

## Tilburg University

### Case Analysis and Storage Environment

Muntjewerff, A.J.; Jordaans, A.; Hoekstra, R.; Leenes, R.E.

*Published in:*

Legal knowledge and information systems: Jurix 2003: The Sixteenth Annual Conference

*Publication date:*

2003

[Link to publication in Tilburg University Research Portal](#)

*Citation for published version (APA):*

Muntjewerff, A. J., Jordaans, A., Hoekstra, R., & Leenes, R. E. (2003). Case Analysis and Storage Environment: CASE. In D. Bourcier (Ed.), *Legal knowledge and information systems: Jurix 2003: The Sixteenth Annual Conference* (pp. 1-10). IOS Press.

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# Case Analysis and Storage Environment

## CASE

Antoinette J. Muntjewerff\*    Astrid Jordaans<sup>†</sup>    Rinke Hoekstra\*    Ronald Leenes<sup>‡</sup>

*\*Faculty of Law, Universiteit van Amsterdam  
Amsterdam, The Netherlands*

*{muntjewerff, hoekstra}@lri.jur.uva.nl*

*<sup>†</sup>Open Universiteit, Heerlen, The Netherlands  
astrid.jordaans@ou.nl*

*<sup>‡</sup>Concept & Code, Deventer, The Netherlands  
ronald.leenes@concept-en-code.nl*

**Abstract.** It is essential for law students to learn about the applicable general rules of law. A major source of applicable rules within the Dutch legal system is the category of decisions by judges. Law students experience difficulties with reading and comprehending these decisions. CASE is proposed to support students in structuring and analysing decisions. CASE stands for Case Analysis and Storage Environment. CASE presents the student an environment in which she can practice with structuring and analysing a decision in order to determine in what way this decision adds to the body of applicable rules. The application can be used by law teachers to store decisions and to add key words and by law students to search, structure and analyze a decision.

## 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 legislator and the judge. While it is obvious the legislator determines rules that apply in general, this is more complicated with judges. A judge has to decide in individual cases, she has to construct a legal solution based on the facts of the case and the applicable legal rules. In the majority of cases that come before the court, a judge formulates a decision that applies only to 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 first has to construct an applicable rule, before being able to decide the case on the basis of this rule, we have a different type of decision.

The rule constructed by the judge to decide the case, may add to the body of applicable rules in the legal system. Legal practitioners and legal scientists 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 function as general rules of law. Law students preparing themselves for the legal profession of course also need these kinds of knowledge. They have to acquire knowledge about the role of decisions by judges in the legal system, and they need to understand the two categories of decisions by judges. A student has to have knowledge about where to look for decisions of the second category, understand the structure of decisions and learn to determine what makes a decision relevant to the body of applicable rules in the legal system. Legal education primarily aims at acquiring insight in the legal sources, their history and background. This basic knowledge is of great importance;

legal problem solving is hardly possible without an understanding of the legal knowledge. To illustrate the use of this knowledge in practice, teachers work through decisions as examples. However, it is difficult, if not impossible, to learn by explanation or by imitation alone. A more effective way to obtain expertise (skill) is by actually performing the task, i.e. students should do the exercises, while the teacher provides feedback on their solutions. Not only feedback on the solution provided by students is important. For effective learning, also the solution process should be monitored and provided with feedback. Furthermore it is desirable for students to be able to ask for help at any time during the process. They should also be able to practice over and over again. An ideal situation would have a teacher available for every student, monitoring the student while practicing and providing support where and whenever necessary. However, this being not practically feasible, the second best option is to offer the student electronic support. Using a computer program as the instructional medium does have a number of advantages. It may offer individualized instruction and practice combined with immediate 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 Storage Environment) is an environment where a law student can practice with finding decisions, with structuring its text and with analysing the decision in order to be able to determine in what way it adds to the body of applicable rules in the legal system. These functionalities are implemented in two integrated modules in CASE:

1. *The Assembler, a module to compile and store decisions.*

In essence the Assembler is a database containing a selection of decisions used in legal education. The law student can do a search (key word and/ or full text) for a specific decision or a set of decisions. Decisions can be added to the database and key words can be indicated for each decision by the teacher. This module can be used separately or in combination with the PAT module.

2. *The Precedent Analysis Tool (PAT), a module to structure and analyse decisions.*

In essence PAT is an instructional environment for learning to structure and analyse a decision to determine how it adds to the body of applicable rules in the legal system. PAT builds on the Assembler module. It presents the student the text of a selected decision together with a framework containing the main elements in a decision text (as, for instance, the different parties and their roles in the various stages of their procedures before the different courts). It allows the student to fill the framework with the relevant parts from the text of the decision. The activities of the student are monitored, and compared to a model where deviations are diagnosed to be able to present the student with a hint or a remediation.

