CASE reading, structuring and analyzing decisions by judges

Muntjewerff, A.J.

Publication date
2013

Document Version
Final published version

Published in
European Journal of Law and Technology

Citation for published version (APA):
CASE reading, structuring and analyzing decisions by judges

Antoniette J. Muntjewerff

CASE reading, structuring and analyzing decisions by judges

1. INTRODUCTION

The law that applies in a legal system such as the Dutch legal system consists of general rules that are determined or acknowledged by authoritative bodies. The two most important authoritative bodies within the Dutch legal system are the legislature and the judge. While it is obvious the legislature proposes general rules that apply in individual cases, she has to construct and specify them. A legal solution based on the facts of the case and the applicable legal rules. In the majority of cases that are brought to court, a judge formulates a decision that applies only in the case at hand. These decisions do not add to the body of applicable rules in the legal system. However, in cases where a judge has to construct an applicable rule, before being able to decide the case on the basis of the applicable legal rules. The role constructed by the judge to decide the case, to add the body of applicable rules in the legal system. Legal practitioners and legal instructors need to have knowledge of the general rules that apply in the legal system. This involves both knowledge of the legislation and knowledge of the decisions by judges that form as general rules of law. The problem of legal problem solving is how difficult it is for novices to learn the legal reasoning. The HYPATIA design approach starts with (re)constructing explicit models of legal knowledge and legal reasoning. In this reconstruction process, two components are distinguished. What is structuring and analyzing a decision? In order to answer this question and to design an environment to support law students in finding, reading, structuring and analyzing a decision, a decision in order to determine in what way this decision adds to the body of applicable legal rules. CASE can be used by law teachers to mirror selected decisions by judges and to add keywords and by law students to search, structure and analyze decisions by judges.

Keywords: Instructional Design, Coaching Systems, Legal Problem Solving

2. PRINCIPLED AND STRUCTURED DESIGN APPROACH

The HYPATIA project (Muntjewerff 2002a, Muntjewerff 2002b) aims at designing and developing an instructional design approach as described in the HYPATIA project (Muntjewerff 2003b). The approach taken in PROSA is reusable for a variety of applications for learning the law. Instructional design decisions are made on the basis of a global theory or learning and instruction. In this way the design process will result in a coherent and consistent instructional model. It finally indicates that electronic multimedia are evaluated extensively (developmental testing and field testing). The design approach was found and used successfully in the construction of an instructional environment for learning to solve legal cases.

PROSA: The approach taken in PROSA is reusable for a variety of applications for learning the law. The legal case-solving research within HYPATIA has been realized and reported in detail (see, for instance, Muntjewerff 2000, Muntjewerff and Groothuisen 1999, Muntjewerff 2003c).

3. ANALYSIS

What is structuring and analyzing a decision? In order to answer this question and to design an environment to support law students in finding, reading, structuring and analyzing decisions to indicate and understand the legal meaning of a decision, it is necessary to analyze the task.

The HYPATIA design approach starts with (re)constructing explicit models of legal knowledge and legal reasoning. In this way construction process, two components are distinguished.

(1) A theoretical component of exploration, conceptualization and specification of legal knowledge and legal reasoning, is explicit models of legal knowledge and legal reasoning. Two perspectives are taken within this approach: a legal perspective and a knowledge engineering perspective.

(2) An empirical component where empirical studies are carried out to acquire insight in the way legal practitioners and legal instructors handle legal knowledge in general and in carrying out specific legal tasks. In this case, law students are studied to see how they handle and use legal knowledge to perform a specific legal task and what difficulties they experience. The legal instructors were examined on the task of reading and comprehending decisions: all describe a series of steps to be taken by the student when reading a decision to determine the legal significance (Ros 2003, Evers et al. 1996, Franke et al. 1995, Herder and Vink van den 1998, Jonasen 1998, Schouten 1998, te Heesen and Waddewaard 2002, Muntjewerff 2005). However, merely instructing a method does not work for novices (see for details Muntjewerff 2002). This is partly due to the fact that instructing a method is a problem in itself, as it is difficult the transition of knowledge into skills. This requires the development of instructional materials that are suitable for novices.

