CASE reading, structuring and analyzing decisions by judges

Muntjewerff, A.J.

Published in:
European Journal of Law and Technology

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
CASE reading, structuring and analyzing decisions by judges

Antoinette J. Muntjewerff

Abstract: It is essential for law students to acquire knowledge about the applicable general rules of law. A major source of applicable rules within the Dutch legal system is, next to statutes, the category of decisions by judges. Law students experience difficulties with finding, reading and comprehending these decisions. CASE (Case Analysis and Structuring Environment) is an instructional environment developed to support students in finding, reading, structuring and analyzing decisions by judges. CASE provides the student an environment in which he can practice finding, reading, structuring and analyzing 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 help selected decisions by judges and to add key words and by law students to search, structure and analyze decisions by judges.

Keywords: Instructional Design, Coaching Systems, Legal Problem Solving

1. INTRODUCTION

The legal system of the Netherlands, as many other legal systems, is based on a combination of statutes, case law and regulations. The legal sources that were examined to model 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 (Bos 2003, Eemeren van et.al. 1996, 2006). Furthermore it is desirable for students to be able to look for help at any time during the process. These problems are acknowledged by law teachers. However, there is no material available to help students to overcome these difficulties. HYPATIA aims to fill this gap by developing electronic materials to offer individualized support and feedback. It can have the capacity to adapt to the individual student’s performance and, last but not least, may support the management of information.

CASE (Case Analysis and Structuring Environment) is an instructional environment developed to support students in finding, reading, structuring and analyzing decisions by judges in order to determine in what way this decision adds to the body of applicable legal rules. CASE provides the student an environment in which he can practice finding, reading, structuring and analyzing a decision in order to determine in what way this decision adds to the body of applicable legal rules.

2. PRINCIPLED AND STRUCTURED DESIGN APPROACH

The HYPATIA project (Muntjewerff 2010, Muntjewerff 2008) aims at designing and developing new additional electronic materials for law students to learn the law. Law students experience difficulties in acquiring legal knowledge and in using this knowledge. The knowledge engineering perspective is used to construct models at a high level of explicitness as they have to be executed by a computer (see, for example, Haan den and Sartor 1999, Breuker and Van de Velde 1994, Valente 1995). This representation of legal knowledge is of great importance; legal problem solving is hardly possible without understanding the legal sources, their history and background.

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, it is necessary to understand the task.

The HYPATIA design approach starts with (re)constructing explicit models of legal knowledge and legal reasoning. In this way a computer can be used to assist law students in finding, reading, structuring and analyzing a decision in order to determine in what way a decision adds to the body of applicable legal rules. CASE can be used by law teachers to help selected decisions by judges and to add key words and by law students to search, structure and analyze decisions by judges.
6. A SESSION WITH CASE

'This indeed is one of the parties in the dispute, but unfortunately it is not the opponent.' To get a basic idea of the functionality of the system we now describe a session with CASE.

A small number of simple functions are implemented using client-side JavaScripts. CASE offers extensive support for administrative-, editing-, browsing-, tracking- and educational tasks. 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 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 her status: administrator, editor, teacher or student. The database module holds the decisions and allows for search 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 and pasting these in the correct part of the frame.

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 place in the solution template. 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. Although a text fragment as described is the basic building block, these fragments can have one or more sub-fragments (such as single words) which can also be selected by the student. For instance, the text fragment Op de beroep van Ronald G, geboren te Amsterdam op 6 aug. 1954, woonde te Amsterdam, rog. van casatie tegen ong. or 973 gewooven arrest van het Hof te Amsterdam van 12 dec. 1977, waarbij in hoger beroep een vanmiste de RBK contains the sub-fragment 'Ronald G', the accused. In some cases the student needs to select the whole sentence, and in others only the sub-fragment. The solution framework consists of a number of tables, such as parties, facts, claim and the argument structure before the Supreme Court. Each table is two dimensional and contains a small number of cells, e.g. fact or claim. As the student is presented with a menu containing multiple options, she decides to add it to the CASE database. The editor's menu gives access to the add decision screen. Here she fills in a few facts about the decision (name, publication date, court etc.) and with copy-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 aim of the CASE project is to realize an environment in which law students are supported in structuring and analyzing a decision. 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 her status: administrator, editor, teacher or student. The database module holds the decisions and allows for search 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 and pasting these 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 (4.2.6) and JavaScript. The MySQL database holds a number of tables, the most prominent ones being a text fragment table, a solution table and a file storing the student's activities. MySQL's primary component is the server-side application implemented in PHP (4.2.6). The application handles form processing, storage and retrieval of information from the various tables in the database and generates the HTML pages that are output to the user.

A small number of simple functions are implemented using client-side JavaScripts. 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 place in the solution template. 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. Although a text fragment as described is the basic building block, these fragments can have one or more sub-fragments (such as single words) which can also be selected by the student. For instance, the text fragment Op de beroep van Ronald G, geboren te Amsterdam op 6 aug. 1954, woonde te Amsterdam, rog. van casatie tegen ong. or 973 gewooven arrest van het Hof te Amsterdam van 12 dec. 1977, waarbij in hoger beroep een vanmiste de RBK contains the sub-fragment 'Ronald G', the accused. In some cases the student needs to select the whole sentence, and in others only the sub-fragment. The solution framework consists of a number of tables, such as parties, facts, claim and the argument structure before the Supreme Court. Each table is two dimensional and contains a small number of cells, e.g. fact or claim. As the student is presented with a menu containing multiple options, she decides to add it to the CASE database. The editor's menu gives access to the add decision screen. Here she fills in a few facts about the decision (name, publication date, court etc.) and with copy-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).
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 colour coded sub 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 keyword of one of the decisions that were found by an initial 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 students' actions. A text fragment is placed in a cell of the solution table by first selecting the cell, and then selecting the fragment to fill this 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. While the current curricula there is not enough time to read and analyze decisions in the presence of a teacher who may 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 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. An initial test with first year law students showed 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 (VSNU).