An instructional environment for learning to solve legal cases: PROSA

Muntjewerff, A.J.

Citation for published version (APA):
Muntjewerff, A. J. (2001). An instructional environment for learning to solve legal cases: PROSA.

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.
Chapter 5

Computer Assisted Legal Instruction

5.1 Introduction

In the preceding chapters the major issues in designing an instructional environment for learning legal case solving were discussed and design decisions were made. These decisions involved the legal case solving task, the content of the domain of practice and the instructional model. Computer programs that provide an environment for students to acquire skills in applying domain knowledge and that assess and correct students in their performance, are called coaching systems. Coaching systems differ from instructional systems that only present subject matter (domain knowledge) and that only check whether the student has understood the presented material\(^1\). Most so called “intelligent” computer assisted instruction systems, i.e. systems using AI technology, are coaching systems\(^2\). A coaching system presents the student with an environment in which she can practice the skill to be acquired. During practicing the student’s activities and outcomes are monitored which makes it possible for the coach to correct errors and mistakes and to plan further practice. The characteristics of coaching systems are described in more detail in section 5.2. Section 5.3 presents a review of current legal coaching systems with the perspective to relate their educational objectives and other requirements with their design characteristics. These computer programs are discussed to make an inventory of what other legal coaching systems do and how they are designed to meet their educational objectives and other requirements.

\(^1\) An example of this type of instructional system within the legal domain is the concept rehearsal program Administrative law (Huskens, 1997). For other examples in the legal domain see Beek, Boerma & Hurts (1988).

\(^2\) If only for the simple reason that instructional systems are hardly cost effective when compared with traditional media such as books and other written course material.
Chapter 5 Computer Assisted Legal Instruction

This is to be used for refining the design of our instructional environment for learning legal case solving and for justifying the design decisions we made so far.

5.2 Coaching Systems

A coaching system consists of an environment in which the student is enabled to perform the task to be learned or trained. The coaching system monitors the activities and outcomes of the student and compares these with the required activities and outcomes. These systems therefore imply some normative view (as most teachers have). A deviation is viewed as an error or inefficiency. When the coaching system encounters a deviation it subsequently diagnoses what may have caused it. Based on the outcome of this diagnosis the coaching system presents remedial information. In Fig. 5.1 we present a conceptual model of a coaching system.

![Figure 5.1: Coaching system.](image-url)
The conceptual model depicts a global functional decomposition of a coaching system offering a coherent, abstract description of the intended behavior of the system\(^3\). In a coaching system the student performs a task and the system interprets the performance of a student, i.e. it monitors discrepancies between intended results and actual results. If a discrepancy is identified this is viewed as an error or an inefficiency, a deviation from what should be the case. Errors are assumed to have causes, and the identification of causes of errors is called diagnosis. The following functional components are distinguished:

- An environment to enable the task to be learned or trained.
- A monitoring component to observe and interpret the student's behavior while she is performing the task and to identify that there is a deviation.
- A diagnoser to identify the cause of the deviation.
- A coach to assist and instruct the student (see, for instance, Wenger, 1987; Breuker, 1990b; Winkels, 1992).
- A student model. A repository where the information about the student is collected to build a model of the individual student (see, for instance, Sleeman & Brown, 1981). The model keeps track of the changes in behavior and registers what the student is doing and how she does it.

Coaching systems may differ in three major factors. The first factor is the degree of similarity of the environment presented to train or learn the task in comparison with the real environment. The second factor is the degree of freedom the student has in performing the task. The third factor is the degree to which a coaching system is able to “understand” what the student is doing and what her results mean. We begin with a short introduction of each of these factors, starting with the environment, followed by the coaching strategies and the representation of the domain knowledge.

- environment [5.2.1]
- coaching strategies [5.2.2]
- knowledge representation [5.2.3]

\(^3\) A conceptual model is a description of components and their interdependencies in terms of functions rather than in terms of machine executable formalisms and components.
Chapter 5 Computer Assisted Legal Instruction

5.2.1 Environment

A task is performed in some environment. This environment defines or instantiates some problem or goal to be achieved and specifies (makes explicit) the conditions (situation) in which this problem is to be solved or this goal is to be achieved. In summary: the environment is a task environment. For real environments coaching systems are in fact “help” systems. Here a user performs a real life task being the task to be learned or trained in the real life setting. These coaching systems present the user the real environment, not a simulation, and offer help to the user during task performance. Help systems are almost always coupled with other interactive computer systems, for instance they may support operators that monitor (industrial) chemical processes, or they may support users of applications as word processors.

A well-known example of the latter is EUROHELP (Breuker, 1990b). In learning to acquire skill in using an interactive computer program, the user may recognize a need for some piece of information and so she may question available help facilities. However, a user may not be aware of having a problem. EUROHELP is a help system that has also the capability of looking over the shoulder of the user, interpret her performance and offer active help accordingly. This active side of the help system is functionally almost identical to coaching systems. There is however, a subtle difference. In a help system the user has the initiative in selecting a task, therefore a help system is by necessity opportunistic, i.e. this means that it cannot prescribe “exercises”, but is engaged in “on the job” training. A help system can get into a coaching mode by prescribing a series of training tasks to the user, so when help is combined with teaching.

In general, however, the environment in a coaching system is not a real environment, but a representation of reality, i.e. a simulation. Simulation environments can vary to a considerable extent in the way in which reality is represented. Two major categories of simulation environments can be distinguished: model based and non-model based. A model based simulation uses generic models which enable the generation and interpretation of all possible situations on the basis of these models, whereas in a non model based simulation all information needed for performing the task has to be made explicit on forehand. There are two types of generic models: behavioral models and structural models. A behavioral model can either be quantitative or qualitative.
Quantitative behavioral models miss matching structural models. These structural models, however, are available with qualitative models which enables a mapping between a behavioral description and input/output relations. To summarize:

- environment
  - real environment
  - simulated environment (simulation)
- model based simulation (generic models)
- non model based simulation (explicit description)

Here we shortly address quantitative and qualitative model based simulations and the non model based situation description.

