A presentation and navigation system for the Multimedia and Education project

Institute for Perception Research
PO Box 513, 5600 MB Eindhoven

04.01.1996

Rapport no. 1088

A presentation and navigation system for the Multimedia and Education project

F. Uittenbogaard

Voor akkoord: Prof.ir. S.P.J. Landsbergen

A presentation and navigation system for the Multimedia and Education project

F. Uittenbogaard

1 Introduction

This document describes the first prototype of a presentation system for multimedia course material. It is developed within the M&E (Multimedia & Education) project [van Loon,1995]. The system provides structured ways of browsing through course material, and assists users in finding the best way to obtain the information they need. The system is based on a subset of the information model for multimedia course development, designed at IST/IT and IST/IPO.
This prototype is a first step towards a system which lets the student take the initiative and responsibility for the acquisition process. The user should be able to define his/her own goals, tempo, and choice of topics, of course obeying external constraints (if present). In a sense, this approach is complementary to the techniques developed in the APPEAL project (see [Masthoff,van Hoe,1994]), where particularly the system determines the course of events. In that project, flexibility is obtained by applying an adaptive 'teaching engine', which continuously adjusts the parameters mentioned, dependent on the student's results. Both approaches, especially in combination, promise to yield more flexible and effective educational systems. User-initiated navigation seems to have an even wider area of application, namely all systems where transfer of information to the user is a major function (content-based applications).
The techniques for navigation which are described in this document make use of the explicit structure of the information model. Other strategies for navigation are based on content (implicit structure, e.g. text retrieval) or hyperlinks (ad-hoc structure, see for example [Nielsen,1990]). These are valuable and complementary approaches, but as research topics they fall outside the scope of the M&E project.
In this document I will describe the relevant subset of the information model, the current functionality and limitations of the prototype. The last section will be devoted to architectural considerations and future developments.

2 The information model

Only the most important aspects of the M&E information model are supported at the moment. See [ter Horst,van Loon,1995] for a complete description of the model. Figure 2 shows the model underlying the presentation system. The course material is decomposed into *units*, which are hierarchically organized through the *subunit* relationships. The model allows units to have multiple superunits. Units have zero or more *preconditions* and at least one *postcondition*, both of type *goal*. Goals are primitive objects without internal structure

