerning the structural design of a draft (arrangement of the elements in the draft);
- c) Recommendations concerning phrasing and terminology (including the use of model clauses, model presentation-letters etc.);
- d) Recommendations concerning procedures.

The text of the Recommendations, however, is not organized along the chronological and methodological lines of the drafting process, but rather thematically in the order of diminishing abstraction. This circumstance makes it quite difficult for legislators (even experienced ones) to find their way through the new Recommendations during the drafting stage. An information system, it was felt, could be the way out of these problems. This meant the start of the LEDA-project.

The goals of the LEDA-project

The main goal in the LEDA-project²⁴ was to make the information of the Recommendations themselves accessible in concordance with the information-need during the different stages of the drafting process. A secondary goal was to make the information, referred to in the Recommendations (secondary information), available to the users. Many Recommendations, as it happens, do not prescribe what the solution has to be in a certain factual situation – as is often the case with ordinary legal rules – but rather prescribe which activity should be undertaken at a certain moment, and what kind of information is needed to be able to perform the prescribed activity. The third goal of the LEDA-project was to offer knowledge-based drafting-support on the basis of the legislative

²³ This group of Recommendations addresses questions like: what is the problem? What are the goals for a solution? Is legislation necessary to resolve the problem? If legislation is inevitable, what kind and sort of regulation will have to be drafted? Which substantive elements does it have to contain? What are the alternatives? Can it be enforced properly? etc., etc.

²⁴ Made possible by a grant of the Dutch Ministry of Justice.

knowledge within the Recommendations, pursuant to the knowledge-based access of the information from the Recommendations.

In 1993 the project resulted in the prototype LEDA-system, which is currently being tested and validated within the Ministry of Justice.

6. LEDA

LEDA is a prototype Legislative Design and Advisory system designed to offer access to the Recommendations (and secondary information) in a methodical way, concurrent with the stages of the drafting process, and through this offer knowledge-based support for the drafting activities of legislators regulated in the Recommendations. To this end LEDA contains four major (integrated) functionalities, namely:

- a) methodological support;
- b) document drafting and assembly support;
- c) knowledge-based information retrieval;
- d) legislative advice.

These functionalities are integrated throughout the system and can best be discussed by a description of the functionalities of the system's modular components. LEDA consists of two major modules:

- 1) the Preparatory (policy) Module;
- 2) the Basic Design Screen.

6.1. *The Preparatory Module*

The preparatory module in LEDA was set up to offer knowledge-based access to the Recommendations concerning substantive, methodological and structural design issues, in a way consistent with the chronology of events in the drafting stage (see for this chronology the design-step-model in § 4).

Representation

In the Preparatory Module of LEDA the different preparatory methodological activities, regulated in the Recommendations are represented in a methodological way. We have pointed out already that the Recommendations are not arranged methodologically, but thematically. In order to be able to offer methodological guidance and assistance in LEDA, we first

had to distil the methodologically important issues and activities from the Recommendations, and assess their interdependencies. To discover the methodologically important elements, we used an analysis-frame, based on a quite traditional model of the different components or elements of a norm.²⁵ Each separate Recommendation was analyzed with the following terms derived from the norm-element model:

- Recommendation (norm) object (or activity):
- Recommendation (norm) condition:
- Recommendation (norm) operator:
- Recommendations (norm) subject:

The next step was to analyze the relations between the activities we discovered. For this we supplemented the original analysis-frame with extra slots in order to be able to conclusively assess the relations between the normalized activities. The second analysis-frame looked as follows:

- Rec. object:
 - activity type:
 - activity trigger:
 - required information input:
 - information output:
- Rec. condition:
- Rec. operator:
- (Rec. subject:)

On the basis of this two-step analysis we were not only able to distil the preparatory legislative activities and their interdependencies from the Recommendations, but we were also able to construct a hierarchical frames-representation of the different drafting activities themselves, and their mutual relations. The latter operation resulted in a model which closely resembled the design-step-model discussed, consisting of seven major consecutive design steps. Within the design steps of the model, several interrelated substantive (subordinate) activities, issues and questions, regarding the preparation of a draft and the draft structure, are represented in their turn. In this way the analysis resulted in a methodological transposition of the Recommendations into a design-step model.