### 5.2.1.1 Quantitative simulations

Quantitative simulations use numeric or quantitative models. Well known examples of coaching systems using a quantitative simulation are SOPHIE I (Brown, Burton & DeKleer, 1982), STEAMER (Holland, Hutchins & Weitzman, 1984) and MS-PRODS (Sime & Leitch, 1992a). Purely quantitative simulations are rare in educational systems, because the use of quantitative formulae presupposes an understanding of the structure of the relevant behavioral states of these systems. The correspondence between what appears on the screen and the underlying quantitative model may therefore be difficult for the student to comprehend. Moreover quantitative formulae do not offer causal explanations. This is a major disadvantage because causal explanations are the type of explanations students can understand. To overcome these disadvantages research effort is put into qualitative simulations.

### 5.2.1.2 Qualitative simulations

Qualitative simulations use qualitative models (Forbus, 1988). Qualitative models imply two points of view or types of models.

---

4 The SOPHIE I project instigated research in qualitative models resulting in SOPHIE III. STEAMER is half quantitative and half qualitative. MS-PRODS is also based on multiple qualitative and quantitative models.
These models describe objects and processes in terms of structural and causal relations. The structural model contains descriptions of components and connections or processes with input and output connections (topology). The second type is a behavioral model which describes the functions or behavior of components from which the functions and behavior of the system as a whole can be derived. Well known examples of coaching systems using qualitative simulation are SOPHIE III (Brown, Burton & DeKleer, 1982), MACH-III (Massey, de Bruin & Roberts, 1988), QUEST (White & Frederiksen, 1990), ITSIE (Sime & Leitch, 1992b), MULEDS (Plötzner & Spada, 1992) and STARlight (de Koning, Bredeweg, Breuker & Wielinga, 2000).

5.2.1.3 Situation description

In both quantitative and qualitative simulations an explicit and general representation of the task makes it possible to present a full simulation of the real life task. However, it is also possible to represent only those aspects of a real life task that are relevant for learning the task. In that case the environment is fixed into a description of the specific task characteristics.

Although these views on simulation, and particularly qualitative simulations, are largely inspired by models of physical processes, and in particular physics, it can be shown that the same philosophy also applies to more complex systems (e.g. ecological ones, see Salles & Bredeweg, 1997) and also social systems (Valente, 1995).

The different ways in which the real environment is conceptually simulated is one aspect of a simulation. These should be distinguished from the sensory qualities of a simulation\(^5\). The conceptual qualities are related to the fact that a simulation must have a correct conceptual correspondence with the real life task to be trained or learned. When there is no conceptual correspondence the simulation is nothing more than facade. The sensory qualities refer to the amount in which the sensory experience in the simulation is identical to the real life experience (for example, in simulating a factory the noise and stench are included).

\(^5\) In virtual reality design sensory qualities are referred to as rendering. In Dutch: vertolking, weergave.
An Instructional Environment for Learning to Solve Legal Cases

PROSA

Showing a movie of a real task environment, for example, showing the proceedings of a case before the court, may have high sensory qualities.

5.2.2 Coaching Strategies

A distinction is made between the environment and the coach. Where the environment simulates the problem situation that defines the task to be learned or trained, the coach sees to the learning or training of the skill to be acquired. The coach may vary on task performance that is required or allowed and, related, tutorial style. Coaching systems vary in the degree of freedom the student has in performing the task. To start task performance the student is presented with an initial situation and a problem specification. However, the tutorial style from thereon may vary from constrained to totally free (see, for instance, Berkum & de Jong, 1991; de Jong, van Jooleing, Scott, de Hoog, Lapied & Valent, 1994). In the constrained setting there is an explicit setting of the task. The task is differentiated into a task directed problem or exercise, the goal is stated and the sub-tasks that have to be carried out are traced. In a more free setting the student is presented with a situation. Without explicitly setting a task the coaching system asks the student to explore the environment on the basis of this situation. Another issue here is the appearance of the coach. The coach can either be present as textual feedback and hints, or as a pedagogical agent who is present in the environment (see, for instance, Lester, Converse, Stone, Kahler & Barlow, 1997; Johnson, 1998; Johnson & Rickel, 1998; Shaw, Ganeshan, Johnson & Millar, 1999). Research on pedagogical agents show promising applications and results (see, for instance, Lester et al., 1997; Johnson, 1998; Johnson & Rickel, 1998; Shaw et al., 1999). For example, the pedagogical agent Steve (Johnson, 1998; Johnson & Rickel, 1998)6. The computer tutor Steve is a human like agent that collaborates with the student in a virtual world to help the student to learn7.

---

6 See http://www.isi.edu/isd/carte for a short demonstration of Steve.
7 These human like agents are also referred to as “atavars”. Using animated pedagogical agents in learning is also referred to as guidebot assisted learning. Guidebots help to keep the learner on track, interact with the students in learning environments, engage in instructional dialogue and enhance motivation.
This virtual world is used for training people how to perform tasks such as operating or maintaining complex equipment. The virtual reality world provides a three dimensional interactive mock up of the students’ working environment allowing her to practice tasks. The student enters this virtual reality by putting on a head mounted display. Steve cohabits the virtual world to help the student. Steve first shows the student how to perform the task by performing the task himself, while the student looks over the shoulder of Steve. Steve also talks to the student. He, for example, tells the student what he is going to do (“Let me show you how to perform the pre-start procedure”). Steve also watches if the student is paying attention. This is followed by the student performing the task while Steve looks over her shoulder. Steve has a specific and meaningful role in learning the task of operating complex equipment. However, this role is already somewhat less obvious in Adele (see, for example, Shaw et al., 1999). Medical students are presented with a simulated patient in a clinical setting (a video presentation of real patients on the computer screen). In this case based diagnosis exercise the student has the role of a physician. The student is able to ask questions about the medical history, perform a physical examination, order diagnostic tests and make diagnoses. Coaching is provided by Adele, the pedagogical agent or tutor. Like Steve, she is an animated computer figure; not an animated video of a real human figure. Adele is depicted as a physician and she presents the hints and feedback to the student both in text and with a synthesized voice. The evaluation of Adele showed that student did not find Adele believable as an attending physician. Adele is a pseudo figure who has no specific and meaningful role other than telling the student the feedback that could also easily be presented to the student as text only. The evaluation however, showed that it is not clear if students prefer the persona to a text-only tutor. "Real life" appearance may have no beneficial effect.