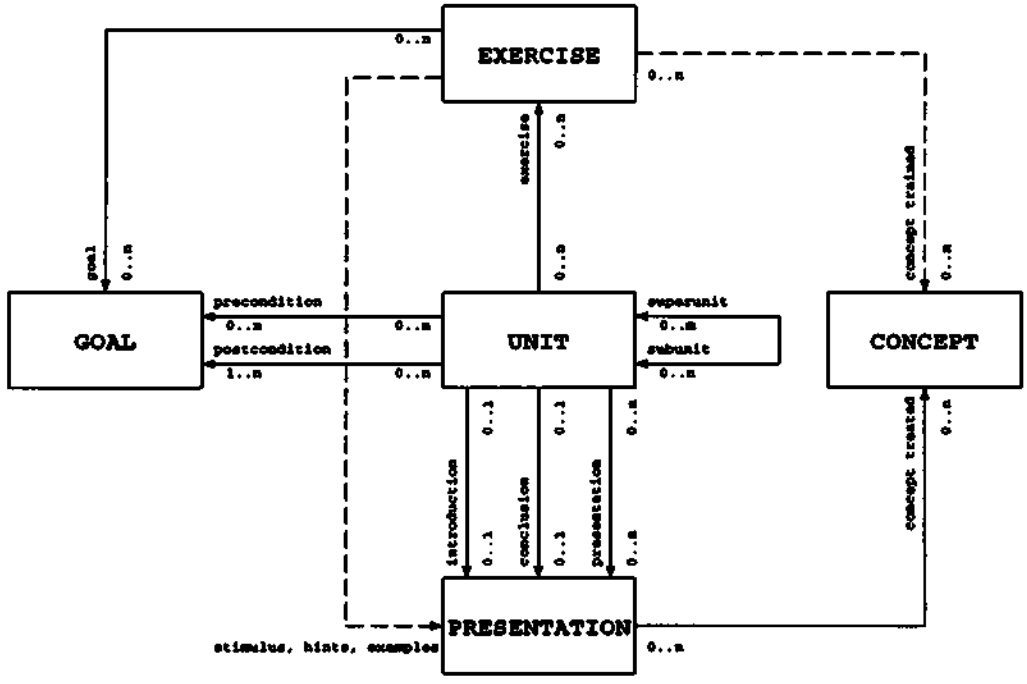

Figure 1: Information model underlying the presentation system

or direct mutual relationships. Their meaning is only indicated informally by their name. Preconditions also apply to all subunits, while postconditions of subunits also apply to their superunit.
Units which have subunits are called composite units. They cannot have exercises, in contrast with basic (leaf) units. Units and exercises frequently make use of *presentations* to convey their content. In principle, this can be any type of multi-media information. However, the system currently supports only still pictures (in various formats) and sound. For other types, external tools will be used.
Both presentations and exercises can be related to *concepts*, which indicate the topic to be treated or trained respectively. Concepts are primitive objects, like goals. They are not formally related to goals, although in practice there is often a strong correspondence. Concepts are interrelated only indirectly, via presentations, exercises and units.
The dashed lines in figure 2 mean that these relationships are not yet fully supported by the presentation system.

3 System description

This section describes the first prototype presentation system. First, some implementation issues will be treated. Next, the various views on the content will be explained, and the way navigation is supported by these views. It will be shown how pictures and sound are combined, and finally a survey of limitations and possible improvements will be given.

3.1 Implementation issues

The system is written in C on a Unix/X-Windows platform. The user interface was developed using X-Designer. The system makes use of code generated by the Elegant compiler generator, for parsing a content-structure file. Structure files contain textual specifications obeying strict syntax rules. Elegant ([Augusteijn,1990]) was used to generate a compiler for these rules, which performs the necessary syntactical and semantical checks. The resulting data-structures are accessed by the presentation system. Dynamic information, such as the student's progress, is stored separately, with references to the static data. The advantage of this approach is, that a new parser-module is available as soon as adaptations to the model have been formalized. However, the presentation system will have to be changed 'manually'. Obviously this is a drawback. In chapter 4 will be indicated how this problem could be solved.
At the moment all the structure-information resides in main memory. Pictures, sound etc. are stored in separate files. In the future, a more database-oriented approach will be necessary, to be able to handle large course structures.

