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.
Chapter 6

PROSA

6.1 Introduction

This chapter is about designing and implementing PROSA. PROSA stands for PROblem Situations in Administrative law. PROSA presents an environment in which the student has the opportunity to solve legal cases, that is, construct a legal solution for a legal case by applying legal rules. The instructional environment offers guided access to the subject matter content, instead of presenting the student an explicit method for legal case solving. In the previous chapters successive steps were taken to specify the major requirements for designing PROSA. We now combine these partial requirements and make additions where necessary to realize PROSA. In paragraph 6.2 we therefore set of with prescribing what we want the system to do. These functional requirements are then specified in a set of functions that describe the system to be designed. This is followed by a description of the implementation of the system in paragraph 6.3. In paragraph 6.4 we describe a session with PROSA.

6.2 Requirements

PROSA has to coach the student during legal case solving to enable her to construct a complete and correct legal solution. The functional requirements and constraints that prescribe what the system to be designed should do are the outcomes of the previous chapters. These, more or less isolated, requirements now have to be combined and necessary additions need to be made to be able to realize PROSA. We therefore first briefly restate these requirements.

1 In Dutch: PRObleem Situaties Algemeen bestuursrecht.
6.2.1 Externalization

In Chapter 2 the task of legal case solving was examined resulting in a conceptualization of the task and an inventory of difficulties in legal case solving. This inventory lead to the conclusion that students should be supported in managing information and prevented from ‘jumping to conclusions’ to enable the student to construct a correct and complete legal solution. We decided that this requires an environment in which the task components and task characteristics are made explicit in such a way that it restricts the set of activities that have to be performed by the student and it presents systematic guidance to the student.

We also discussed the issue of instructing an explicit method or not. Using a method or having a method available can be useful to prevent jumping to conclusions. However, we also showed that instructing an explicit method is not very productive in learning to solve legal cases.

We described that performing, for example, the ‘select fact’ activity is not as trivial as it may seem. To select a relevant fact from the situation description requires much leafing that in turn requires the storage of many intermediate findings. As a consequence (short term) memory is exceeded in no time. Making notes may function as a kind of external memory, however, this involves copying articles from statutes and facts from the situation description. This not only is much work, it is also something that students will not do or will not do exhaustively. However, we do want the student to work in a systematic way to prevent them from ‘jumping to conclusions’ and to enable them to construct a complete and correct solution.

We decided to present an environment that meets both requirements: (1) relieve the student of the task of keeping track and recording intermediate results and (2) enable the student to work in a systematic way. We will not instruct an explicit method, although a certain mapping to a method is required to be able to restrict the set of possible activities to keep the student on track. To summarize:

Externalization:
- restricts and guides activities
- offers systematic support
We want the student to construct a legal solution by herself. By actually having the student work on the construction of a legal solution she may experience what it takes to construct a solution and to “go through the problem” so to speak. In Chapter 2 we examined legal case solving. We found that the different theoretical sources we consulted present a more or less identical decomposition of the task. However, we also found that the difficulties in legal case solving are not primarily related to what activities have to be carried out, but more to how these activities have to be carried out. This revealed that the major role in solving a legal case is reserved for legal knowledge.

6.2.2 Functional Differentiation of Knowledge

In Chapter 3 a legal ontology of law (FOLaw) was discussed which distinguishes various types of legal knowledge. This made clear that the understanding and legal interpretation of a situation description requires a correct mapping between this situation description and the LAM, i.e. the world knowledge implied by the domain of practice, in our case the GALA. Based on our insights in students difficulties with solving legal cases we decided that this mapping should be made explicit in such a concrete fashion that it also should act as an external memory that marks which propositions in the legal case have been covered by the law and which have not been. The latter may mean that the proposition is either not relevant in legal terms or has been overlooked. This lead us to differentiate between the legal case, the legal rules and the legal solution, where we also discussed the different components that make up a solution. We also found out that students have difficulties finding their way in the legal knowledge, therefore we should improve the conceptualization of the legal knowledge. It is therefore important to differentiate types of knowledge based on their role in legal case solving. Because we also think it is important to be able to address the knowledge right from the start we decided to externalize these different types of knowledge. To summarize:

Guided access using a functional differentiation of the legal knowledge:
- supports the acquisition of a conceptual structure
- enhances accessibility of the legal knowledge
- enhances readability of the legal knowledge
6.2.3 Instructional Model

In Chapter 4 we specified the instructional model for learning to solve legal cases. We defined the learning outcome using the performance - content classification matrix and selected the primary and secondary presentation forms that most likely will result in achieving this outcome.

One of the major issues here is the separation of the instructional material and the support material. Because the model only defines the instructional model in abstract and general terms we have to add the specific content and required performance, being the legal knowledge and the legal case solving components. Another issue in arranging instruction was enhancing and sustaining motivation. In Chapter 4 we reviewed the ARCS model. The motivational strategies formulated on the basis of this model should be incorporated in the instruction. This sets requirements for the type and content of the exercises and for deciding on the amount of student control, but, most important, it sets requirements for the type and content of the feedback. It is very important to present immediate feedback to keep the student from muddling and making serious mistakes. However, we think it is also very important that the student may request feedback whenever she wants. The feedback should inform her about her past, e.g. “how well did I do until this point?”, as well as on her future, e.g. “can I go on this way?”. 