5.2.3 Knowledge Representation

Coaching systems vary in the way the knowledge is explicitly represented in the system.

---

8 Adele stands for Agents for distributed learning environments.
Systems that use an implicit knowledge representation encode decisions not knowledge (Wenger, 1987; Sergot, 1991). These systems are for that reason classified as non-intelligent. Systems that do explicitly encode the knowledge are labeled as ‘intelligent’. Explicitness of knowledge representation comes in degrees. With an explicit knowledge representation it is possible to make inferences and to give explanations on the basis of the representation. There is a strong relationship between the type of simulation of the environment and the type of knowledge representation. A qualitative simulation uses an explicit representation of the knowledge. There is also a strong relationship between knowledge representation and system architecture. Where with an implicit encoding of the knowledge the architecture consists of the sequence of the successive decisions, systems with an explicit representation allow a more modular architecture. Components can be separated on the basis of the function they have in the coaching system.

5.2.4 To Sum Up

Environment, coaching strategies and knowledge representation are distinguished as separate factors, which have specific dependencies between them. It is, for example, difficult to imagine a full qualitative simulation without an explicit representation of the knowledge. The ideal is to construct a coaching system with an explicit qualitative simulation environment and a coach with explicit knowledge about this environment, where an explanation consists of elements of that explicit representation. This ideal is not so much that it enables the system to be more ‘intelligent’ as well as that this intelligence allows more individualized and flexible reactions to the performance of the student. Moreover, it allows the system to search for underlying causes (misconceptions) of the student’s errors or inefficient task performance. A qualitative simulation environment with an explicit knowledge representation makes it easier to interpret what the student is doing and to offer the student the proper guidance and remedial. An explicit qualitative simulation makes it easier to construct a fully individualized and adaptive coaching system. There is no need to anticipate explicitly all the possible behaviors of the student as is the case with an implicit representation.
Chapter 5 Computer Assisted Legal Instruction

5.3 Legal Coaching Systems

In the world of coaching systems, legal coaching systems are not very numerous. Here we will describe five of these systems that are typical highlights. They are described to make an inventory of what existing legal coaching systems do and how they are designed to meet their educational objectives and other requirements. This inventory is to be used in refining the design of our instructional environment for learning legal case solving to be realized as a coaching system and for justifying our design decisions made so far. The selected systems share the following characteristics. The task to be learned or trained is legal assessment (Valente, 1995). Task performance is based on a legal case presented to the student together with a stated goal. The method to reach the goal and the content of the product are both important for correct task performance. The systems OBLIGATIO, LITES and Zomerweelede are set in the Dutch legal system which is primarily statute based. STATUTOR and CATO are both set in the Anglo-American legal system. Although this legal system is primarily based on precedents, STATUTOR is statute based; CATO is the only case based system.

There are of course more computer assisted legal instruction programs. Part of these programs instruct domain knowledge, other programs train legal tasks as presenting a case before the court, pleading and argumentation. See, for example, the LEX system, one of the first legal tutoring systems developed, Jones (1986), Haft, Jones & Wetter (1987a), Haft, Jones & Wetter (1987b). Virtual Court Action project, see Barton, McKellar & Maharg (1998), Blackie & Maharg (1998). Computer mediated legal argument in education (DiaLaw) see Lodder (1996). Computer supported environment for the teaching of legal argument, see Bench-Capon, Leng & Staniford (1998). The Dutch Open University presents electronic legal case instruction (Pleading), see van der Heijden (1999). The University of Maastricht (EDIT) developed CoCo, computer assisted legal case solving (Span, 1999, http://www2.unimaas.nl/~edit/). The CoCo program was presented at the end of 1999. We did not incorporate it in our review, mainly because it is a continuation and elaboration of the LITES program.

There are not many other types of legal tasks. Besides legal assessment Valente (1995) distinguishes legal planning and legislation (legal drafting). There are, to my knowledge, no legal coaching systems to practice legal planning or legislation. In traditional legal education legal planning and legislation also play a minor role. The major role is for legal assessment, a task that is included in legal planning and legal drafting.
First a short description of each system is given to get an impression of what they do and how they work. The systems are discussed in the following order OBLIGATIO (1985), LITES (1992), STATUTOR (1992), CATO (1995) and Zomerweelde (1997). On the basis of the review of these systems we discuss what decisions we make in designing our coaching system.

5.3.1 OBLIGATIO

The coaching system OBLIGATIO focuses on two neglected issues in legal case solving instruction (Fernhout, Cohen, Crombag, Pinckaers & Temme, 1987; Fernhout, Otto, Span, van Rijthoven, 1988; Temme & van Willigenburg, 1988). The first one is to question a client to gather the information needed to construct the legal case. The second issue is the legal qualification of the client’s problem definition. Fernhout et al. (1987) state that learning to interact with real clients should be part of the legal curriculum. Because it is almost impossible to realize this in traditional legal education, given the time and teachers needed, OBLIGATIO was developed.

OBLIGATIO mimics real life problem solving dialogues with clients. The student plays the role of the legal practitioner questioning a client on the basis of an outset of a legal case. The student has to learn to ask relevant questions to complete the specific legal case presented. On the basis of the complete case description the student has to solve the case. OBLIGATIO is available to the students at the Faculty of Law of the University of Limburg where the program was developed. There has been no evaluation of the program, although a small validation experiment was carried out to see if the program mimics a real life conversation (Fernhout et al., 1987).

---

11 OBLIGATIO is the name of the application as well as the name of the authoring tool that was built to create the application. The authoring tool is implemented in Turbo Pascal. The idea was that the authoring tool would make it easy for teachers to develop applications in OBLIGATIO. Obligatio was the name of a specific type of examination in the Middle Ages.
5.3.1.1 A session with OBLIGATIO

OBLIGATIO starts with presenting a list of titles. Each title refers to a related situation description. The student may select a title. The incomplete legal case description is presented to the student together with the question that has to be answered by the student. We selected the title 'The Blood Test', which resulted in the presentation of the following (incomplete) situation description. We picture the screen to give a basic idea of the design (see Fig. 5.2), however, the text presented on the screen has been translated into English and is presented here. The (incomplete) situation description 'The Blood Test' reads:

You are a lawyer. A woman of about 25 years of age, with her little son of 2 years old on her lap, is sitting opposite you. She has introduced herself. She just handed you the summons served to her. The woman asks you to be her lawyer in this case. You read the indictment “that she on or around January 10 1986 in the municipality of Maastricht, while being accused that as a driver of a motor vehicle she has acted against article 26 of the Road & Traffic Act after she was ordered by the criminal investigator to submit to a blood test, refused to follow this order or refused to cooperate.”