An obvious advantage of the frames representation in the model was, that we were able to assess the information-basis of the different activities

²⁵ See for this analysis scheme D.W.P. Ruiter, *Bestuursrechtelijke wetgevingsleer*, Assen/Maastricht 1987.

formulated in the model. This resulted in the conclusion that, although many activities were information-based, they relied on formally representable (e.g. legal) knowledge only to a very small extent. That part of the knowledge which could be formalized (e.g. the knowledge about structural design), was, together with the methodological drafting model, formalized and represented using the frames-representation formalism. Most knowledge was represented in simple frameslot procedures regarding hierarchical and referential relations and serving to address relevant blocks of information, or support limited inferencing.

Knowledge-based modelling of a hypertextnetwork

The analysis and the methodological frames representation proved that drafting activities rely strongly on information. This indicated that hypertexttechnology was a suitable candidate for the technical implementation. From a functional point of view the hypertexttechnology aims to enable users to make their way through a body of complex information in a manner that facilitates its ready appreciation or visualization.²⁶

From a more conceptual point of view, hypertexttechnology provides the means for non-linear text organization in computers by associating windows on the screen with objects in the database and providing links between those objects both graphically (as labelled tokens) and in the database (pointers).²⁷ To make this possible hypertextnetworks possess nodes and links, governing the relationships between the various nodes. Links and nodes can have a variety of properties. Nodes can, for instance, consist of (or better: correspond with database objects which 'contain') chunks of textual information, but they can also contain (a piece) of a knowledge-based template, which contains hypertextlinks in its turn.²⁸ Links can connect nodes in different ways. To establish this connection they can consist of simple or quite elaborate (knowledge-based) procedures. There are two methods for explicitly linking two points in a hypertextnetwork: by reference and by organization. The referential method supports non-hierarchical (for instance: associative) linking of

²⁶ See V. Mital and L. Johnson, *Advanced Information Systems for Lawyers*, London 1992, Chapters 7-10, p. 125-166.

²⁷ See J. Conklin, *Hypertext: an Introduction and Survey*, in *IEEE Computer*, september 1987, p. 17-41.

²⁸ See for a detailed discussion on the use of knowledge-based templates combined with hypertext techniques to enable userfriendly document-drafting and document assembly, Mital et al., *O.c.* 1992, p. 123-166 (Chapters 7,8 and 9).

nodes. The organizational method on the other hand explicitly creates hierarchical connections, by connecting a parent node with its children, thus establishing a strict tree subgraph within a hypertext network graph.

Without a further discussion of all the different potent possibilities of hypertexttechnology, it will be evident that it was not hard to transpose the methodological frames-representation (within our design-step-model) into a hypertextnetwork. We used the frames-representation specifically to model the hypertextnetwork to our needs. For instance: in order to model the hierarchical links in the hypertextnetwork, we used the methodological knowledge about drafting activities represented in the frame-network. In the same way we modelled the network's referential links and inference procedures. This enabled us to create a hypertextnetwork which does not only provide very flexible information linking, but which also dynamically produces knowledge-based templates, and substantively as well as methodologically supports legislative drafting.²⁹ Following the same principal we organized the linked information in the hypertextnetwork. Following the different nodes in the network, the user will be confronted with different kinds of (authoritative) arguments ranking from high valued arguments to lower ranking arguments. The Recommendations themselves tell the user if an argument is authoritative or not.

For the experimental realization of the system, we initially used a development tool called Toolbook (an MS-Dos version of Hypercard). The LEDA-system in its present form is however developed in Borland C++, using the object-oriented programming paradigm.