The legal instructors were examined on the task of reading and comprehending decisions: all describe a series of steps to be taken by the student when reading a decision to determine the legal significance (Ros 2003, Evers et al. 1996, Franke et al. 1995, Herder and Vink van den 1998, Jonasen 1998, Schouten 1998, te Heesen and Waddewaard 2002, Muntjewerff 2005). However, merely instructing a method does not work for novices (see for details Muntjewerff 2002). This is partly due to the fact that instructing a method is a problem in itself, as it is difficult the transition of knowledge into skills. This requires the development of instructional materials that are suitable for novices.

Reading and understanding a decision is not a trial activity. Observations with first year law students reading decisions showed that they experience difficulties with seeing through the composition of the decision, with reconstructing the argument structure and with determining the legal significance of the decision. These difficulties are not the result of the extent of the text. Not to the fact that the decision contains many references, both explicit and implicit, to regulations, other decisions and concepts. The fact that a decision has a structured is often not supported by recognizable cues or elements in the text. This does not mean, however, that students do not notice the presence of these references. Students are often aware of the references that are made in the text.
All of this means that the student has to reconstruct the process and the product which involves keeping track of intermediate results. To support the student in performing these tasks, the following remediates are proposed:

Present the student a structure to help her to reconstruct the decision,

Support the management of information and

Engage the student in structuring and analyzing the decision by having her actually carry out these tasks.

This is realized by presenting the student with both the full text of the decision and a framework which visualizes the elements in a decision necessary to reconstruct the decision in order to determine the legal significance of the decision. There are no applications available that support law students in structuring and analyzing a decision using the Continental legal system. For the Anglo-American legal system, the CASTO application is available (Allen 1985). In CASTO the student is trained to construct arguments with cases.

4. ARCHITECTURE AND MODULAR DESIGN

The aim of the CASE project is to realize an environment in which law students are supported in structuring and analyzing a decision. The issue is to realize that both the decision at hand has to be presented to the student, as well as in the framework for analysis. The student must be able to select text fragments from the decision and paste them within the correct cell in the relevant table in the framework. Since finding cases is also part of the training of law students search facilities have to be available in the environment. The functionality of searching for a decision is implemented in the first module. The functionality of structuring and analyzing a decision is implemented in the second module. Other basic requirements are maintenance and accuracy. It should be possible to make changes to the system and its content without impairing user interface. Errors in the system and its content should be easily traceable and correctable. It must be possible to add and delete content without causing problems elsewhere in the system. Transparency of the architecture and tools are therefore design goals, as is the facilitation maintenance.

The system has functions for adding decisions, adding key words to decisions and preparing decisions for analysis. System functionalities are attributed to a user on the basis of user-status: administrator, editor, teacher or student. The database module builds the database and allows for searching and retrieval of cases and allows teachers to prepare cases for use in the analysis module. Students can see the database module to locate cases on the basis of key words and/or full text search to find specific decisions. When the student wants to structure and analyze a decision she can select one of the reported decisions. The decision and the analyzing framework are then made available to the student. The student can start structuring the decision by selecting text fragments in the decision and pasting them in the correct part of the frame.

5. PLATFORM AND IMPLEMENTATION

CASE is implemented using a web-based server-side application model. The user interacts with the system using a standard web browser, such as Netscape Navigator, Apple Safari or MS Internet Explorer. CASE is developed using Open Source Software, MySQL (4.1.14) and PHP (4.2.2) and JavaScript. The MySQL database holds contain a number of tables, the most prominent ones being a text fragment table, a solution table and a table storing the student's activities. Each table is two-dimensional and contains a small number of cells, e.g. facts as presented by the initiator, and facts presented by the opponent.