The question for you is: what verdict are you going to aim for?
An Instructional Environment for Learning to Solve Legal Cases
PROSA

Figure 5.2: Presentation of the case.

At the same time a set of questions is presented to be asked by the student/lawyer and to be answered by the computer program/client.

1. Did you drink?
2. Is it true what is stated in the indictment?
3. First tell me what happened that day.
4. Do you realize that you may end up in jail?
5. I know enough.
Chapter 5 Computer Assisted Legal Instruction

The student has to decide which is the most relevant question by typing in the number (KEUZE = Select).

This is followed by the presentation of the clients’ answer (see Fig. 5.3 ANTWOORD = answer).

The conversation continues with the student asking another question. The question that was already asked is replaced by new questions.

Figure 5.3: Client answering the question.
An Instructional Environment for Learning to Solve Legal Cases

When the student/lawyer thinks she knows enough to be able to answer the question presented with the legal case, she has to select the ‘I know enough’ option from the question list. A screen then pops up where the student can type her answer.

The computer program gives immediate feedback if the student activates the available tutor option (see Fig. 5.3). OBLIGATIO assesses the student’s answer by comparing it with a list of standard formulations of the correct answer. The questions the student has selected are also assessed. Each legal case has a score. This score can be influenced negatively by selecting a question that is marked as stupid and positively by selecting a question that is marked as very good.

5.3.2 LITES

OBLIGATIO was the inspiration for the development of LITES. Both programs were developed by the same research group. LITES is developed as a research vehicle to demonstrate that it is possible to develop a legal intelligent tutoring system. LITES stands for Legal Intelligent Tutoring/Expert System and is also Latin for legal dispute (Span, 1988; Span, 1992; Span, 1993; Span, 1994). A major effort has been put into the representation of the knowledge of the domain.

The domain was represented as a flow-chart and then implemented using the constructed knowledge representation language\textsuperscript{12}. The student has to use her knowledge about the domain of practice in three sub-tasks. The domain of practice is the acquisition of ownership by a third party in good faith\textsuperscript{13}. What is at stake in this domain is that property is passed from a person, not being the owner, to another person who did not know that this person was not the owner, but assumed him to be the owner.

\textsuperscript{12} The knowledge representation language used is PL+. This language was developed by Fernhout. PL+ is based on the knowledge representation language developed by Nieuwenhuis (1989). PL+ uses production rules to represent knowledge. Inferences are made using backward reasoning.

\textsuperscript{13} Article 3:86 Dutch Civil Code (3:86 BW) acquisition through an incompetent party and protection of a third party who is in good faith. In Dutch: de bescherming van de derde verkrijger te goeder trouw. The situation description introducing Rosa, Kees and Jet used in our think aloud experiment described in Chapter 2 is an example of the third party in good faith problem.
Chapter 5 Computer Assisted Legal Instruction

The question is under what conditions the person who is the owner can reclaim the property. The first sub-task in LITES is to find relevant facts. The second sub-task is to translate the facts into legal terms used in the domain of practice (qualification). The third sub-task is to construct the set of reasoning steps to be able to come to a conclusion, that is to solve the case. LITES is made available to the students at the Faculty of Law at the University of Limburg. There has been no evaluation of the program.

5.3.2.1 A session with LITES

Find relevant facts To find relevant facts LITES communicates to the student that a case based on article 86, book 3 of the Civil Code was constructed by the computer program. However, this case remains hidden (see Fig. 5.4).

It is up to the student to collect the relevant facts about this hidden case by selecting questions from a list of questions presented by LITES (Fig. 5.5).

| I constructed a case based on article 86 chapter 3 Dutch Civil Code (this case is not callable) |
| A case such like this concerns three parties involved |
| * Agnes is the “original” owner |
| * Beatrice is the alienator having the power over the property |
| * Constance is the third obtaining party |

Keeping this in mind I am now going to give you the opportunity to find out the relevant facts in the case at hand. Therefore I am going to present you a set of multiple choice questions from which you have to select the most relevant question.

Figure 5.4: The hidden case in LITES.

---

14 The screen design consists of text only as depicted here. Therefore instead of using pictures of the real screens, which are all in Dutch, a translation of the screen contents was made to get a better understanding of the program.

15 Callable = oproepbaar.
The student is then asked to select the most relevant question from a set of questions.

<table>
<thead>
<tr>
<th>You have to ask me one of the following questions. Select the questions in such a way that you acquire the most relevant information about the case at hand.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is Agnes in this case the owner of:</td>
</tr>
<tr>
<td>1. a right to bearer</td>
</tr>
<tr>
<td>2. a right to order</td>
</tr>
<tr>
<td>3. a right to pledge</td>
</tr>
<tr>
<td>4. a right to mortgage</td>
</tr>
<tr>
<td>5. a right <em>in rem</em></td>
</tr>
<tr>
<td>6. movable property</td>
</tr>
<tr>
<td>7. registered property</td>
</tr>
</tbody>
</table>

Make your choice [1 -7]

Figure 5.5: Collecting the relevant case facts.

**Translate facts into legal terms** To translate the facts into legal terms (qualification) LITES constructs a case and presents the case to the student (see Fig. 5.6).

<table>
<thead>
<tr>
<th>Text of the case presented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agnes is the owner of money.</td>
</tr>
<tr>
<td>On October 2 1989 Beatrice steals the money from Agnes.</td>
</tr>
<tr>
<td>The money is now in the power of Beatrice.</td>
</tr>
<tr>
<td>On February 14 1990 she pays Constance with Agnes' money.</td>
</tr>
<tr>
<td>Beatrice delivers the money to Constance.</td>
</tr>
<tr>
<td>Constance did not have any reason to doubt that Beatrice was not the owner of the money.</td>
</tr>
<tr>
<td>On May 24 1991 Agnes finds out that Constance has the money.</td>
</tr>
<tr>
<td>Agnes wants to revendicate the money in the hands of Constance.</td>
</tr>
</tbody>
</table>

Figure 5.6: Presentation of a case.
Chapter 5 Computer Assisted Legal Instruction

The computer program asks questions based on the case facts. The student has to answer these questions with either ‘yes’ or ‘no’ (see Fig. 5.7).