Functionalities of the Preparatory Module

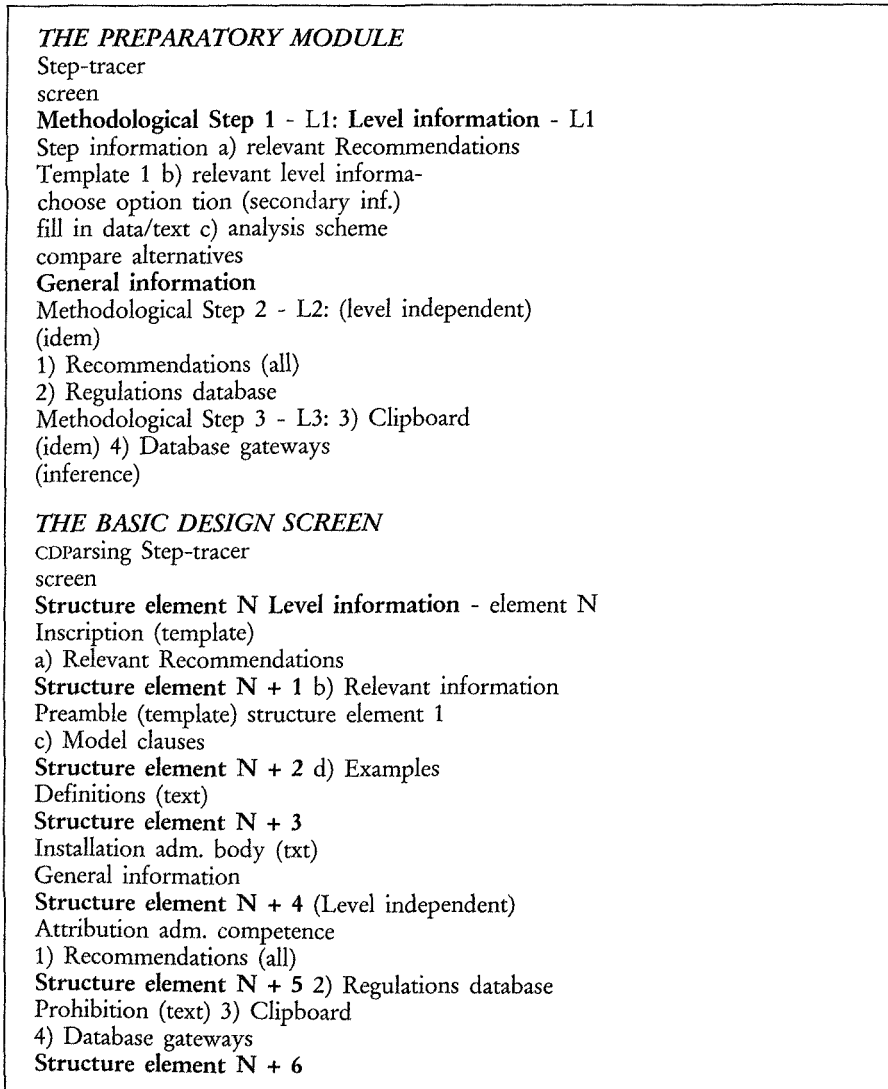
The Preparatory Module (PM) combines the functionalities of a hypertextsystem with a knowledge-based (KB-) template system. The hypertext-based PM of LEDA permits the user not only to draft a preparatory document (e.g. a policy memorandum), but also supports the creation of a skeletal form for a KB-template, to be used for the actual structural design and formulation of a draft (Basic Design Screen). To this end the Preparatory Module guides the user through a hypertextnetwork of semantic hierarchical and referential links. To offer guidance, the hypertextnetwork of the PM is divided into different levels, corresponding with

²⁹ See W. Voermans, Computer-aided legislative design: worth while the effort?, in C. van Noortwijk, A.H.J. Schmidt, R.G.F. Winkels, *Legal knowledge based systems*; Aims for research and development, Lelystad 1991, p. 87-96

the different methodological steps of the design-step-model derived from the Recommendations. The levels in their turn serve as a checklist, expressing important points of attention regarding methodological and substantial aspects and the structural design of a draft.