6.2.4 Coaching the Student

We want the student to engage in legal case solving while the computer program monitors her activities and outcomes and corrects her during performance. This type of system is labeled coaching system. In Chapter 5 we described the characteristics of coaching systems. We committed ourselves to a non model based simulation environment. The environment in the coaching system for legal case solving consists of a simulation of a legal case by using textual descriptions. We do not use an explicit knowledge representation, because for a non model based simulation this is not required and we do not use the representation for additional functionalities either, for example, we do not keep a student model that keeps track of the student’s mastery of individual concepts. We considered some sort of curriculum planning. Curriculum planning involves some sort of sequencing of introducing topics and matching exercises with increasing complexity.
However, we decided to leave it to the student to select a specific topic and difficulty level. The decision was more or less dictated by our view on motivation, where freedom of control by the student is highly valued. The requirements summarized so far define the basic functions which now have to be specified to be able to realize them in PROSA.

6.3 Functional Specification

PROSA is an instructional environment for learning legal case solving. The system enables the student to construct a legal solution while coaching her. To be able to realize the system we now specify what and how to present the materials, how to monitor the student’s activities and results, how to determine deviations and how to present support.

6.3.1 The Legal Case Solving Task

Legal case solving involves the construction of a legal solution for a specific problem situation by applying legal rules. We want the student to be able to construct a complete and correct solution. We do not instruct an explicit method, but we do want the student to work in a systematic way to prevent them from jumping to conclusions and to keep them on track. To support the management of information we restrict the activities that have to be performed by the student. The systematic guidance and the restriction of activities are realized by externalizing the basic components in legal case solving. These components are the legal case, the legal rules and the legal solution. We therefore divide the legal case solving space in three separate areas, depicting the components (see Fig. 6.1).
In the **LegalCaseArea** a **SituationDescription**, presented as text, is accessible to the student. In the **LegalRulesArea** the legal rules are accessible to the student. Legal rules are classified based on their legal source. We use three kinds of legal sources: **Statutes**, **OtherRegulations** and **CaseLaw** or precedents. In a statute the unit of communication is an **Article**. Within an article the unit of communication is an **ArticleComponent**, an article component is a term referring to a specific type of legal knowledge.

In the **ConstructLegalSolutionArea** the student has to construct a legal solution to the question posed with the situation description.
Chapter 6 PROSA

The ConstructLegalSolutionArea is subdivided into a CaseQuestionArea, where the CaseQuestion related to the actual situation description is presented, a SelectLegalRuleArea, a SelectFactArea and a LegalSolutionArea, where the actual LegalSolution is constructed.

A LegalSolution consists of an ArgumentStructure and a Conclusion. To construct a legal solution the student has to select a legal rule, for example a statute, and select an article within the statute and bring it to the SelectLegalRuleArea, where the student can drop the article. She may either bring this article to the LegalSolutionArea, or she may start decomposing the article into article components. An applicable article is part of the solution. To select an article component from the article to see if the article component matches a fact in the situation description a fact has to be selected from the case to match the article component. This fact has to be dropped in the SelectFactArea to be able to match the article component present in the SelectLegalRuleArea. A match is part of the solution, and therefore a match is automatically copied to the LegalSolutionArea. Here the argument structure is the set of components and the conclusion is the answer. So a correct and complete legal solution consists of the complete set of components, in their correct order, concluded with the right answer. The complete set of components contains applicable articles and matched article components and facts.

The LegalCaseArea, the LegalRulesArea and the ConstructLegalSolutionArea are realized as three separate windows. There is a LegalCaseWindow, a LegalRulesWindow and a ConstructLegalSolutionWindow. In constructing a legal solution for a legal case there is a continuous interaction between the legal rules and the legal case. Therefore the ConstructLegalSolutionWindow is located in the middle. We situated the LegalCaseWindow on the right and the LegalRulesWindow on the left. However, it could also have been the other way around, there are no special requirements with regard to positioning these two windows.

In constructing a legal solution a further distinction is made between the process of solving legal cases (Process) and the legal solution (Product).
6.3.1.1 The legal case solving process

Solving a legal case is to construct a legal solution on the basis of legal rules. The activities are in the first place cognitive activities, i.e. they require the use (application) of (pieces of) domain knowledge (see Fig. 6.2).

Three areas are separated in the instructional environment on the basis of the different types of knowledge to be applied. There is a legal case area (SituationDescription), a construct legal solution area (CaseQuestion, LegalSolution) and a legal rules area (Domain knowledge). The activities to be carried out in the distinctive areas are:

- structure, translate and select fact in the legal case area.
- select article component, match article component - fact, and formulate conclusion in the construct legal solution area.
- select legal rule, select article in the legal rules area.
6.3.1.2 The legal solution

The legal solution is the outcome of the process of solving a legal case. The construction of a legal solution is a series of activities (Process) that has to result in a legal solution (Product). The completeness of a solution is closely related to the legal case solving activities. As was described in Chapter 2 a solution consists of three components being a case model, an argument structure and a conclusion. The case model is made up of the student’s understanding of the situation description both in common sense terms and in legal terms. Where the text of the situation description presents the surface structure, the case model presents the deep structure. The case model changes based on the growing understanding of the situation description by the student. The construction of a case model involves the indication of the relevant legal concepts and the use of these concepts. What legal concepts are relevant depends on the interpretation of the situation description. For example, looking at a situation description from a criminal law point of view activates an interpretation leading to distinguishing certain relevant legal concepts in the situation description different from an interpretation based on an administrative law point of view. A legal solution should contain an argument structure and a conclusion. The argument structure states the explanation of the conclusion. A legal solution is not only the conclusion, but, more important, all the components that make it possible to infer the conclusion. These components also have to be stated in the correct order. Therefore a legal solution should contain the complete set of components (Component). These components are either an Article or a match between an ArticleComponent and a Fact linked by a LinkOperator. The components should be in the correct order (ComponentOrder) and followed by the right answer (Answer).