A small number of simple functions are implemented using client-side JavaScript. CASE offers extensive support for administrative, editing, browsing, tracking- and educational tools. Using the same portal, administrators can add, remove and change users and cases; editors can add keywords to cases and prepare the solution framework of a case for use; teachers can use the interface to track the results of students, previewing the solution framework and for browsing and searching the database; and students can browse and search the database, and test their analysis skills.

The search engine allows for both Boolean keyword- and free text search in combination with metadata fields such as: date, name, court etc. The principal concept in CASE is that a precedent can be seen as an ordered set of text fragments, each of which can be labeled according to their position in the solution framework. The student can select a text fragment and place it in a specific position within the solution framework. Text fragments can be as short as a single sentence, but more often, they are as long as a paragraph. The text fragments are stored in a database along with metadata such as a reference to their position in the solution framework. Although a text fragment as described in the basic building block, these fragments can form one or more sub-fragments (such as single words) which can also be selected by the student. For instance, the text fragment:

'The next step is the preparation of the decision for use. The prepare tool offers an interface that mimics the regular screen. Here she fills in a few facts about the decision (name, publication date, court etc.) and with copy-and-paste actions she adds the text of the decision to the database.'

contains the sub-fragment 'The next step is the preparation of the decision for use. The prepare tool offers an interface that mimics the regular screen. Here she fills in a few facts about the decision (name, publication date, court etc.) and with copy-and-paste actions she adds the text of the decision to the database.'

The metadata editor interface is used to add or change metadata of a decision and, more importantly, to add new keywords or remove existing ones. After completing this procedure, the decision can be searched for using the search interface.

The metadata editor interface is used to add or change metadata of a decision and, more importantly, to add new keywords or remove existing ones. After completing this procedure, the decision can be searched for using the search interface.

The metadata editor interface is used to add or change metadata of a decision and, more importantly, to add new keywords or remove existing ones. After completing this procedure, the decision can be searched for using the search interface.

The next step is the preparation of the decision for use. The prepare tool offers an interface that mimics the regular structuring and analysis interface: the editor needs to place pieces of text in the correct position within the correct frame. Since finding cases is also part of the training of law students search facilities have to be available in the environment. The functionality of searching for a decision is implemented in the first module. The functionality of structuring and analyzing a decision is implemented in the second module. Other basic requirements are maintenance and accuracy. It should be possible to make changes to the system and its content without impairing user interface. Errors in the system and its content should be easily traceable and correctable. It must be possible to add and delete content without causing problems elsewhere in the system. Transparency of the architecture and tools are therefore design goals, as is the facilitation maintenance.

The system has functions for adding decisions, adding key words to decisions and preparing decisions for analysis. System functionalities are attributed to a user on the basis of user-status: administrator, editor, teacher or student. The database module builds the database and allows for searching and retrieval of cases and allows teachers to prepare cases for use in the analysis module. Students can see the database module to locate cases on the basis of key words and/or full text search to find specific decisions. When the student wants to structure and analyze a decision she can select one of the reported decisions. The decision and the analyzing framework are then made available to the student. The student can start structuring the decision by selecting text fragments in the decision and pasting them in the correct part of the frame.

6. SESSION WITH CASE

As mentioned above, CASE distinguishes four types of user: administrators, editors, teachers and students. User rights are distributed in an incremental fashion in CASE: this means that a teacher has access to both student- and teaching facilities; an editor has access to editing-, teaching- and student facilities; and the administrator user has rights to do everything the other users can, plus adding, removing and changing users, and removing cases from the database. This section describes a typical process from preparation to analysis of a case. After login, the editor is presented with a menu containing multiple options. Since the editor recently came upon a decision relevant for law students, she decides to add it to the CASTO database. The editor’s next goes access to the add decision screen. Here she fills in a few facts about the decision (name, publication date, court etc.) and with copy-and-paste actions she adds the text of the decision to the database.

Next, she visits the metadata editor (see figure 1).