A look at the systems' interface architecture (which closely resembles the functional architecture) may illustrate these features:

FIGURE 1. LEDA: Interface-architecture (functional modules and components)

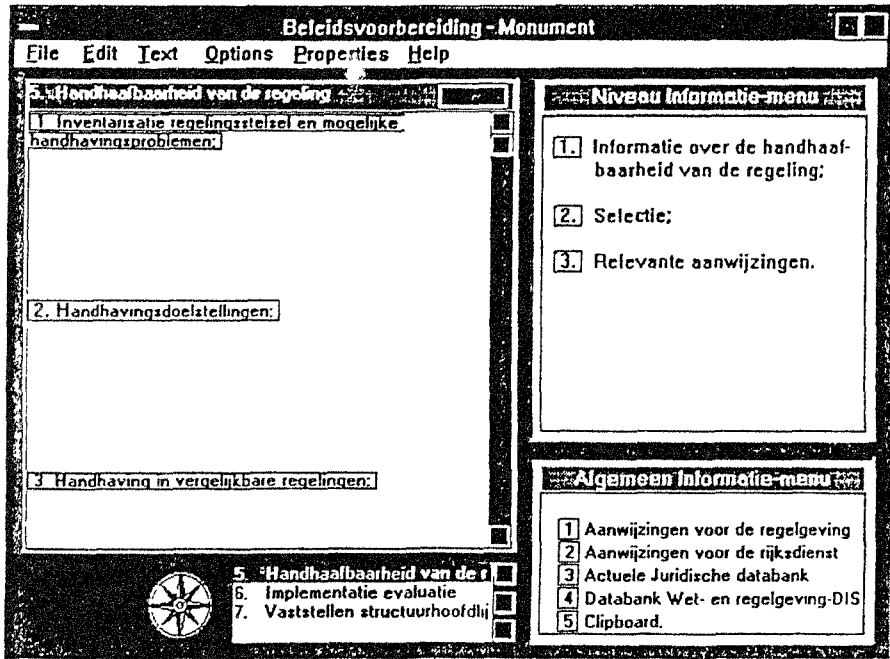


As the figure shows, the Preparatory Model consists of various methodological and consecutive levels (dotted lines on the left hand side). These methodological levels are referentially linked with level information (box at the upper right hand side). The level information component consists of (access to) the relevant Recommendations, access to relevant secondary information (as referred to by the relevant Recommendations), and a graphic template-scheme for user-analysis of certain options. Level information changes according to the level which is active (i.e. the level in which the user is working).

The methodological levels themselves consist of fields containing information (about what is to be done within a certain level) and knowledge-based templates. The level-template-documents which mainly serve to insert (or draft) text, also support the identification of important sub-items, and the choice between options. Both on the basis of the choice of the user and automatic analysis of text-input in the template, the system makes inferences regarding the arrangement of levels further down the network's path (e.g. the arrangement of the levels in the Basic Design Screen). From the point of view of the user, the levels form an interactive word-processor which provides methodological guidance and provides relevant (semantically interlinked) information, in the form of authoritative arguments.

The user may progress randomly through the level-structured hypertextnetwork. This fundamental openness of the system is necessary as the user-legislator is always free – when drafting a legislative text without the use of the system – to deviate from the Recommendations themselves whenever there is a good reason.³⁰ To accommodate reluctant users, there is even a possibility of to shut down the levels altogether. What remains is a word-processor linked to information in a single default-information level explaining the methodological approach of the Recommendations, and providing (links to) the relevant Recommendations and secondary information. To prevent getting lost in the hypertextnetwork, user-guidance is provided by the levels themselves, together with easy backtracking procedures and a step tracer, which consist of a major and minor active compass which visibly records the path hitherto followed in the network. On top of this the PM is provided with a General Information-component to offer non-hypertextual access to various internal or external databases. Users can retrieve text from these databases while working in the different levels. The text in the internal databases, however, is hypertextually linked.