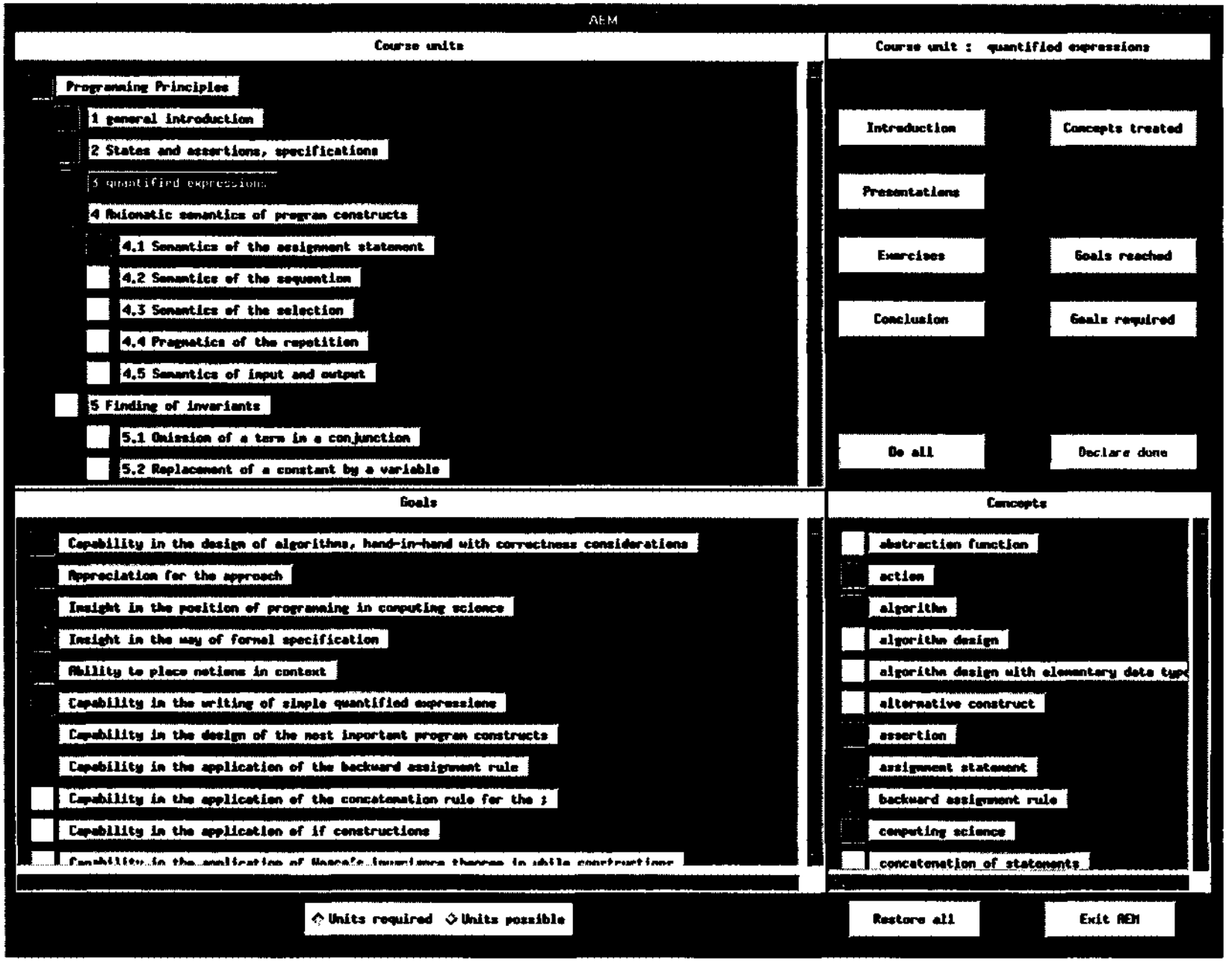

Figure 2: Main screen of the presentation system

3.2 The main screen

Figure 3.2 shows the main screen of the presentation system. It has 4 areas. The upper right area represents one course unit. The other areas present 3 different views on the course structure.

3.2.1 The unit-window

The unit-window has buttons for starting presentations and exercises, following links and changing the dynamic status of the unit.
The buttons *Introduction* and *Conclusion* are used to start the related presentation. If there is no presentation, the buttons are shaded. The same applies to the *Presentations* button, but in case of multiple presentations, an intermediate pick-list is presented, frow which a presentation can be selected. Pressing *Exercises* causes an exercise-window to be displayed, containing buttons for viewing the stimulus, hints, goals etc.
The *Do all* button is meant to successively start all presentations and exercises, after which the unit is supposed to be *done*. The user can also declare the unit done, by pressing the *Declare done* button. The system will assume that the user has mastered the unit's material in some way.
The three remaining buttons will alter the contents of the *Goals*- and *Concepts*-views, which will be described below.

3.2.2 The units-view

The left upper area contains an indented list of course-units, representing the hierarchical unit structure. It roughly corresponds to the contents in a normal book. Each entry consists of a name, a number and a status indicator. Clicking on a unit's name will make it the *current unit*. It will be highlighted and presented in the unit-window.
Numbers are assigned dynamically by the presentation system, enabling units to be moved around freely during development. A unit might also occur more than once. The top unit, representing the course as a whole, does not get a number.
The status indicator discerns three cases. A black indicator means that this unit has been done already. A grey one means that the student has the prerequisite knowledge to do the unit. All other units have a white indicator. The user is not yet qualified to do these units but he/she is free to look ahead.
Initially this view shows all units, but this can be influenced by events in the other windows. The original content of all views can be restored with the *Restore all* button.