6.3.2 The Legal Knowledge

Legal knowledge plays a determining role in solving legal cases. The major difficulties experienced in solving a legal case are related to insufficient mastery of, or insight in, the subject matter content. To enable students to acquire a conceptual structure of the legal knowledge it is necessary that they solve legal cases, while at the same time a functional differentiation of the knowledge and guided access to that knowledge is offered as support.
A functional differentiation is made by separating the situation description, the related question and the domain knowledge. The domain knowledge refers to the legal rules. Within the domain knowledge a further differentiation is made between types of knowledge. Support is offered in the form of a list of domain concepts and a topic model.

### 6.3.2.1 The legal cases

A legal case consists of a situation description, a related question and a legal solution (Chapter 3). A situation description can be viewed as a set of facts that make up a story of what has happened. Although only some of the facts may have legal relevance with regard to a particular legal source, the situation description may contain many legal irrelevant facts in order to make up a coherent, understandable story that enables the reader to infer motives of agents, causal relationships between events etc. In summary, the situation description has to be interpretable both in common sense terms and in legal terms. A SituationDescription in PROSA is presented as a text that contains sufficient facts to solve the legal issues at stake. There is no need for the student to gather and proof facts or to reconstruct the story. The student also does not have to formulate a (the) question. With every situation description presented in PROSA comes a CaseQuestion (see Fig. 6.3).

<table>
<thead>
<tr>
<th>facts</th>
<th>question available</th>
<th>formulate question</th>
</tr>
</thead>
<tbody>
<tr>
<td>all facts present</td>
<td>PROSA</td>
<td></td>
</tr>
<tr>
<td>gather facts</td>
<td>OBLIGATIO</td>
<td>moot court</td>
</tr>
</tbody>
</table>

Figure 6.3: Case types.

The legal cases in PROSA are categorized by (legal) topic (CaseTopic). The topics are the outcome of structuring the content of the domain of practice. Within each topic the legal cases are sorted from very easy to very difficult (DifficultyLevel).
Chapter 6 PROSA

The domain of practice as described in Chapter 3 resulted in the following five major topics. The principles for designing a case for a specific difficulty level were also described in Chapter 3.2

- interested party
- order
- administrative authority
- appeal
- objection before appeal

The student may select a SituationDescription of a specific CaseTopic and DifficultyLevel. This situation description is than presented in the LegalCaseWindow. The student can select a Fact from the SituationDescription. Within each situation description all relevant facts can be selected, some irrelevant facts can be selected.

6.3.2.2 The domain knowledge

The legal knowledge in a domain of practice can be found in legal sources as statutes, other regulations and case law. The student who wants to select a legal rule from one of the legal sources can choose between the three different sources Statutes, OtherRegulations and CaseLaw. Within each specific legal source we use the customary principles for organizing them. For example, the legal source category Statutes is organized on the basis of the different areas of law, where within each area the statutes are organized according to the order in which they appear in the law books students use3

2 Before putting effort in constructing virtual but realistic cases it is worthwhile to see if it is possible to use existing cases. Nicolai (1994) presents legal cases and their solutions in administrative law. However, these cases and solutions do not entirely meet our design requirements. To be able to use the cases and the solutions in the instructional environment adaptations are necessary. For example, the solutions do not show a complete argument structure, so reasoning steps need to be reconstructed.

3 We used the law books by Vermande edition 1999/2000 (Drupsteen et al., 1999).
When, for instance, a student selects the legal source Statutes, she is presented with a list of areas of law, when she selects the area of administrative law, she is presented with a list containing many administrative law acts and regulations, here she can, for instance, select the GALA. The full text of this act is then presented in the LegalRulesWindow. The student can select an Article from the act and bring it to the SelectLegalRuleArea in the ConstructLegalSolutionWindow. She then may decide to bring the article to the LegalSolutionArea of this window, or she may decide to select an ArticleComponent from the Article in order to match it with a Fact from the situation description.

6.3.2.3 The list of concepts and the topic model

The list of concepts and the topic model are the outcomes of structuring the knowledge in the domain of practice as described in Chapter 3. Both are available in PROSA. When the student wants to strengthen her prerequisite knowledge, being the knowledge necessary for solving the legal cases in the domain, she can select either one of them.

6.3.2.4 Structure and system of the statutes

All available statutes are presented as full text. However, as is described in Chapters 2 and 3 a specific difficulty in solving a legal case is to select the applicable statutes. To be able to find your way a conceptualization of the knowledge is required. System and structure of statutes should therefore be made available to the student in different formats as decision trees representing the backbone of a statute, decision trees representing the dependencies between various procedures, flow diagrams, an index (both on alphabet and on article number) and in the form of a table of contents with hyper links.