I have constructed a case for you and I will start questioning you about the qualification of the facts relevant for this case.

Try to realize why I am asking only these questions!

You will benefit from first reading the case carefully.

* Does it concern a registered property? (Y/N):

Figure 5.7: Qualification of facts.

**Construct the set of reasoning steps** The third sub-task consists of ordering the reasoning outcomes (the qualified facts) as a list of successive steps (see Fig. 5.8).

Whenever you do not receive the feedback “This is correct”, your answer is wrong. State logically and detailed as possible the steps that you use in reasoning.

1. the property is no registered property
2. the property is registered property
3. Beatrice had power over the property
4. Beatrice had no power over the property
5. Beatrice delivered to Constance
6. Beatrice did not deliver to Constance
7. Beatrice had power of disposition
8. Beatrice had no power of disposition

Make your choice [1-8]

Figure 5.8: Selecting the reasoning steps.

Finally the student is presented with a set of answers from which she has to choose the right answer. The answer is the conclusion based on the set of reasoning steps selected by the student.
Construct a case LITES also allows the student to construct a case from a set of pre-defined categories (see Fig. 5.9).

We are now going to construct a case based on article 86, chapter 3 Dutch Civil Code.

Such a case involves three parties:

* first of all the “original” owner of the property who loses the power over the property
* next the alienator having the power over the property and planning to deliver this property
* finally the third obtaining party to whom the alienator delivers the property

Keeping this scenario in mind I am going to try to picture the case by asking a set of questions

What is the name of the original owner? :

Support in LITES is available in a number of ways. The student has access to explanation about the domain of practice. The explanation is presented as text about the subject matter content and as a decision tree representation of the article 86. LITES gives immediate feedback in case of an incorrect actions by the student. The computer program also assesses the outcome of the selection of reasoning steps (see Fig. 5.10).
Your reasoning is incorrect or partially incorrect!

The reasoning is as follows:

In this case there is no registered property and Beatrice, having no power of disposition, had power over the property at the moment of delivery to Constance.
Because Constance was in good faith and paid for the property she may appeal for the protection of article 86.
Appeal by Agnes on the basis of article 87 is impossible because the property involved is money. We may conclude that Constance is the owner of the property.

In solving this case you selected one or more incorrect reasoning steps!!

We will try again with another case.

Figure 5.10: Feedback in LITES based on assessment of reasoning steps.

5.3.3 STATUTOR

The research background of STATUTOR is the formalization of statute law for use in a knowledge based system. The formalization was reused in a prototype system for presenting exercises that resembled the case analysis exercises of traditional legal education (Routen, 1991; Routen, 1992). The prototype was further developed to produce a system to be used in law schools (Centinia, Routen, Hartman & Hegarty, 1995). STATUTOR stands for statute tutor. The student is trained in constructing legal arguments for a case on the basis of a statute (i.e. the British Data Protection Act 1984, DPA, sections 1 and 21).
The student can demonstrate her understanding of the structure of a legal source by constructing an argument. The student selects facts, not-facts (negated facts), conditions and a conclusion and constructs the argument structure by selecting and linking the elements (see Fig. 5.11).

\[16\] Not-facts or negated facts are facts of which it is not known whether they are true or false or facts that have been effectively negated.
The argument is constructed as a graphical tree-like structure. There are two (global) strategies that the student can use: (1) she can start with a legal conclusion (e.g. De Montfort University is a data user) and then construct supporting arguments (backward reasoning) or (2) she can select facts and prove a condition (forward reasoning). A formative evaluation was carried out. A small number of subjects worked with the program and were asked to give their opinion.

5.3.3.1 A session with STATUTOR

The student first selects a conclusion (see Fig. 5.11). In this case the conclusion to prove is “De Montfort University is a data user”. Then the student has to select the facts (e.g. National Insurance Numbers has been recorded), the not-facts (e.g. Computers can operate automatically on National Insurance) and the conditions (e.g. Individual Working Record is data according to the DPA) she considers necessary for proving the conclusion.

Now the student has to prove the conclusion by constructing the argument using these items. To prove this conclusion the student may opt for a backward reasoning strategy.

By linking the condition “De Montfort University holds the data National Insurance Numbers” the student states that the conclusion can be inferred directly from this condition (e.g. IF De Montfort University holds the data National Insurance Numbers THEN De Montfort University is a data user). The inference is labeled using the section number of the statute where the condition can be found (e.g. Data Protection Act, section 1, subsection 5).

---

17 However, despite all our efforts we did not succeed in finding a report of the results.
The student may also opt for a forward reasoning strategy starting with the known facts (e.g. Individual Working Record has been recorded AND Computers can operate automatically on Individual Working AND NOT Computers can operate automatically on National Insurance) and inferring that section 1, subsection 2 of the DPA applies (e.g. Individual Working Record is data according to the DPA).
STATUTOR also presents ‘forwards-only’ exercises. The student only has disposal of facts (see Fig. 5.12).

![Diagram of 'Forwards-only' exercise.](image)

Figure 5.12: ‘Forwards-only’ exercise.

The screen shows a box with a question mark presenting an unknown condition. The strategy is now restricted to forward reasoning. The student links facts to infer an intermediate conclusion depicted by a question mark (e.g. Individual Working Record has been recorded AND Computers can operate automatically on Individual Working AND Instructions are given to operate on Individual Working Record).
Chapter 5 Computer Assisted Legal Instruction

If the student’s argument structure is correct the intermediate conclusion is presented in the question mark box (e.g. Individual Working Record is data according to the DPA). A new question mark box appears being a new intermediate conclusion to be inferred on the basis of new facts, not-facts and conditions. The student continues her argument construction process until she reaches the final conclusion.

![Argument structure diagnosis](image)

Figure 5.13: Argument structure diagnosis.