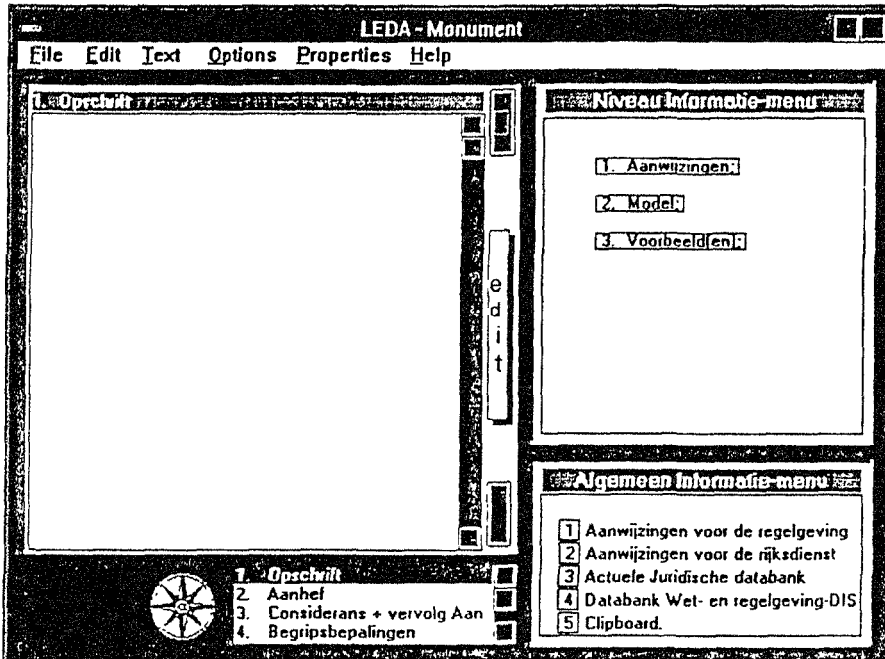
³⁰ See directive 5 of the *Recommendations for regulations*.

FIGURE 2. *The PM-interface*

6.2. *The Basic Design Screen*

The Basic Design Screen Module (BDS) is developed and structured in a way very similar to the Preparatory Module. Like the PM it consists of a level structure, linked with level information. The levels (see the dotted line in the BDS-module of figure 1) contain templates mainly consisting of free-text fields, which allow system supported insertions (e.g. of model clauses or examples). The templates within the levels of the BDS however do not express points of attention with regard to the preparation and structural design, but important phrasing, terminology and terminology-related (substantive) issues regarding the structural elements of a draft. The arrangement of the levels in the BDS is both based on knowledge (gained from the Recommendations) and knowledge-based inferences made by the PM module. The BDS itself can be regarded as one large knowledge-based template which is shaped and directed by the PM. The BDS represents the preferred structure of a draft, modelled to the needs of the user.

Like the PM the BDS has a very open structure: the user may progress

FIGURE 3. *The BDS-interface*

randomly, do away with the levels altogether and receive default-information, and delete or add certain levels. The user-guidance function is similar to the one in the PM. The BDS has, however, one distinctly different feature compared to the PM. It possess a conceptual dependency parser.³¹

The CD parser

When a user has finished the drafting of a text (within a certain level of the BDS), he may be interested to know whether he has overlooked a relevant Recommendation. In other words did he/she overlook a high an authoritative or high ranking argument? To accommodate this interest LEDA possesses a conceptual dependency parser (CDP). This CDP automatically analyzes (parses) the user-inserted text in