6.3.3 The Instructional Model

The instructional model determines what will be learned and how it is instructed. For law students to learn how to solve legal cases the instructional model takes the form of practicing legal case solving.
Chapter 6 PROSA

The major interaction can be described as between the system presenting the instructional material in such a form that the desired performance is being encouraged, or provoked, or that the options are constrained, and the student demonstrating the required performance (demonstrate competence). As is described in Chapter 4 a basic distinction is made between primary and secondary presentation forms. This functional distinction is realized by dividing the screen in two Layers, depicting the PrimaryPresentationForms in the UpperLayer, where the SecondaryPresentationForms is visualized in the LowerLayer of the screen.

6.3.3.1 The primary presentation forms

The primary presentation forms present the stimulus materials to elicit the construction of a legal solution for the specific legal case using legal rules. The three components distinguished in legal case solving are visualized as three separate windows in the upper layer on the screen. Each window differs in content and required performance.

6.3.3.2 The secondary presentation forms

The secondary presentation forms present learner guidance and feedback. The same three components are distinguished because the content of the support is related to the materials presented and the performance provoked in the primary presentation forms, where each differs in guidance and feedback presented. The components are realized as three separate windows on the screen. Two types of support are available in the instructional model for learning legal case solving: Elaborations and Feedback.

Elaborations have the function of a reference to prerequisite knowledge, to strengthen or consolidate prerequisite knowledge or to refresh or revive prerequisite knowledge. Elaborations are under the control of the student. Each window in the SecondaryPresentationForms has its specific types of elaborations. In the LegalCaseWindow the student has the opportunity to select from five different elaborations (CaseElaboration). There is an elaboration on the issue legal case, there is background information on structuring a legal case and translating case terms into legal terms.
An Instructional Environment for Learning to Solve Legal Cases

PROSA

There are also two domain specific elaborations being the list of domain concepts and the topic model. The list of domain concepts presents an elaborate description of the concepts in the domain of practice. For each concept the articles that refer to the concept are listed, a description of the concept is presented and a reference to related concepts is available. The topic model presents the main topics in the domain of practice in relation to each other. The model presents the sequence of the topics and references to applicable articles.

In the ConstructLegalSolutionWindow four elaborations are available to the student (ConstructLegalSolutionElaboration). The student may elaborate on the meaning of the question that comes with a legal case. There is background information available on the basic distinction between the legal case solving process and the legal solution or product. The student may also ask for a description of the legal case solving task structure.

In the LegalRulesWindow the student may ask for background information on legal rules, how to find them and how to read them, and may ask for guided access to legal rules (LegalRulesElaboration). The student may use an index (on alphabet or on article number), a table of content or an alternative representation to comprehend a legal rule and find an applicable statute or article.

**Feedback** has the function of informing the student about her performance and her solution. Information about the activities and solutions of the individual student are compared with the activities and pre-specified correct solutions defined as the standard in the instructional environment. The system gives feedback when the student performs activities that are either irrelevant or show that she does not understand the windows and their buttons. However, the feedback is basically student controlled. The student can ask feedback by requesting an Assessment. This can either be a SubAssessment, indicating that the student wants to receive information on her performance so far, after which she will proceed, or a FinalAssessment, when the student thinks she has completed the solution and wants to quit this legal case. When the student requests a SubAssessment she may ask for either an assessment of her process resulting in ProcessFeedback or for an assessment of her product leading to ProductFeedback.
Chapter 6 PROSA

The feedback is presented in the ConstructLegalSolutionWindow of the SecondaryPresentationForms. The ProcessFeedback is based on a comparison of the activities of the student with the required series of activities. There are three series of activities that are allowed by the system: one is a recommended route and there are two alternatives. Although there is no explicit prescribed method, these routes map onto a method. The routes indicate restrictions in the activities that are assessed as correct, this for the purpose of keeping the student on the track and to make some distinction between activities that are and that are not useful (see Table 6.1).

<table>
<thead>
<tr>
<th>recommended route</th>
<th>alternative route 1</th>
<th>alternative route 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>select case</td>
<td>select case</td>
<td>select case</td>
</tr>
<tr>
<td>select process option select</td>
<td>select statute</td>
<td>select process option select</td>
</tr>
<tr>
<td>select statute</td>
<td>select article</td>
<td>select article</td>
</tr>
<tr>
<td>select article</td>
<td>bring to solution</td>
<td>bring to solution</td>
</tr>
<tr>
<td>bring to solution</td>
<td>select article component</td>
<td>select article component</td>
</tr>
<tr>
<td>select fact</td>
<td>select fact</td>
<td>select fact</td>
</tr>
<tr>
<td>select article component</td>
<td>match component - fact</td>
<td>match component - fact</td>
</tr>
</tbody>
</table>

Table 6.1: The three routes.

The ProductFeedback is presented based on a comparison of the outcomes of the student with the correct and complete legal solution as available in the system. A legal solution consists of the complete set of components. These components have to be placed in the correct order. The legal solution closes with the right answer.

<table>
<thead>
<tr>
<th>Component</th>
<th>ComponentOrder</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[complete, incomplete]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[correct, incorrect]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[right, wrong]</td>
<td></td>
</tr>
</tbody>
</table>

There are eight different situations, each with there specific feedback. For example, the ideal situation is where the components are complete, the order of the components is correct and the answer is right. The worst situation is the situation with the components incomplete, the order incorrect and the answer wrong.
An Instructional Environment for Learning to Solve Legal Cases

PROSA

A second ProductFeedback is presented on the basis of a comparison of the content of a match ArticleComponent <LinkOperator> Fact indicated by the student, with the correct content of the match. There are eight different situations, each with their specific feedback. The ideal situation is where the correct ArticleComponent is matched with the proper LinkOperator to the correct Fact.