STATUTOR offers two types of support. Support that is fully under control of the student and support controlled by the system.
Examples of student controlled support are: (1) the student may click on a part of the argument under construction to get a hint or help from the system. (2) The student may browse the statute text. (3) The student may ask for an immediate diagnosis of her argument under construction. System controlled support is a feedback regarding a wrong or unnecessary construction step taken by the student (error; inefficiency).

A diagnosis option for the complete argument structure is available. STATUTOR compares the correct argument structure with the structure constructed by the student. The outcome of the comparison is presented as a graphical representation were missing, wrong and redundant parts in the structure are depicted (see Fig. 5.13). A box with a cross indicates a wrong part in the argument structure, where an empty gray box indicates a missing part. For example, the fact 'Instructions are given to operate on Individual Working Record' does not belong in the argument structure.

5.3.4 CATO

Both STATUTOR and CATO train students to construct an argument. However, where STATUTOR is statute based, CATO is case based. In CATO the student is trained to construct arguments with cases (Aleven 1997; Aleven & Ashley, 1992, 1993, 1995, 1997; Ashley & Aleven, 1991). Argumentation is a basic activity of legal practitioners. Therefore it is part of legal training. However, teaching the skill is very labor-intensive. Designing a computer program that teaches the basic skills of argumentation may leave the teachers with time to focus on the more advanced skills in argumentation.

With the construction of CATO two claims are tested. The first is that the instructional approach of CATO is an effective way of learning to make arguments with cases. The second claim is that the model of case based argumentation incorporated in CATO contributes to research on case based reasoning (Ashley, 1990; Ashley, 1991). To learn to make arguments with cases students have to practice two different tasks in CATO: (1) theory testing and (2) argumentation. In the theory testing task the student tests a general proposition about the domain of practice against a body of cases. In the argumentation task the student has to construct an argument by comparing and contrasting the case to precedent cases. The example domain of practice in CATO is trade secrets law. CATO is available to students of the University of Pittsburgh School of Law.
Different formative evaluations have been carried out during the development of CATO. CATO was evaluated in the context of a second-semester legal writing course at the University of Pittsburgh School of Law involving 30 students. Instruction with CATO was compared to small-group instruction by an experienced legal writing instructor. Both the instruction by CATO and by the legal writing instructor led to statistically significant improvements in students’ basic argumentation skills. However, in a second test involving a more complex context the control group even outperformed the CATO group.

5.3.4.1 The model of case-based legal argumentation in CATO

The CATO argumentation model addresses arguments in which two adversaries compare a problem to past cases, assess the significance of similarities and differences on the basis of the domain knowledge and organize arguments by issues. An important goal in CATO is for students to learn to analogize and distinguish cases and to assess the similarities and differences between cases. The CATO argumentation model consists of eight Basic Argument Moves (see Fig. 5.14). These moves are activities to be carried out by the student when citing cases in an argument (Aleven, 1997, p. 19).

1. Analogizing a problem to a past case with favorable outcome
2. Distinguishing a case with unfavorable outcome
3. Downplaying the significance of a distinction
4. Emphasizing the significance of a distinction
5. Citing a favorable case to emphasize strengths
6. Citing a favorable case to argue that weaknesses are not fatal
7. Citing a more on point counterexample to a case cited by an opponent
8. Citing an as on point counterexample

Figure 5.14: Basic Argument Moves.

In the CATO argumentation model relevant similarities and differences (factual strengths and weaknesses) between cases are represented using factors.
Factors are abstractions of case facts that make a case weaker or stronger for a party involved. These factors are related to more abstract legal knowledge in a knowledge structure that is called the Factor Hierarchy. The Factor Hierarchy is used for reasoning about the significance of differences among cases, identifying issues in a problem and organizing arguments by issue. Aleven (1997) identified 26 factors for the trade secret domain, half of which favor the plaintiff, half of which favor the defendant. The argumentation model is used to present dynamically generated examples and to visualize (reify) the argument structure. The model is also used to make the student’s tasks more manageable.

5.3.4.2 Instruction with the CATO argumentation model

For learning the argumentation task CATO presents an argumentation problem to the student involving two parties. The student constructs an argument by comparing and contrasting the problem to precedent cases. The student has to outline written arguments on behalf of both parties. The instructional sessions in CATO see to the following activities:

1) part 1
   • analyze facts of the problem case presented.
   • identify factual strengths and weaknesses related to trade secrets claim
   • map to applicable factors using the Case Analyzer tool.

2) part 2
   • study and practice Basic Argument Moves, guided by examples, generated by CATO’s Argument Maker tool, which involve the problem case and cases retrieved from the database.
   • compare arguments to CATO’s arguments with the same cases. practice identifying distinctions with CATO’s Mini Dialog.

3) part 3
   • search CATO database for relevant cases that plaintiff can cite in his argument.
   • evaluate relevance of retrieved cases on the basis of argumentation examples generated by CATO (Argument Maker) and the textual summary of cases (Squib Reader).
Chapter 5 Computer Assisted Legal Instruction

4) part 4
- identify issues in the problem case using issue-based argument generated by CATO as guide (Issue Based Argument Window).
- select the most relevant cases.
- organize plaintiff’s argument by issues.
- outline an argument.
- compare with CATO’s issue based argument.

5) part 5
- outline defendant’s response and compare to CATO’s argument.

The student has to work on the problem on the basis of the CATO argumentation model using the available tools (Aleven, 1997, p. 28) (see Fig. 5.15).

<table>
<thead>
<tr>
<th>CATO Database</th>
<th>contains textual summaries (squibs) and factor sets for 147 trade secrets cases.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factor Browser</td>
<td>provides information about CATO’s set of 26 factors for trade secret law.</td>
</tr>
<tr>
<td>Case Analyzer</td>
<td>lets students compile a list of applicable factors for a case and compares students’ list of factors to that stored in the CATO database for the same case.</td>
</tr>
<tr>
<td>Argument Maker</td>
<td>presents argumentation examples in the context of the students on-going work. Conducts mini-dialogue to give students practice in identifying distinctions.</td>
</tr>
<tr>
<td>Issue Based Argument Window</td>
<td>presents examples of arguments organized by issues with multiple cases selected by students.</td>
</tr>
<tr>
<td>Squib Reader</td>
<td>displays squibs of retrieved cases.</td>
</tr>
</tbody>
</table>

Figure 5.15: CATO Tools.