3.2.3 The concepts-view

The right lower area contains a list of concepts. This view is comparable to the index in a normal book. Each entry consists of a name and a status indicator. Clicking on a name causes the units-view to show only those units which treat that particular concept. Because concepts are related to exercises and presentations, the selected units should contain at least one exercise or presentation

which refer to the concept. At the moment only the link to presentations is implemented. The indicators have the same function as for the units-view. A concept is supposed to be done (i.e. 'mastered'), if at least one course-unit, which contains at least one presentation or exercise which treats or trains the concept, is done.
Although the units-view will probably not reflect the unit hierarchy anymore, the original order, indentation and the original numbers are preserved, to give the user an idea of the units' position in the course. For the rest, the units-view's functionality remains the same.
Initially, the concepts-view contains all concepts, in the order of their definition. When the *Concepts treated* button in the unit-window is pressed, the concepts-view will show only concepts which are treated or trained by at least one of the unit's presentations or exercises. The original contents can be restored in the normal way.

3.2.4 The goals-view

The lower left area contains a list of goals. This view has no direct analogue in conventional books. Each entry consists of a name and a status indicator. The indicator works the same as in the concepts-view. A goal is done (i.e. 'reached'), if at least one course-unit for which the goal is a postcondition is done.
This view has two modes: *Units required* and *Units possible*. In the first mode, clicking on a goal's name causes the units-view to show only those units which are required to reach the goal. In the second mode units are selected having the goal as a precondition.
Initially, the goals-view contains all goals, in the order of their definition. When the *Goals reached* button in the unit-window is pressed, the goals-view will show only the goals of the current unit. At the same time the goals-view's mode is set to *Units possible*, creating a kind of 'forward' mode. In this mode, the user can easily travel from elementary to more advanced parts of the course. The reverse situation arises if the *Goals required* button is pressed. The original contents of this view can be restored with the *Restore all* button.

3.3 Presentations

One of the goals of the M&E project is to make the step from conventional courses to computer-based applications as smooth as possible. Courses which were originally meant to be given in lecture-form, can be quickly converted by presenting the overhead-slides on the computer screen, along with spoken explanations.
Figure 3.3 shows an example of a presentation. It comprises a slide and some navigation buttons. If a slide comes with a sound file, the picture remains on the screen as long as sound is produced. Then the next slide is shown or the system returns to the previous level, which in most cases is a pick-list of presentations with the current position highlighted. The user can browse through the presentations via the pick-list, or at the lower level via the previous and next buttons. If there is only one presentation, e.g. a unit's introduction, then the