6.3.3.3 Student administration

As we indicated in Chapter 5 and in the requirements paragraph of this chapter we do not install and maintain a complete student model, but we do record some information about the individual student. This is necessary to be able to give individualized feedback and to be able to inform the student on her accomplishments. To keep an individual record we ask the student to type in her name and student number. To be able to inform the student about her sessions with PROSA we keep record of the number of cases selected by the student specified to CaseTopic en DifficultyLevel. We also keep record of the Product and the Process for each case. The records are stored and used each time the student starts working with PROSA. The student may decide to ask for the information just before quitting PROSA.

6.3.3.4 How to navigate?

Our main principle in deciding how we have the student navigate in the instructional environment is freedom of choice and control for the student. The student may do what she wants to do and whenever she wants to do it. The instructional environment allows her to look around and to examine each and every detail. There are no time restrictions, the student may take as long as she wants to solve a case, she may even decide not to solve it at all or not completely. However, although the student may do what she wants, there is only one complete and correct legal solution for each case and there are three pre specified routes that are regarded as the best ways to proceed. So in the feedback the student will be informed about the deviations concerning both the route and the content of the legal solution. She will also receive feedback whenever she carries out an activity or uses some knowledge element that will lead her nowhere. However, it is up to the student to do something with the information or not.
As we described in Chapter 5, the student can not type in text, text can be manipulated using copy and paste. We will use buttons to enable the student to select a specific activity or certain types of knowledge. A button in turn contains pop up menus each showing a list of specific options to select from. The legal case and the legal rules are presented as text in the specific windows. Because there are space limits it is inevitable, particularly with the legal rules, to scroll text.

6.3.4 Architecture

The architecture of a system describes the structure of a system in terms of processing components and data structures. As we described in Chapter 5 we will not use an explicit knowledge representation to realize our system. This has consequences for the architecture as well. Our architecture is less modular and less transparent because we do not separate content (knowledge) and control (reasoning) as is the case in knowledge based systems.

6.3.4.1 Flow of Control

Authorware is a visual, icon-based authoring environment. A program is made by assembling icons on what is called the flow line. Different types of icons represent different types of objects like text, graphics or a set of instructions and herewith the content of a program. In Fig. 6.4 the top level of the architecture of PROSA is shown.

Figure 6.4: Top level architecture of PROSA.
The architecture as a ‘control of flow’ diagram is explicit in an Authorware program. The icons on the main flow line are visible at this level. When PROSA is run, the icons are executed from top to bottom along the flow line. We discuss the icons in the implementation paragraph.

6.3.4.2 Flow of Data

The architecture as a data flow diagram is depicted in Fig. 6.5. The input of the student, activity outcome, is compared with a standard to detect deviations. These deviations are recorded by PROSA to be presented to the student. The rounded boxes denote data, the rectangles denote components and the arrows depict the data flow.

![Data flow diagram of PROSA](image)

Figure 6.5: Data flow of PROSA.
There is an input component where the student engages in solving a legal case. The activity outcome is compared to the standard, being the routes (process) and the correct and complete legal solution (product). The outcome of the comparison may give rise to actions by the PROSA coach (feedback). The outcome is also registered in the student's administration.

6.3.4.3 Visualization

The interface of PROSA visualizes the instructional environment we present to the student to learn legal case solving. PROSA does not explicitly instruct a method. However, the design of the screen constraints the ways the solution can be constructed. We argued that it might be more supportive to present an environment in which the basic legal case solving components are externalized. This way the student is not enforced to work in a pre-specified way, however, she does have something to go by that may support her to work in a systematic way. Externalization may also take over cognitive activities from the student that hinder correct task performance. For example, intermediate results no longer have to be administrated internally, the results can be left in a specific window on the screen in this way diminishing the students' memory load. She also does not have to check data and intermediate results "by heart".

The leading principle in designing PROSA is "divide & conquer". We not only made a distinction within legal case solving between legal case solving method and legal knowledge, we also distinguished between types of legal knowledge, which in turn dictate distinctive components in legal case solving. In the instructional model we distinguished between instructional material and support. These distinctions were realized in the interface in such a way that it presents students with an environment that makes it easy to "conquer" legal case solving. This is accomplished by a spatial design of the interface (see Fig. 6.6). We opted for a fixed composition of the screen. This way the student can easily recognize the legal case solving components, their content and their functionality which in turn may support a systematic approach to solving a case.
The screen is divided into two horizontal layers. The primary presentation forms are presented in the upper part of the screen, the PrimaryPresentationForms. The secondary presentation forms are presented in the lower part of the screen, the SecondaryPresentationForms.