Different types of support are available in CATO: examples, reification, information management and feedback. The support is based on the CATO argumentation model and uses the CATO tools.
CATO uses argumentation examples in order to help students construct case-based arguments. CATO presents argumentation examples that illustrate CATO's Basic Argument Moves and Issue Based Arguments. These examples are generated dynamically by CATO and can therefore be tailored to the students' analysis of the problem. Students have to study these examples and evaluate their own arguments against CATO's. CATO annotates each example with a brief explanation of its structure. CATO presents a visualization of the argument structure. In CATO students work with textual representations of cases and formal representations of cases as lists of factors. CATO explicates how these factors relate to their use in arguments by reifying these connections. CATO offers a set of tools to make the process of legal argumentation more manageable. CATO helps the student to find relevant cases and to evaluate the relevance of retrieved cases by providing a database of cases indexed by the factors that guide their applicability in arguments. By reducing the complexity CATO enables the student to focus on making arguments with cases, and testing and refining theories in the light of retrieved cases. CATO presents some limited feedback on students' activities for the sub-tasks identifying strengths and weaknesses and identifying distinctions among cases.

5.3.5 Zomerweelde

The computer program "District Court Zomerweelde" (Summer Bliss) allows students to practice criminal proceedings. Zomerweelde was developed for the Open University in the Netherlands. Proceedings skills are important in legal practice and should therefore be part of legal education. These skills are sometimes practiced using a moot court. However, for the Open University, being a distance learning institute, a moot court is no realistic option for the students. Therefore an electronic moot court was developed. In Zomerweelde the student has to complete a meeting of the criminal court quickly and adequately (Zitzen, 1996; Nadolski & Wöretshofer, 1998). Students have to respond to events, distinguish relevant facts, construct an argument and assess arguments. Zomerweelde contains four different cases. The student plays the role of legal adviser in three of the cases and the role of public prosecutor in one.

18 Proceedings skills in Dutch: proces vaardigheden.
19 We refer to the 1998 version of Zomerweelde.
Chapter 5 Computer Assisted Legal Instruction

The system is available to students of the Open University. There has been no evaluation of the program.

5.3.5.1 A session with Zomerweelde

By starting Zomerweelde the student enters the court house. The porter asks the student to fill in student data and offers explanation on the tool icons. The first thing the student has to do is to go to her study (see Fig. 5.16 werkkamer = study).

Figure 5.16: In the study.
In the study the student can select a case. Each case includes a file and a meeting of the court. The file includes the charges, the indictment, case law and expert reports. The file is available in the form of a text book. After a case is assigned, the student has to prepare the case by going through the file and prepare her memorandum of oral pleading. When she is finished she may go to the courtroom (rechtszaal).

Figure 5.17: Argument input.
Present in the courtroom are the judge, the public prosecutor (when the student has the role of legal adviser), the suspect and possible witnesses. The student/legal adviser can use the icons on the screen to specify her activity. There are icons for the different tools that are available, there is an assessment icon, an icon to go to the study or to the exit, and there is a microphone icon to indicate that the student/legal adviser wants to speak. The icons are depicted on the screen only when the functionalities are made available. The conversation takes place in the form of selecting arguments from a prestructured set of arguments. The arguments the student/legal adviser wants to bring forward are introduced using the argument input scheme (see Fig. 5.17). All parties in the courtroom may react to questions or arguments presented by the legal adviser. These reactions are presented either as text, drawings or audio-fragments.

Three types of support are available in Zomerweelde: feedback, explanation and tools. Both the judge and the public prosecutor respond to an answer or argument of the student/legal adviser during the proceedings. Explanation is available in the form of an electronic guide and the student can use tools as a memorandum of oral pleading, a logbook and a list of professional terms. After closing of the proceedings the students’ strategy is evaluated by the system and brought to the student by the patron. The strategy is assessed on items as correctness, efficiency, right questions, right arguments, adequacy and speed.

5.3.6 Design Issues

On the basis of the preceding description of the characteristics of coaching systems in general and the review of a selection of legal coaching systems we will now discuss what to adopt and what to reject. The basic decisions in designing a coaching system concern the environment, coaching strategies (the task performance and tutorial style), the knowledge representation, the student model and the interface.

5.3.6.1 Environment

As stated earlier, the ideal is to construct a coaching system with an explicit qualitative simulation environment.
An Instructional Environment for Learning to Solve Legal Cases

Such an environment not only contains all modeling components and behavioral descriptions for simulation, but it provides at the same time an explicit knowledge representation from which explanations can be drawn. However, none of the legal coaching systems discussed has such an explicit qualitative simulation of the task environment within the system itself: CATO’s factor hierarchy comes closest. All the systems are therefore classified as non model based, pre-specified situation description simulations (see Fig. 5.18). However, that it is possible to model the law in such a way that all possible situations can be generated and interpreted on the basis of this model is shown in e.g. TRACS (Den Haan, 1996). TRACS generates traffic situations for which a normative qualification is provided on the basis of an explicit representation of the Dutch Regulation on Traffic Rules and Traffic Signs. Valente (1995) also demonstrated that it is possible to make an explicit representation of legal sources. However, to realize a full qualitative simulation of legal case solving is very complicated and time consuming. Moreover, the simulation may even not add much to what is relevant for learning the task of legal case solving as compared to using a restricted model. A legal world is for a large part a common sense world, i.e. a world that can be (safely) assumed to be known to the student. For example, in the world regulated by the Dutch Regulation on Traffic Rules and Traffic Signs common sense notions as left and right are very important. However, these common sense issues are not what the student has to learn. Most common sense notions are no problem for a student. However, anticipating and checking all the implications and inferences that result from the description do cause difficulties for the student. This indicates that the world the regulations refer to does not have to be explicitly part of the instructional environment. A more realistic and pragmatic option is then to construct a environment that is not model based but pre-specifies the task to be learned. Therefore there is no need for a generic model based simulation, a pre-specified description of the task to be learned satisfies.

Besides the conceptual, knowledge representation aspect, sensory qualities of a simulation may play a role. In what way are we going to present the legal assessment task to the student?