³¹ See R.L. Schank, and C.K. Riebeck, *Inside Computer Understanding*, Hillsdale, NJ, 1981.

a BDS level and dynamically creates links to a particular concept in the database or a the text of a Recommendation if the text-analysis indicates the relevance. To be able to do this the CDP not only detects key-words and key-word-combinations and matches them with patterns in the database (stringmatching), but also analyzes concepts in text sentences (by using the linguistic conceptual dependency method and matches them with concepts in the database (automated conceptual information retrieval).³² Concepts in a user-inserted text within a level are analyzed using the norm-element model.³³ This norm-element-model distinguishes between four major concepts within a sentence expressing a norm,³⁴ namely:

1. a (deontic) *normoperator* (expressing in a natural language terms an Obligation, a Permission, a Prohibition or a Command. Grammatically speaking this operator will always be a conjugation of a verb);
2. a *normobject* (grammatically: a set of substantive and/or adjective nouns, conjugated verbs and conjunctions constituting the direct object-clause of the sentence);
3. a *normsubject* (grammatically: a pronoun or a substantive noun combined with a definite or indefinite article constituting the subject of a sentence);
4. a *normconditon* (grammatically: verbs, (pro)nouns, conjunctions and articles constituting the adverbial clause of a sentence).

Natural language-analysis on the basis of this norm-element-model is possible because the concepts in the database, which are modelled as frames on the basis of knowledge derived from the Recommendations, have slots corresponding to this norm-element-model. An example may illustrate.

Suppose a user inserts the following text:

“Our minister can set rules regarding the administration of licences.”

The CD-parser in LEDA will match this piece of natural language with concepts in the database. Two relevant concepts will be found. First of all the concept (better pattern-concept):

³² See note 31.

³³ See D.W.P. Ruiter, *Bestuursrechtelijke wetgevingsleer*, Assen/Maastricht 1987.

³⁴ The beginning and the end of a norm sentence do not necessarily concur with the beginning and the end of natural language sentences.

Frame-Legislative Terminology

type:	indication-minister
natural language	
indicator:	minister*, our minister*
Recommendations:	30,69,73,74,75
related frames:	delegation-minister, subscription-minister, attribution of administrative authority-minister
operation:	if-indicator → link-indicator → show link in text → on demand show corresponding leaflet {indication-minister}
leaflet indication-minister:	<i>“Terminology-Indication of ministers</i>

The following Recommendations concerning the way in which minister are to be indicated in legislative texts are most likely to be relevant:

73 (indication of a minister)

74 (indication of more than one minister)

75 (...)

See also:

Delegation of regulatory authority to ministers (30,69)

Subscription & ministers

Attribution of administrative authority & Ministers”

NB. The italicized texts are hypertextlinks to the text of a Recommendation or to leaflet-texts in another concept in the database.

The second concept that will be found reads as follows:

Frame-Delegation

type:	regulatory authority
operator-indicator:	can, may, set, sets, regulate, regulates (etc.)
object-indicator:	rule, rules, set... rules (etc.)
subject-indicator:	Minister, ministers, government
condition-indicator:	{language}
operation 1:	if-operator-indicator and object-indicator and subject-indicator → link indicators → show link in text → on demand show corresponding leaflet A {delegation minister}

operation 2: if-operator-indicator and object-indicator → link indicators → show link in text → on demand show corresponding leaflet B {delegation}

leaflet A-delegation-minister:

“Delegation of regulatory authority to ministers

The following Recommendations are most likely to be relevant:

30 (Delegation of regulatory authority to ministers)
69 (Terminology ministerial delegation)