In this way a clear distinction is maintained between the presentation of the content and the expected performance on the one hand and the presentation of support on the other. The use of color has functionality in distinguishing the subsequent components that play a role in legal case solving.
Chapter 6 PROSA

There are three different components, so there are three different windows with each a specific color. The LegalCaseWindow is yellow, the ConstructLegalSolutionWindow is orange and the LegalRulesWindow is light blue. The distinction between the presentation of the materials and the presentation of the support is expressed using bright colors for the windows where the materials are presented and using pastel colors for the windows where the support is presented. The LegalRulesWindow contains the legal rules, i.e. the ‘theory’ that should be applied to the legal case description in the LegalCaseWindow in the upper layer. The student can select a legal case (legal case button) and legal rules (legal rules button). The middle window in the upper layer the ConstructLegalSolutionWindow is where the student constructs her legal solution by matching selected article components with selected facts (the process button contains the select and the match options, the product button contain facilities to ‘edit’ the legal solution). The specific problem posed to the student is put at the top of this window. The students’ workspace allows her to keep track of her local decisions. Because there is no prescribed method or order to the way she matches legal rules to facts, the student may work both ‘theory’ driven or ‘case’ driven. Therefore, in the end the student is capable to come to a conclusion on the basis of the argument structure. The legal solution is the actual work space of the student. However, we have to deal with the fact that our space on a computer screen is limited. This may result in a rather small work space where it may be difficult for the student to keep an overview. Therefore we introduce an option ‘large screen’ under control of the student for requesting the larger workspace. The student may ask feedback (assess button), which in turn is presented in the lower layer in the ConstructLegalSolutionWindow. The student may also ask for elaborations (support buttons) on the legal case, the legal rules or the legal solution to be constructed. This support is then presented in the respective window in the lower layer. In paragraph 6.6 we describe a session with PROSA to illustrate how the student is coached in solving a legal case.
6.4 Implementation

Authorware is used for implementing PROSA. Authorware is an authoring (programming) environment for creating and publishing interactive information and can be used for the construction of interactive learning and training applications. Authorware provides tools for creating multimedia presentations that use digital movies, sound, animation, text and graphics to engage the user in the learning process.

We now describe in more detail the realization of the flow of control as depicted in Fig. 6.4. We introduce these distinctive parts in the order they appear in the figure. For a more detailed description of the implementation we refer to Muntjewerff & Groothuismink (1999a).

Initialise The first icon that is executed is the map icon ‘initialize’. This map icon contains a number of display icons that contain the first screen of PROSA, the so-called start screen. Furthermore the variables used in PROSA are defined and initialized in a calculation icon. These variables are used to keep track of the students’ actions and to store and use general PROSA information like available items for the menu buttons. And finally the ‘initialize’ icon contains display icons which contain the instructional environment screen with the three windows in both layers (see Fig. 6.6). This ‘initialize’ icon is executed only once per session.

Perpetual interactions The second icon ‘perpetual interactions’ contains two kinds of so called perpetual interactions. These are called perpetual because they remain active all the time. An interaction is accessed by an interaction icon with different types of response type symbols attached to it. These response type symbols tell the interaction icon whether to display a button, a menu, a text-field or some other element. An interaction monitors the actions of the student and sends that information to the response type symbols attached to it. If a perpetual interaction is encountered the interaction is activated.

Authorware 4 copyright ©1997 Macromedia, Inc. San Francisco, California, USA. In multimedia environments Authorware is considered the standard. Authorware is maintained and extended continuously. Recently it has been extended with an option to present Authorware applications on the internet (Shockwave). See for more information http://www.macromedia.com/software/authorware
Chapter 6 PROSA

The first perpetual interaction displays the menu buttons available for the six different windows and defines the reaction of PROSA when the student uses the buttons. The student can use these buttons throughout the whole session of solving a legal case. The second perpetual interaction defines the feedback of PROSA to the students' activities regarding the construction of a legal solution in the parts select legal rule, select fact and legal solution in the construct legal solution window. An example of such a student activity is pasting an article in the ‘select rule’ part of the construct legal solution window.

**Legal case** After the perpetual interactions the main loop of the program occurs, the so-called decision icon ‘legal case’ with map icons for every case topic attached to it. Within each case topic map a similar decision icon for the different levels of difficulty is used. At a decision icon the program branches to a path according to the criteria specified. In PROSA these criteria are the choices the student makes. The student uses the menu button legal case and the choices she makes are stored in two variables CaseTopic and DifficultyLevel. On the basis of these variables Authorware first branches to the map icon of the chosen case topic and then to the map icon of the chosen difficulty level. A difficulty level map icon contains (1) display icons which contain the legal case text and the accessory question (2) a calculation icon in which the correct legal solution is stored in a variable (3) an interaction icon to monitor the students' activities specific to the case selected. In this way PROSA is able to give case- and student specific feedback. Also many student characteristics can be recorded per case, like, for instance, the sequence of the student's activities in solving the legal case, the legal solution the student constructs and the cases the student selects. These student characteristics are an example of characteristics that are recorded and maintained during all sessions of the student working with PROSA. In this way a student history is built to be able to adapt to the individual students' activities and to evaluate the individual student.

**Subroutines** The last icon at the top level flow line is the map icon ‘subroutines’.
An Instructional Environment for Learning to Solve Legal Cases

This icon contains a number of subroutines implemented as map icons attached to framework icons. These subroutines appear only once in PROSA, but are called many times by various parts of the program. There are subroutines for checking the legal rules, for checking the legal solution, for checking the legal case, for checking the drag and drop functions, for example, drag an article from a statute to place it in the legal solution part, and for presenting the feedback to the student. An example of a subroutine is adding a student activity to the list that is used to keep track of the series of activities the student carries out to construct a legal case solution.