CASE is developed using the principled and structured design approach as described in the HYPATIA project [1, 2]. A short description of this approach is followed by an analysis of the learning task, the difficulties law students experience and the remedies proposed on the basis of both the task analysis and the stated difficulties. This is followed by a description of architecture, functionality, platform and implementation of CASE and a description of a session with CASE and future work.

## 2 Principled and Structured Design Approach

The HYPATIA project [1, 2] 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. 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. HYPATHIA develops electronic tools because they can offer

individualized instruction and practice by adapting to the individual student's performance combined with immediate support and feedback. They can also support the management of information and present different representations and visualisations of legal knowledge and legal tasks. The principled and structured design approach guides the development process in such a way that difficulties and mistakes encountered during the design process may be accounted for. The design process involves three interrelated research streams: basic research, applied research and integration research. Basic research is concerned with developing well-founded models of legal knowledge and skills to be learned by law students, examining the difficulties they have with acquiring legal knowledge and legal skills and finding remedies to enhance effective and efficient learning of the required knowledge and skills. In the applied research part, these findings are used to construct computer supported models of legal knowledge and legal reasoning to diagnose and remedy the specific difficulties of law students in learning the law. Instructional design decisions are made on the basis of a global theory on learning and instruction. In this way the design process will result in a coherent and consistent instructional model. It finally indicates that electronic materials are evaluated extensively (developmental testing and field testing). The integration research part is concerned with the construction of an infrastructure for learning object repositories. The HYPATIA project is divided into specific research programs. The design approach was founded and used successfully in the construction of an instructional environment for learning to solve legal cases: PROSA [3, 4, 5].

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 [3, 4, 5]<sup>1</sup>.

### 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 analyse the task. The HYPATIA design approach starts with (re)constructing explicit models of legal knowledge and legal reasoning. In this (re) construction process, two components are distinguished. (1) A theoretical component of exploration, conceptualization and specification of legal knowledge and legal reasoning resulting in explicit models of legal knowledge and legal reasoning. Two perspectives are taken within this approach: a legal perspective and a knowledge engineering perspective. From the legal perspective different legal sources, being legal empirical research, legal educational practice, legal dogmatic and legal theoretical research, are examined to specify the required models. 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, [6, 7, 8]). This explicitness of models is exactly what is needed in instruction. (2) An empirical component where empirical studies are carried out to acquire insight in the way legal practitioners and legal scientists 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 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 [9, 10, 11, 12, 13, 14, 15, 16, 17].

However, merely instructing a method does not work for novices (see for details [3]).

---

<sup>1</sup>See for a demonstration of PROSA <http://www.lri.jur.uva.nl/munt/manualprosa.html> and <http://www.lri.jur.uva.nl/munt/prosa.html>.

This is partly due to the fact that instructing a method is a problem in itself, as it is difficult to communicate a method, because this requires the translation of actions into words. A method is in fact empty; explaining content is much more "substantial" and therefore easier. The somewhat paradoxical situation is that novices have to learn to determine the legal meaning by determining the legal meaning (see also [3]). Law students especially have difficulties with determining what the decision adds to the body of applicable rules in the legal system. Based on findings in research in legal problem solving (e.g. [3]) it is stated that the difficulties are first of all caused by insufficient mastery of, or insight in, the subject matter. Secondly, especially for novices, methods, often as a side effect, emerge from (novice) problem solving, instead of being the driving force. The subject matter appears to be the major source for finding or trying (a) solution (steps). On closer inspection, a decision is a legal solution for a specific problem situation constructed on the basis of abstract legal rules. Structuring and analysing a decision is in fact the task of reconstructing the problem situation (consisting of a reconstruction of both the facts and the legal question), tracing the abstract legal rules that were applied and specifying the legal solution consisting of the argument structure and the conclusion.

Reading and understanding a decision is not a trivial 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 first of all caused by the fact that a decision is an incomplete reproduction of what happened. Next to that the text of the decision contains many references, both explicit and implicit, to regulations, other decisions and concepts. The fact that a decision has a stratified structure which is also not supported by recognizable clues or elements in the text does not help either. 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 remedies 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 analysing the decision by having her actually carry out these tasks. This is realised by presenting the student with both the full text of the decision and a framework which visualises the elements in a decision necessary to reconstruct the decision in order to determine the legal significance of the decision (see Fig. 1).