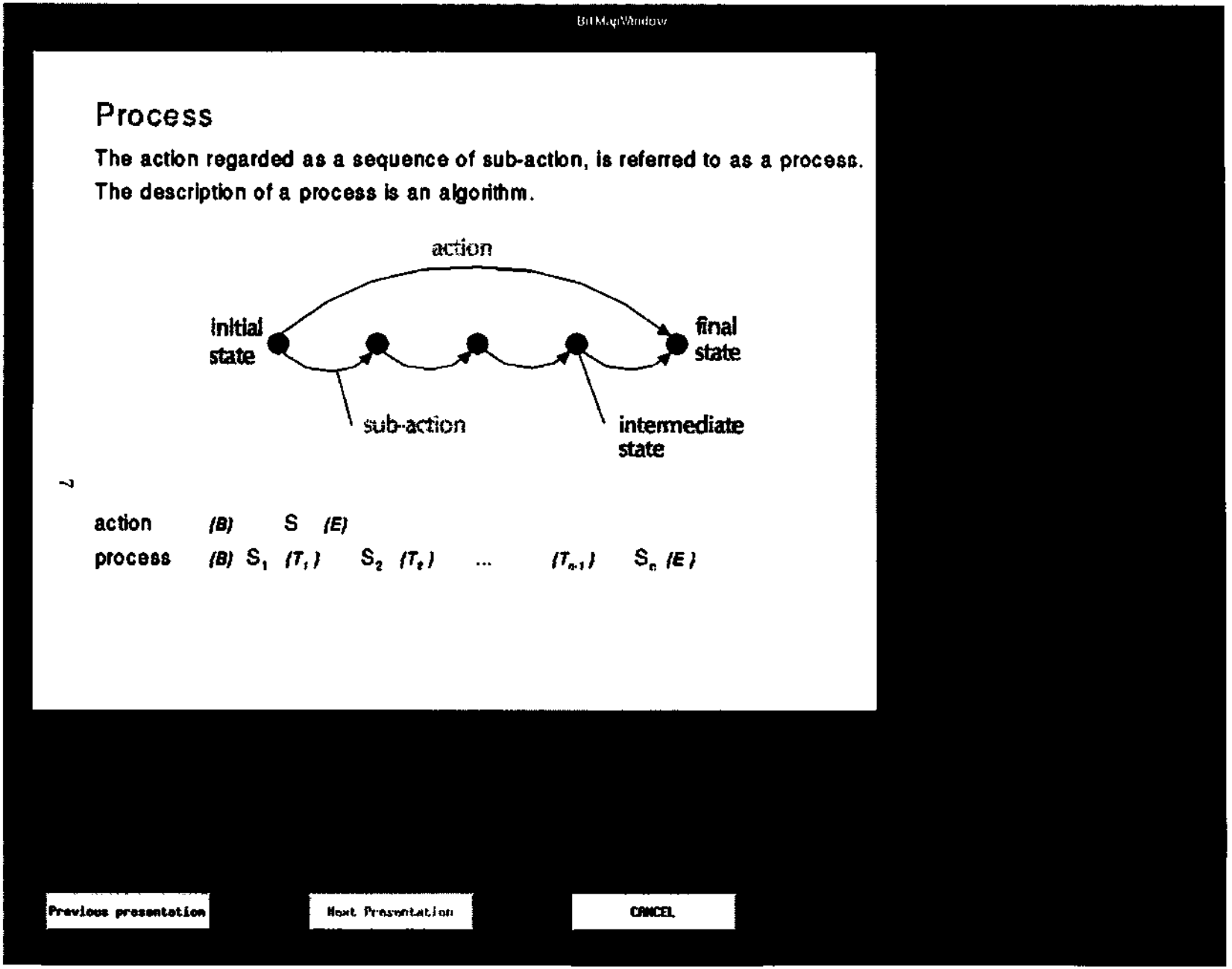

Figure 3: Example of a presentation

pick-list is skipped.
Animation within slides is achieved by successively showing pictures with only small differences, e.g. a pointer moving to the next bullet. If there is no sound file, a picture remains on the screen until one of the navigation buttons is pressed.

3.4 Current limitations and possible improvements

This first prototype was built to demonstrate some ideas with respect to structuring and visualizing course material. It implements a kind of book metaphor (units, table of contents, index), extended with multimedia presentations, goal-driven navigation and progress monitoring. Two courses have been converted to the M&E format, with surprisingly little effort (one of the courses is described in [Albertsmeier,1995]).
First experiences show that these kind of structuring and navigation mechanisms are a valuable addition to conventional courses. However, a lot of things need to be improved, before serious evaluation can take place. This paragraph discusses the elements which can be improved, without violating the basic design principles underlying this prototype. In the next section a more general view will be presented, with respect to applying these techniques within an educational context, or even wider.

3.4.1 Omissions

First of all, the prototype does not yet support the complete M&E information model ([ter Horst,van Loon,1995]). A lot of aspects, such as feedback types, abstraction levels, and support for exercises, are not (completely) incorporated. Only a few presentation types are allowed (no video for instance), and there is no mechanism for storing and retrieving student results. No educational knowledge is included at the moment, but integration with an APPEAL-like teaching engine is foreseen (see section 4).

3.4.2 Model-related issues

Some details of the information model have not yet been worked out properly. For example, the relation between units and subunits is not clear. The current prototype assumes superunits to be units in their own right, so they can be 'done' independently from their subunits. If units would be considered as just containers for their subunits, this would have to change. It would also affect navigation, e.g. doing a unit, would mean presenting the introduction, presenting all subunits recursively, doing the exercises and finally showing conluding remarks.

Another point is animation of pictures. This is now achieved by a sequence of presentations, synchronized with the accompanying sound. The differences

between these presentations are too small to be named, but yet names are required for the presentation pick-list.

Finally, concepts and goals are still weakly defined. They are primitive objects without internal structure. There is no way of defining semantic relationships between members of both types, e.g. specialization-generalization, goal-subgoal and goal-concept relationships. This severely restricts their usefulness for the presentation system. Especially for large domains, the two related views will become unmanageable long list of unrelated items.

3.4.3 Screen design

The current unit-window is not very informative. Only the title gives an indication of the current unit. In stead of buttons activating pop-up lists of exercises and presentations, those lists could be included in the unit-window directly. The same could be done with treated concepts, preconditions and postconditions, which would also make navigation more efficient. These measures would require the unit-window to become more prominent. The three views could be made either smaller, overlapping or optional. The units-view could be made more surveyable by enabling subtrees to be hidden, similar to many graphically oriented file managers.