6.5 Maintenance and Re-Use

Although maintenance and re-use may be classified as basic requirements, we did not discuss these issues up to this point. This is partially due to the fact that both issues are closely related to design and implementation, partially due the fact that it is more or less incorporated in our handling of the other requirements. Our analyses of legal case solving, the domain of practice and arranging instruction resulted in abstract models that can be reused as well as maintained. The way the domain of practice is modeled, for example, provides us with guidelines for adding new knowledge, or deleting knowledge that is out of date. The fundamental approach we took in setting up these requirements was continued in the specification and the implementation. We are arranging instruction in a field where the knowledge is liable to minor and major changes due to decisions by the legislator or the administration of justice. To be able to test our claims we had to set limits to both the amount and types of legal knowledge as to the number and topics of legal cases we could include in the system. All this made us very susceptible to the issues of maintenance and re-use. Here we restrict the description to the way in which we attended these issues in the architecture and the implementation.

6.5.1 Maintenance

Maintaining a system as PROSA, requires that the system can be changed. If the system can be changed it is possible to repair mistakes and to add or delete materials, laws change. It is also necessary that changes can be made without too much costs and effort.
Chapter 6 PROSA

It is important that the cause of a mistake can be detected and corrected easily, that materials that are out of date can be deleted without causing problems elsewhere in the system, and that new materials can be added without causing difficulties in other parts of the system. Transparency of architecture and specific tools may facilitate maintenance. To enhance the transparency of the architecture for implementing PROSA we used levels and modules. The main loop of PROSA is depicted on the top level (see Fig. 6.4). Each part depicted on this top level is in turn specified following a structure of levels and modules. Take, for instance, the way in which the legal cases are available in PROSA. The legal cases are grouped by topic and are placed in isolated maps. Selecting a specific topic, for instance interested party, brings us to level 2 which is entirely concerned with interested party cases, showing separate maps for each difficulty level. Selecting, for instance, the map interested party easy cases brings us to level 3, where we find a map for each easy interested party case. Opening a specific map takes us to level 4 where we find the map containing separate display icons for the text of the situation description, the question related to this situation description and a calculation icon containing the correct and complete legal solution which are displayed on level 5. The way PROSA is structured makes it possible to correct mistakes in existing situation descriptions, questions and solutions and to add legal situation descriptions on existing topics and difficulty levels. The same can be said for the legal rules. Here the structure of modules and levels of increasing specificity are also applied making correcting, adding or deleting entities feasible. We did not construct specific maintenance tools, mainly because we could manage with the tools available in Authorware and because editing display icons is also rather easy.

6.5.2 Re-Use

Solving legal cases requires the availability of legal knowledge. This knowledge can be found in the legal sources. Legal sources are grouped to areas of law they refer to. The basic areas of law are public and private law. Within these areas different fields are distinguished, for example, within public law we distinguish constitutional law, administrative law and penal law. To limit ourselves we restricted the domain of practice in PROSA to administrative law, and even more specific to general administrative law.
An Instructional Environment for Learning to Solve Legal Cases

However, many other domains of practice are available and we even may think of combinations of domains. Where we defined adding more knowledge of this domain as maintenance, we define the introduction of another legal domain of practice as re-use. The use of levels and modules in the architecture for implementing PROSA not only supports maintenance, but also supports re-use. However, for real re-use generality of design is also required. This is realized by using a domain independent design. PROSA therefore uses basic components as legal case, legal solution and legal rules as well as basic elements in the solution as component, component order and answer. For example, PROSA uses separate modules for each category of legal rules, being statutes, other regulations and case law. If we take the category statutes new acts can be added easily using a display icon for each individual act. Furthermore, legal cases of new topics and difficulty levels can be added. For each new case topic and difficulty level a new map icon containing the case specific display, calculation and interaction icons have to be added. We did not construct specific tools for re-use either, because the tools available in Authorware are adequate.

6.6 A Session with PROSA

To get a basic idea of the functionality of the system we now describe a session with PROSA. The description of the session is based on the recommended route. Starting PROSA brings us to the first screen which shows us the PROSA logo and four buttons. There is a start PROSA button, an explain PROSA button, an info PROSA button and a stop PROSA button. The start PROSA button brings us to the data request screen where we have to insert our name and student number. This is required so PROSA can keep our individual record. When we indicated that we are ready the PROSA screen appears (see Fig. 6.6).

Imagine we are sitting in front of PROSA. The first we have to do is to select a legal case from the set of available legal cases using the menu button legal case. The legal cases in PROSA are arranged by topic. We decide to select a case with topic interested party from the list of topics that pops up. Within each topic the legal cases are arranged by level of difficulty. We decide to select difficulty level easy from the list.
The situation description selected is then presented in the upper layer in the legal case window (e.g. the Dapper Market case) (see Fig. 6.7). At the same time the question that belongs to the situation description is presented in the upper layer in the construct legal solution window (e.g. Is Alexander Boer an interested party according to the General Administrative Law Act?). We now select the menu button process in the upper layer in the construct legal solution window. The list with the two activities select and match pops up. Being presented with a legal case the next thing to do is to select either a legal rule or a fact from the legal case. So to start constructing the legal solution we have to choose the select option. This results in a change in the construct legal solution screen. The distinction between selecting a legal rule from the set of available legal rules and selecting a fact from the situation description is now visualized. There also appears a specific part in the construct legal solution window that is titled legal solution. This is where we have to put our intermediate results to construct our legal solution. We select a legal rule by choosing the legal rules button in the upper layer in the legal rules window. This button shows the three different categories of legal rules: statutes, other regulations and case law. Within the statute option a further classification of statutes is made based on the area of law the statutes belong to. We choose the option statutes from the legal rules button and then select the act we think applicable given the specific legal case and question to be answered. This act is presented in the legal rules window (e.g. the General Act of Administrative Law). We now have to select an applicable article from this act. This article has to be copied to the construct legal solution window, in the specific part select legal rule (e.g. Interested party means the person whose interest is directly affected by an order). We may bring the article to the legal solution using the product option bring to solution. We now have to select an article component from the article (e.g. the person) and a fact from the situation description to be matched to the article component (e.g. Alexander Boer). We have to use the match option available in the process button to link the article component to the fact. The match option shows us the available link operators that we can use. Because we argue that the person is Alexander Boer we opt for the operator '='. Our match is automatically put into the legal solution (e.g. the person = Alexander Boer). We have to repeat the select activity until there are no statutes, articles, article components and facts left. The match activity has to be repeated until there are no more article components or facts.
At that stage we have to formulate the final answer to the question. We choose the option **formulate answer** in the menu button **product** and select what we think is the right answer (e.g. A. Boer is not an interested party in the meaning of the GALA). One by one the various elements where put on the screen. At this point we are facing the following screen (see Fig. 6.7).