There are no applications available that support law students in structuring and analysing a decision suiting the Continental legal system. For the Anglo-American legal system, the CATO application is available [18]. In CATO 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. This means that both the decision at hand has to be presented to the student, as well as the framework for analysis. The student must be able to select text fragments from the decision and paste these 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 module called 'Assembler'. The functionality of structuring and analysing a decision is implemented in the module called 'PAT'. Other basic requirements are maintenance and re-use. It should be possible to make changes to the system and its content without much costs and efforts. Errors in system and content should be easily traceable and

Structuring the decision					
name					
court	place	date	source (publication source, year, number)		
area of law	private law				
	public law	criminal law			
		administrative law			
		constitutional law			
		international law			

Parties			Facts		
instance	initiator	opponent	instance	indisputable facts	disputable facts
first instance			first instance		
appeal			appeal		
cassation			cassation		

Claim		Question			
instance	claim/request	instance	factual	qualification	legal
first instance		first instance			
appeal		appeal			
cassation		cassation			

Legal ground			Dictum	
instance	legislation	decisions	instance	dictum
first instance			first instance	
appeal			appeal	
cassation			cassation	

Reconstructing the legal conclusion		
legal rules	reconstruct legal solution	facts
	conclusion	
	argument structure	
	legal question	

Figure 1: Framework

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 it may facilitate maintenance. The system has functions for adding decisions, adding key words to decisions and preparing decisions for use in PAT. System functionalities are attributed to a user on the basis of her status: administrator, editor, teacher or student. The CASE architecture is depicted in Fig. 2. The Assembler module holds the decisions and allows for search and retrieval of cases and allows teachers to prepare cases for use in the PAT module. Students can use the Assembler 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 analyse a decision she can select one of the reported decisions. This decision and the analysing framework are then made available to the student in PAT. The student can start structuring the decision by selecting text fragments in the decision and pasting these in the correct part of the frame.

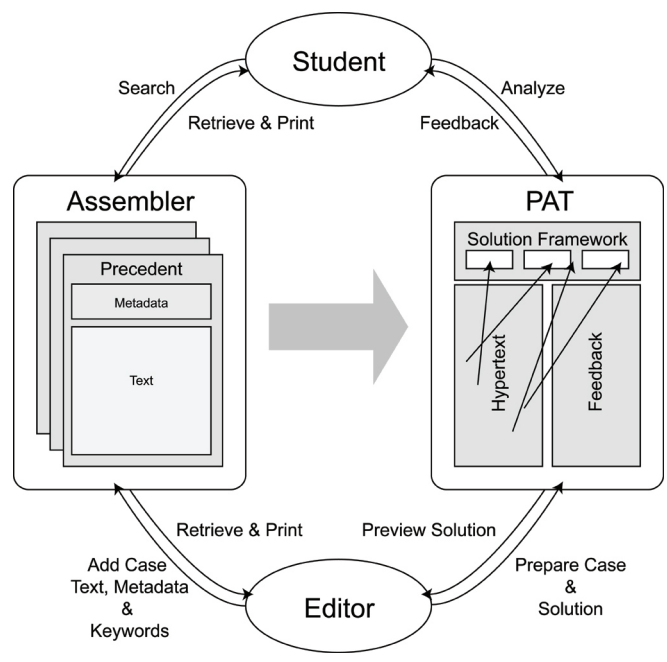


Figure 2: CASE Architecture

## 5 Platform and Implementation

CASE is implemented using a web-based server-side application model<sup>2</sup>. The user interacts with the system using a standard web browser, such as Netscape Navigator, Apple Safari or MS Internet Explorer. The Casetool application is developed using Open Source Software, MySQL (4.0.14) and PHP (4.3.2) and JavaScript. The MySQL database backend contains a number of tables, the most prominent ones being a text fragment table, a solution table and a table storing the student's activities. Casetool's primary component is the server-side application implemented in PHP (4.3.2). This application handles form processing, storage and retrieval of information from the various tables in the database and generating the HTML pages that are output to the user. A small number of simple functions are implemented using client-side JavaScripts. Casetool 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 in PAT; 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 using PAT.