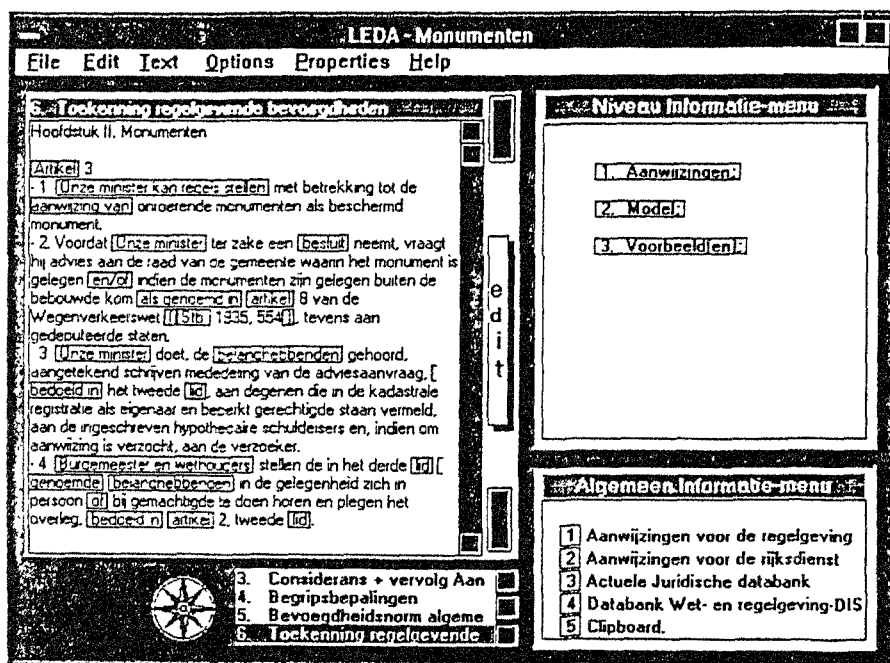
See also:

Indication of ministers”

leaflet B-delegation: *“Delegation of regulatory authority*

The text seems to indicate delegation of regulatory authority. To whom is this authority to be delegated?

FIGURE 4. *The CD parser at work*



government (see: delegation of regulatory authority to government)

a minister (see: delegation of regulatory authority to ministers)

(etc. ... p.m.)

a mouse-click on the italicized text will indicate your choice”

NB. The italicized texts are hypertextlinks to the text of a Recommendation or to leaflet-texts in another concept in the database.

This – due to the inherent limits of this paper – briefly illustrated form of conceptual dependency parsing combined with automated conceptual information retrieval is very powerful because both the concepts in the level-related text and the concepts in the database can be quite accurately defined. For the user it supplies a powerful intelligent Recommendations check of his natural language draft.

7. Conclusion

The information-oriented approach to the development of practical legimatic systems seems to pay off. OBW and LEDA are – though still on a modest scale – both being used in the actual departmental drafting process in the Netherlands. By pre-structuring the draft-process and offering knowledge-based access to relevant (authoritative) information LEDA (as well as the OBW-system of the Dutch Ministry of Education and Science) are first steps on the way to really intelligent drafting support systems. In a number of ways these semi-intelligently support different aspects of the complex task of drafting a bill.

The modest success of the information-oriented approach to legimatics does however not mean that the AI-approach isn't profitable.

The way I see it, is that AI-based tools will, in the near future, dramatically improve the functionality and the quality of existing legimatic information systems. The AI-approach bears a lot of promise when it is combined with the results of the information-oriented approach. Legimatic AI-tools, suitable for consistency checking and considering the deontological consequences of a draft, will not only be able to improve the quality of drafting support systems, they can also initiate a new way of thinking about legislative quality and kick off new approaches to legislative drafting.

This combination of drafting support and purely AI-based legislative analysis and review systems is, however, for the moment, blocked by the necessity of – user unfriendly – complex knowledge representation and formalization of natural (draft) language to accommodate analysis and review systems. There may be a way out of these problems, however: one day conceptual dependency parsing of natural language may well provide the solution, by allowing for automatic knowledge representation and formalization of knowledge-concepts, contained in the natural language of a draft.