Figure 6.7: Construct legal solution.
Chapter 6 PROSA

We notice that up till now all materials are presented in the windows of the upper layer. This indicates that we did not request an assessment or an elaboration and that there also was no need for PROSA to correct us.

Since many years there is the Dapper Market in the Dapperstieet in the district Zeeburg. The qualified administrative authority takes an order as meant by the General Administrative Law Act to run the Dapper Market also on Sundays. Alexander Boer who lives in the district Amsterdam Oud Zuid, in De Liessestieet does not like it that the Dapper Market will be open on Sundays as well. He is a light sleeper, his health might be in danger. He makes an objection against the order of the district Zeeburg.

Therefore we now assume that we do not know where to start when we are confronted with the Dapper Market case.
We understand that we have to find out if Alexander Boer is an interested party in the meaning of the GALA, however, we do not know where we may find the legal knowledge. We therefore decide to ask for an elaboration by using the support button of the legal case window in the lower layer. The buttons have different options available as a list of concepts and a topic model. We select the 'list of concepts' option resulting in the presentation of an alphabetical list of terms used in the domain of practice. After selecting the term 'interested party' we get a description of the article components (the legal terms) and a reference to the legal rule. When we click on this reference the legal rule is presented in the lower layer of the legal rules window (see Fig. 6.8).

We not only may ask for elaborations, we may also ask for an assessment of our (intermediate) results using the assess button in the construct legal solution window. The two types of assessment available are sub assessment and final assessment. When we are sure we want to quit working on the particular case we may ask a final assessment, however, when we want to proceed but also want to have feedback on how we are doing we may ask for a sub assessment. The final assessment will present an overview of what we did right and what we did wrong subdivided to process and product, and within the product specifying the status of the components, component order and answer. A sub assessment gives us the opportunity to get separate feedback on our route (process) and on our constructed solution (product). We may request an assessment any time we want. Fig. 6.9 shows a sub assessment of our product.
When we want to quit working with PROSA we select the stop button which brings us to the stop screen. Here we have different options. We may exit PROSA, we may go back to working with PROSA and we may, before we select one of these options, ask to look at our results. When we choose to look at our results we are presented with our PROSA history.
This overview shows how many sessions we had with PROSA, how many cases we solved in each session specified to the topic and difficulty level and for the most recent session it specifies for each case our achievements both on product and process.

6.7 Conclusion

In this chapter we specified the requirements for PROSA and implemented the system. PROSA is an instructional environment for learning legal case solving. When sitting in front of PROSA in what way is what a student has to do different from the way she is used to solve a legal case (using printed materials and her memory)? It is for certain that in solving a legal case the student is running out of memory quite fast. Therefore students make notes. However, making these notes gives lot of work, students have to copy text which is inefficient, and then these notes are often incomplete. For one PROSA does not put as heavy a load on a students memory as traditional written legal case solving does. The way of working becomes quite different because PROSA takes over the managing of information by externalizing materials, intermediate steps and intermediate results in an automatic way. The student can just start working with PROSA and her legal solution evolves on the way. Secondly PROSA facilitates the acquisition of a systematic approach in solving a legal case by creating a path through the knowledge by differentiating the knowledge on the basis of its function in legal case solving. The legal case is separated from the legal rules and the legal solution is divided into partial components. Working with PROSA is therefore more efficient. However, is being more efficient also being better? It is, because the most important factor in problem solving is what is expressed as “going through the problem”. The more a student actively engages in legal case solving the more she learns to differentiate the knowledge, the more systematic her approach will become and the better a legal case solver she will become. Facilitating the problem solving process, as PROSA does, results in knowledge differentiation and a systematic approach in a more efficient way than when solving legal cases in the traditional way.
Chapter 6 PROSA

PROSA presents an environment (not a method) in which:

- the student is facilitated and encouraged to work in a systematic way, the chances to miss or leave out something are nil
- the student does not have to manage her information and she does not have to keep track, the coach takes care of keeping track

PROSA is a try out. In PROSA the student is supported by presenting guided access to the knowledge of the domain of practice instead of instructing an explicit method. We have to gather information to be able to test our hypothesis concerning the important, even determining, role of domain knowledge in legal case solving. Therefore the evaluation of the effectiveness of PROSA is the subject of the next chapter (Chapter 7).