3.4.4 Navigation

Only units can be (declared) done at the moment. The status of goals and concepts is derived from the status of units. However, it is quite conceivable, that the user wants to change the status of goals and concepts directly. One way to achieve this, is to implement the indicators as toggle buttons. Now, the system should also be able to derive a unit's state from the state of related goals and concepts. As has been mentioned in paragraph 3.4.2 a unit's status might also depend on the status of its subunits.

Some navigation principles could have been implemented more consistently. This is illustrated with two examples:

Default action on single valued lists: e.g. if a unit contains only one presentation, it is started immediately, while otherwise a selection must be made from a pick-list. This principle can also be applied to other functions, e.g. if the units-view shows only one item, it can be made the current one immediately.

Two-level navigation through ordered lists and trees: Presentations can be browsed through at two levels. On the one hand they can be picked from a scrollable list, on the other hand from within each single presentation the previous and the next one can be reached immediately. One would expect this principle to be applied to exercises and units as well. An obvious extension would be buttons for the first and the last element.

For tree structures, such as the unit hierarchy, additional buttons for the parent and the first child could be convenient.

Other strategies for navigation could be considered. Examples are text retrieval, hyperlinks, content based searching etc. However, as research topics they fall outside the scope of the M&E project. In section 4.2 the topic of (particularly model-based) navigation will be further elaborated.

4 Architectural issues and future directions

This section deals with the question how the functionality offered by this prototype can be integrated into a computer-based educational system. The discussion is split up into two parts. The first part describes the interaction with a teaching engine and the second part discusses the integration with an authoring environment.

4.1 Interaction between presentation system and teaching engine

As explained in the introduction, both types of systems have opposite goals. A presentation system enables the user to select the course parts to be done next. A teaching system tries to figure that out automatically, dependent on the student's results. Both approaches have their pros and cons. A combination of the two would be optimal, yielding a kind of mixed-initiative interface.
In the M&E approach, the presentation system is the starting point of a migration path. After defining the basic course structures, it is guaranteed that the result can be handled by the presentation system. After that, the necessary information to make the system more adaptive can be added gradually.
Which component is in control, depends on the situation. A student following a course from begin to end will probably prefer to be guided by the system as much as possible. If a courses comprises large amounts of heterogeneous information (hence not intended to be mastered as a whole), or if a course is used as a reference source, the student needs to be able to find his own path through the information space.

The following setup seems flexible enough to meet the diverging requirements mentioned above:

- The author may specify a number of default goals, which the student has to achieve in order to finish the course successfully. Some of the goals might be marked as optional or as arbitrary members of a set. For example, the following goals might be specified:
  - Goal1
  - Two goals from {Goal2, Goal3, Goal4}
  - Goal5 (optional)

- The teaching system will try to reach the default goals. If the system cannot proceed, either because it lacks the educational knowledge to decide upon the next step, either because it encounters optional or alternative goals, the user is offered a restricted choice. If all default goals have been reached, the system falls back on the presentation system, enabling the student to read back or choose new goals.

- At any time the teaching system may be interrupted, after which the presentation system takes over the control. The user then has the opportunity to examine his progress, read back or ahead, specify other or additional goals, or ask for extra information about a specific topic.

- If the author did not specify any default goals, the system falls back on the presentation system immediately. Apparently, the student is free to choose topics to be explored. This situation is typical for a range of systems which are not intended to be courses, but which contain (large quantities of) information the user might be interested in, e.g. reference books, encyclopedia, product catalogs.

The category of systems mentioned in the last point, the so called *content-based* applications, might become very important in the near future. There is a tendency to large on-line databases and individualized, just-in-time learning. Good navigation systems will be of paramount importance.