The metadata editor interface is used to add or change metadata of a decision and, more importantly, to add new keywords or remove existing ones. After completing this procedure, the decision can be searched for using the search interface.

The next step is the preparation of the decision for use. The prepare tool offers an interface that mimics the regular structuring and analysis interface: the editor needs to place pieces of text in the correct position within the solution framework (see figure 2).

Figure 1: Metadata editor
Where the regular interface checks whether the correct text is in the correct position by consulting the database, the prepare tool writes the action of the editor to the database. Note that the editor does not have to add feedback to the database. Feedback is provided to the student in a case-independent way. When the teacher only wants part of the text fragment to be part of the solution, the editor can simply mark these smaller parts. This results in a text fragment with color coded text fragments that can be placed in the solution table (e.g. Mr Funke in figure 3). After the editor has finished the above steps, the decision is ready for use by both teachers and students. The teacher is not allowed to change the information or the solution framework of a decision. However, he can add students to the CASE user database, and preview the correct answers (the prepared solution framework) for each decision.

More importantly, the teacher has access to a student tracking facility to analyze student behavior. This way the teacher can determine whether a student came to his or her end-result by simply trying every option, or by purposefully placing fragments in the solution framework.

Students can search the decision database using the search interface (see figure 3). This interface allows for metadata search - i.e. on publication date, publication place, court type, court location - but also supports Boolean keyword search and Boolean full text search. The student can also browse through all decisions in the database. The search result page offers support for associative search because key words and other attributes of the cases found are shown. The student can click on any of these to start a search on this attribute. Thus, for example, searching on all decisions with the same key word of one of the decisions that were found by the original search is done by simply clicking on that keyword in the results page. From the same page, the student can print a decision or open it for structuring and analysis.

The structuring and analysis interface, shown in figure 4, is divided into three frames. The left frame shows all text fragments of the decision at hand. The top right frame contains the tables of the solution framework. The bottom right frame provides feedback to the student's actions. A text fragment is placed in a cell of the solution framework by first selecting the cell, and then selecting the fragment to fill the cell. Once placed, the application will check the combination of cell and fragment and provide a feedback message from the database in the feedback frame. Text fragments can be removed from a cell by clicking the ‘x’ button in the table. Once the student has placed all correct fragments in a specific table, she is notified of this through the feedback frame.

### 7. SUMMARY AND FUTURE WORK

Learning the law involves reading, structuring and analyzing decisions to be able to indicate the legal significance of the decision. Law students experience difficulties especially with determining what the decision adds to the body of applicable rules in the legal system. With the current curriculum there is not enough time to read and analyze decisions in the presence of a teacher who could provide immediate feedback. Law students are also not presented with models that may guide them in the process of reading and analyzing decisions. In learning the law it is essential to know how to structure and analyze a decision.

CASE was designed to present the law student with an instructional environment in which she is able to analyze a decision in such a way that the structure is made explicit and the legal meaning can be extracted. CASE is implemented as a web-based server-side application model using open source software. CASE is easy to maintain and re-use and can be made available in different languages. At the moment we are working with a CASE 2.0 version in which some administrative functionalities have been improved. The claim that law students are supported by CASE in structuring and analyzing a decision in such a way that they are able to grasp the legal significance of the decision should be tested in depth. A first small test with first year law students shows that it may be necessary to extend CASE with a knowledge model (Muntjewerff, 2012b).

### 8. ACKNOWLEDGEMENTS

The CASE project was partially funded by the Association of Universities in the Netherlands (VUAN).

### BIBLIOGRAPHY


Bos, A.M (2010), Jurisprudentieinstructie, in Encyclopaedia der Rechtsgeneeskundige Vraagstukken (Amsterdam: Universiteit van Amsterdam).


Assistant J. Muntjewerff is Assistant Professor of General Legal Theory at the Faculty of Law at the University of Amsterdam. She has a Masters in Social Science, a Masters in Law and a PhD in Computer Science & Law. She studied Computer Science (more specifically Artificial Intelligence).