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 labelled according to their place in the solution template. The student can select a text fragment and place 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 het beroep van Ronald G, geboren te Amsterdam op 6 aug. 1954, wonende te Amsterdam, req. van cassatie tegen een bij verstek gewezen arrest van het Hof te Amsterdam van 12 dec. 1977, waarbij in hoger beroep een vonnis van de Rb.”,

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 (see Fig. 1). 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. Each cell in the solution, therefore, can be designated by three coordinates: table, row and column. These coordinates are used to mark the proper location of text fragments within the solution framework. They allow the student's solution to be tested against the solution defined by the teacher; the cell in which the student places the fragment has to match the metadata reference of the text fragment. In the case of an incorrect placement of a fragment, its position relative to the correct place is also known. This allows for standardised responses to common errors. For instance, when a student puts the initiator's name in the opponent's cell, the following response can be generated on the basis of this mixing up of the parties in the dispute: "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.

---

<sup>2</sup><http://www.casetool.nl>

## 6 A 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.

**The Editor** After login, the editor is presented with a menu containing multiple options: editor's menu, teacher's menu, Assembler, PAT, change password and logout. Since she recently came upon a decision relevant for law students, 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 Fig. 3).

The screenshot displays the 'Metadata Editor' interface, divided into two main sections: 'Funke' and 'Keywords'.

**Funke Section:**

- Name:** A text input field containing 'Funke'.
- Instance:** A dropdown menu with 'EHRM' selected.
- Place:** A dropdown menu with 'Strasbourg' selected.
- Date:** A text input field containing '25-02-1993' with a note '(Format: dd-mm-yyyy)'.
- Published:** A field with 'NJ' followed by two input boxes containing '1993' and '485'.
- Change:** A button located at the bottom right of the Funke section.

**Keywords Section:**

- A list of existing keywords, each with a 'Remove' link:
  - fair trial
  - huiszoeking
  - inbeslagneming
  - nemo tenetur
- Add an existing keyword to this case:** A dropdown menu with 'select keyword' and an 'Add' button.
- Add a new keyword to this case:** A text input field and a 'New' button.
- Footer:** A link that says 'Click here to visit the Keyword management page.'

Figure 3: Metadata Editor

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 in PAT. The PAT Prepare tool offers an interface that mimics the regular PAT interface: the editor needs to place pieces of text in the correct position within the solution framework (see Fig. 4).

Where the regular PAT interface checks whether the correct text is in the correct position by consulting the database, the PAT prepare tool writes the action of the editor to the database. The editor in a sense teaches PAT the solution of the case at hand. 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



Figure 4: PAT Prepare Tool

solution, the editor can simply mark these smaller parts. This results in a text fragment with color coded sub fragments that can be placed in the solution table (e.g. Mr Jean-Gustave Funke in figure 4.).

After the editor has finished the above steps, the decision is ready for use by both teachers and students.

**The Teacher** 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 PAT 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.

**The Student** Students can search the decision database using the Assembler search interface (see Fig. 5). 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 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 in PAT.

The PAT interface, shown in Fig. 6, 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 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.

Figure 5: Search the Database

Tekstfragmenten in casus Funke	
Klik op een tekstfragment (of een deel daarvan) om deze in de geselecteerde oel van de PAT label te plaatsen.	
In the case of <i>Funke v. France</i> ,	
The European Court of Human Rights, sitting, in accordance with Rights and Fundamental Freedoms ("the Convention")** and the relevant provisions of the Rules of Court, as a Chamber composed of the following judges: Mr R. Bernhardt, President, Mr Thór Vilhjálmsson, Mr F. Matscher, Mr L.-E. Pettiti, Mr C. Russo, Mr N. Valticos, Mr J.M. Morenilla, Mr M.A. Lopes Rocha, Mr L. Wildhaber, and also of Mr M.-A. Eissen, Registrar, and Mr H. Petzold, Deputy Registrar,	
AS TO THE FACTS	
I. The circumstances of the case	
6. <u>Mr Jean-Gustave Funke</u> , a German national, was born in 1925 and died on 22 July 1987. He worked as a sales representative and lived in France, at Lingolsheim (Bas-Rhin). His widow, Mrs Ruth Funke, née Monney, is French and lives in Strasbourg.	
(gehele fragment)	
A. The house search and the seizures	

partijen		
instantie	initiatiefnemer	wederpartij
eerste aanleg	<input type="checkbox"/>	<input checked="" type="checkbox"/> Mr Jean-Gustave Funke
hoger beroep	<input type="checkbox"/>	<input type="checkbox"/>
cassatie	<input type="checkbox"/>	<input type="checkbox"/>

= selecteer cel,  = toon volledige tekst,  = wis tekst

**Feedback**

Dit is inderdaad de wederpartij in eerste aanleg.