4.2 Interaction between presentation system and authoring system

There is an obvious overlap between a presentation system and an authoring system. The structuring and navigation techniques provided by the presentation system are equally useful for the author, namely for keeping the information surveyable and manageable. It is also important that the author can simulate the learning environment, while working on the course structure. It is hard to distinguish between the potential information requirements of user and author respectively. In the current prototype, user navigation is restricted to some predefined views, all of which aim at the selection of a particular unit. There is no way of searching directly for a presentation or an exercise, let alone of specifying complex conditions like "with actors X and Y but not Z, treating (a special case of) concept C and abstraction level L ...". Still, for both student and author these kind of questions might be useful, if that is just the information they remember of the object(s) they want to see back.
Another observation is, that the presentation system is currently model dependent. Any change in the model requires a change in the presentation system. This is caused by the fact that the model is an ad-hoc structure of related entities. If the model would be described in terms of a more generic datamodel, a presentation system could be generated from it, providing generic navigation mechanisms. This is not a new insight. The generic datamodel might be the relational model ([Codd,1970]) or the object oriented model ([Khoshafian,1993]).

In this view the authoring system would be nothing else than a generic query and update system, just as the presentation system (the latter with less update privileges). Unfortunately generic query and update systems are difficult to use. The relational model is notorious in this respect, because of the many joins which are needed for even modest queries. Furthermore this model is unable to describe semantic relationships, e.g. specialization, in an elegant way (see for example [Kent,1979]). Semantic or object oriented models are better candidates. Query languages are much more intuitive and semantic relations are easily expressed ([Bekke,1992]). However, these models still suffer from a number of problems, which hamper unambiguous and intuitive modeling of information [Kim,1991]. Yet, this is a major prerequisite for intuitive information retrieval. An attempt to solve some of these problems can be found in [Uittenbogaard,1995].
Authoring and presentation system are actually slightly different views on the same database in this respect. It is essential, that for the student navigation is as intuitive as is structuring for the author. Therefore both systems should be based on the same generic datamodel and offer the same means for navigation. The views offered by the current prototype could be provided as predefined queries. The design of such a datamodel, with an optimal balance between expressive power and support for intuitive authoring and navigation should be a major goal of the M&E project.
For very large on-line applications the distinction between author and student will disappear, many people contributing to the same database and learning from each other. This is another argument for a tight integration of authoring and presentation system.

5 Acknowledgements

In the M&E project, Herman ter Horst originally had the idea of developing a presentation system (as an alternative to an adaptive teaching system), assisting a user with information on course structure and on course parts that have already been done and that can be done directly. He also made the initial design. The contribution of Josee Zielman, Jan Dirk Wolters and Johan Ero to the realization of the presentation system cannot be overestimated.

References

[Albertsmeier,1995] B. Albertsmeier (1995), Square Dance - course material and program description, IPO handleiding no. 137.

[Augusteijn,1990] Augusteijn L. (1990), The Elegant compiler generator system, in *Attribute Grammars and their Applications* edited by P. Deransart and M. Jourdan, *Springer Lecture Notes in Computer Science* 461, Springer-Verlag, Berlin.

[Bekke,1992] J.H. ter Bekke (1992), Semantic Data Modeling, Prentice Hall.

[Codd,1970] Codd E.F. (1970) A relational model for large shared data banks. *Communications of the ACM, 13*, 377-387.

[ter Horst,van Loon,1995] H. ter Horst, P. van Loon (1995), Preliminary information model for multimedia course development.

[Kent,1979] Kent W. (1979) Limitations of record-oriented information models. *ACM transactions on Database Systems, 4*, 107-131.

[Khoshafian,1993] Khoshafian S. (1993) Object oriented databases, Wiley & Sons.

[Kim,1991] Kim H.J. (1991) Algorithmic and computational aspects of OODB, in Gupta R. and Horowitz E. (eds.): *Object-Oriented databases with applications to CASE, networks, and VLSI CAD*, 26-61.

[van Loon,1995] P.van Loon (1995), Multimedia Educational Systems (Project Contract).

[Masthoff,van Hoe,1994] J. Masthoff, R. van Hoe (1994), APPEAL: A multi-agent approach to interactive learning environments, *Proceedings of the 1994 MAAMAW workshop, Denmark.*

[Nielsen,1990] J. Nielsen (1990). Hypertext & Hypermedia. Academic Press, London, ISBN 0-12-518410-7.

[Uittenbogaard,1995] F. Uittenbogaard (1995), Improved taxonomical organization for searching in object-oriented databases, IPO Annual Progress Report no. 29.