---

20 A quantitative simulation is no realistic option for the legal domain.
21 In Dutch: Reglement Verkeersregels en Verkeerstekens.
Chapter 5 Computer Assisted Legal Instruction

Legal practice works with documents, rather than with real life situations: cases are described by documents. Therefore, visualization can also be restricted to presenting text. Of course it is possible to present a movie or to have students work with real files. However, there are many reasons not to do this. When presenting a movie or real files the conceptual correspondence needs careful attention. The focus of legal assessment is not on handling files. So in fact it is quite a detour to learn legal assessment by showing a movie or presenting files. Not to mention the management difficulties and the lack of control over all possible interpretations and inferences students may make. An environment with high sensory qualities, but poor conceptual qualities presents us with no more than a décor. In simulating the legal assessment task we therefore opt for only simulating the specific situation, or legal case, being the real environment.

5.3.6.2 Coaching strategies

We argued that we did not expect that explicit instruction of a case solving method would result in better learning. OBLIGATIO and LITES, both based on the research of Crombag and colleagues, do instruct an explicit method. However, their is no evidence that this approach is effective. STATUTOR and CATO make use of visualization, however, they put the emphasis on visualizing the outcome, where we also plan to visualize the task components.

Using agents in the environment to help students with the task performance can be useful when the agent has a meaningful role, more than just popping up to tell something that could also be presented as text. In this case the agent instead of being helpful gets in the way. The agents in Zomerweelde do not seem to have a role other than to present hints and feedback.

5.3.6.3 Knowledge representation

An explicit knowledge representation is only under discussion when a simulation is model based.
A classification of the reviewed systems with regard to environment and knowledge representation results in the following overview in which we also included some coaching systems in other domains (see Fig. 5.18).

<table>
<thead>
<tr>
<th>representation</th>
<th>simulation category</th>
<th>qualitative</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>knowledge</td>
<td>定量的 SOPHIE I</td>
<td>QUEST ITSIE</td>
<td>CATO</td>
</tr>
<tr>
<td>decisions</td>
<td></td>
<td></td>
<td>OBLIGATIO LITES STATUTOR Zomerweelde</td>
</tr>
</tbody>
</table>

Figure 5.18: Relating environment - knowledge representation.

A situation description environment can be realized without an explicit knowledge representation. The only things we plan to represent are the specific situation or legal case, the legal rules in relation with the case to be able to match case facts with legal rules, and the correct solution for each case. However, this representation is hard wired, that means that there is no internal representation, only text. Architectural issues do not play a significant role either now we have opted for a coaching system with an implicit knowledge representation.

5.3.6.4 Student model

A student model may serve many purposes in instruction, for example, founding explanations (see, for instance, Clancey, 1986; Dillenbourg & Self, 1992; McCalla & Greer, 1994; Corbett, Anderson & O’Brien, 1995). This requires the full realization of the functionalities as depicted in Fig. 5.1. We do not realize all these functionalities in our coaching system. Therefore we also do not have the possibility of maintaining a real student model. Because the system does not have a full and explicit representation of the knowledge, the student model content is not based on reasoning about the kind of deviations, their causes and how to plan a remedial.
Chapter 5 Computer Assisted Legal Instruction

To a certain extend student modeling can take place, however, all decisions have to be set out in advance. However, some administration of the students activities and outcomes may come in handy for both student and teacher. To be able to present the student with the possibility of an overview of her achievements and an assessment of these results is informative and motivating (Keller & Suzuki, 1988). The teacher may use the information for diagnostic purposes. In Fig. 5.18 CATO is situated within the explicit knowledge representation category. Although CATO has all characteristics of an explicit representation containing a computational model of case based argumentation, the system does not have to understand the students argumentation because CATO does not maintain a student model. The computational model in CATO is only is used to generate argumentation examples.

5.3.6.5 Interface

The interface is designed to realize the environment and the coach. As we stated before, we ‘simulate’ the real world indirectly in presenting the student with a textual representation of a legal case. However, we also want to present the student the characteristics and basic components of the task. In our view depicting these components in the interface may serve as an external memory already supporting the student in her task performance. The instructional model presented in Chapter 4 prescribes primary and secondary presentation forms. The functional difference between these presentation forms can in our view best be depicted by also separating them clearly in the interface. Colors are used to support this functional separation. Using a standard sized screen it is inevitable to scroll text. We present no facility for students to type in text. We prefer using drag and drop facilities, allowing the student to cut and paste text. In this way she does not have to retype text, being a waste of time and boring, and we do not have to take care of typing errors.

5.4 Conclusion

In this chapter we refined and justified the design of the instructional environment for learning to solve legal cases.
An Instructional Environment for Learning to Solve Legal Cases

PROSA

We decided to realize the instructional environment as a coaching system because this type of systems is best suited for problem solving learning, that is, learning by using knowledge. A coaching system monitors the students during task performance and offers support to make the student acquire the skill. We discussed the major design issues in realizing the coaching system. These issues are the environment, the coaching strategies, the knowledge representation, the student model and the interface. We argued that an instructional environment using a simulation is rejected. It is better to use a situation description environment whenever a simulation is either impossible, too complex or trivial. To really simulate (parts of) the legal system where manipulations can be evaluated is too complex and therefore too time consuming. To learn to use domain knowledge in a task as legal case solving (legal assessment) it is not necessary to use a fully simulated world of the domain in question.

The environment to be realized in the computer program will therefore be the non model based situation description. The simulation will be in fact no more than a piece of text. This text is a derivation of a case file which in turn is a derivation of the real life event or situation. Every exercise will contain elements of the real situation. However, these elements are compiled out. Although the environment will only be a fragmented and implicit description of the real life environment there will be an explicit task description. However, the task performance and the legal case description are restricted to what is most relevant for learning legal case solving.

The tutorial style to be incorporated in the computer program will be more or less constrained. The task will be differentiated into a task directed exercise. The task is explicitly set by stating the goal with the exercise. The sub-tasks to be performed by the student are traced by the computer program and the outcomes are assessed.

The computer program will not be realized as an intelligent system. There will be no explicit knowledge representation, only a decoding of decisions. In the now following chapter (Chapter 6) we will describe the functional specification and the implementation of PROSA on the basis of the decisions made so far.