Figure 6: Structuring and analysing a decision

## 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 with tasks, especially with determining what the decision adds to the body of applicable rules in the legal system. Within the current curriculum 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 has been implemented as a web-based server-side application model.

It is implemented using open source software. The Casetool is easy to maintain and re-use and can be made available in different languages. Future work involves testing the effectiveness of Casetool. The claim that law students are supported by Casetool in structuring and analysing a decision in such a way that they are able to grasp the legal significance of the decision should be tested. The claim that it is easy to add a decision, to add key words and to

prepare a decision for use in PAT should also be tested.

## Acknowledgements

The CASE project is partially funded by the Association of Universities in the Netherlands (VSNU) within the project RechtenOnLine.

## References

- [1] Muntjewerff, A.J. (2002). HYPATIA research program. Models of Legal Knowledge and Legal Reasoning in Electronic Materials for Learning the Law. Amsterdam: University of Amsterdam, ISBN 90-806969-1-9.
- [2] Muntjewerff, A.J. (2002). Principled and Structured Design of Electronic Materials for Learning the Law in T.J.M. Bench-Capon, A.Daskalopulu and R.G.F. Winkels (eds.), Legal Knowledge and Information Systems. Jurix 2002: The Fifteenth Annual Conference. Amsterdam: IOS Press, 2002, pp. 133-142.
- [3] Muntjewerff, A.J. (2000). An Instructional Environment for Learning to Solve Legal Cases. PROSA. Amsterdam: University of Amsterdam.
- [4] Muntjewerff, A.J. & J. Groothuismink (1999b). Supporting the Learning of Legal Case Solving using a Computer Program as the Instructional Environment. In Proceedings CATE'99 Computers and Advanced Technology in Education. Philadelphia, USA: IASTED.
- [5] Muntjewerff, A.J. (2002). Evaluating the Instructional Environment for Learning to Solve Legal Cases PROSA. In Gustavo A. Santana Torrellas and Vladimir Uskov (eds.). Proceedings of the IASTED International Conference Computers and Advanced Technology in Education. Anaheim Calgary Zurich: ACTA Press: , pp. 374 - 379.
- [6] Haan, N. den & Sartor, G. (1999). Editorial: Model-based Legal Knowledge Engineering. In Gaines, B.R. (Ed.). International Journal of Human-Computer Studies. Special Issue Model-based Legal Knowledge Engineering. Academic Press, pp. 1037 - 1042.
- [7] Breuker, J.A. & Van de Velde, W. (1994). CommonKADS library for expertise modeling. Reusable problem solving components. Amsterdam: IOS Press.
- [8] Valente, A. (1995). Legal Knowledge Engineering. Amsterdam: IOS Press.
- [9] Bos, A.M. (2003). Jurisprudentieleidraad. In Encyclopedie der Rechtsgeleerdheid Vaardigheden. Amsterdam: Universiteit van Amsterdam.
- [10] Eemeren, van F.H., E.T. Feteris, R. Grootendorst, T. van Haaften, W. den Harder, H. Kloosterhuis & J. Plug (1996). Argumenteren voor juristen. Oefenboek. Groningen: Wolters Noordhoff.
- [11] Franken, H. e.a. (1991). Practicum Methoden en Technieken II. Casusoplossen. Bundel ter dienste van het Practicum Methoden en Technieken van de faculteit der rechtsgeleerdheid van de Rijksuniversiteit te Leiden. Arnhem: Gouda Quint.
- [12] Henket, M. & P. van den Hoven (1999). Juridische Vaardigheden in argumentatief verband. Groningen: Wolters Noordhoff.
- [13] Jansen, L. (1999). Nieuwe introductie in het recht. Utrecht: LEMMA.
- [14] Scholten, P. (1974). Algemeen Deel. Zwolle: W.E.J. Tjeenk Willink.
- [15] Soeteman, A. & A.J. Wolhuis (2003). Syllabus Juridische argumentatie 2003. Amsterdam: VU.
- [16] Stolwijk, S.A.M. & M. Bosch (2002). Het bestuderen van arresten in strafzaken. In Arresten strafrecht/strafprocesrecht met annotaties ten behoeve van het onderwijs verzameld. Deventer: Kluwer.
- [17] Werkgroep Vaardigheden-onderwijs Rechten (1990). Syllabus Vaardigheden I. Amsterdam: UvA.
- [18] Alevan, V. (1997). Teaching Case-Based Argumentation Through a Model and Examples. Pittsburgh: University of Pittsburgh.