A model for hypermedia learning environments based on electronic books

Ignacio Aedo, Paloma Díaz and Nadia Catenazzi
Laboratorio DEI, Universidad Carlos III de Madrid, Spain

Designers of hypermedia learning environments could take advantage of a theoretical scheme which takes into account various kinds of learning activities and solves some of the problems associated with them. In this paper, we present a model which inherits a number of characteristics from hypermedia and electronic books. It can provide designers with the tools for creating hypermedia learning systems, by allowing the elements and functions involved in the definition of a specific application to be formally represented. A practical example, CESAR, a hypermedia learning environment for hearing-impaired children, is presented, and some conclusions derived from the use of the model are also shown.

1. Introduction

Current hypermedia learning environments do not have a common development basis. Their designers have often used ad-hoc solutions to solve the learning problems they have encountered. However, hypermedia technology can take advantage of employing a theoretical scheme – a model – which takes into account various kinds of learning activities, and solves some of the problems associated with its use in the learning process. The model can provide designers with the tools for creating a hypermedia learning system, by allowing the elements and functions involved in the definition of a specific application to be formally represented.

This paper outlines some basic principles of computer-supported learning and the problems related to the use of hypermedia learning systems. It then summarizes a number of hypermedia and electronic-books models, which represent the basis for the development of the theoretical model presented later, and it describes CESAR, a hypermedia learning environment for hearing-impaired children which is based on this model. Lastly, it reports some conclusions drawn from the definition of the model and the development of CESAR.

2. Learning and hypermedia

Hypermedia systems, being non-linear nets of information, offer a representation similar to human knowledge, and can thus be considered as useful learning tools. However,

hypermedia efficiency in computer-aided environments is not proved, nor is hypermedia devoid of problems (Reader and Hammond, 1994), a common one being related to a user's ability to navigate freely within the information offered. This problem suggests that there should be a change in the design of hypermedia courseware [7], with a view to imposing some navigation constraints. On the one hand, unnecessary links must be avoided. On the other hand, as Haga and Nishino (1995) suggest, students should be prevented from excessively deepening their knowledge of a subject, considering three as the maximum number of levels (depth of hyperlinks) that should be allowed. Structuring information can be considered as being contrary to the hypermedia philosophy, but students can still benefit from structured courseware. For example, Beltran (1993) proposes a courseware structure made up of courses, examples and practice. Courses will include information about the main subjects; examples will be particular or simplified cases related to the courses; and practice will consist of problems and presentations which require a creative involvement.

Another problem is that of determining the kinds of tasks to be provided to the student and how these activities will integrate into the learning process. A computer-aided learning system might include tasks such as reading, creative writing, problems resolution and self-evaluation. Practical tasks may also be required, for instance editing information, gathering, annotating and restructuring the material, and marking useful sections (Page, 1991). In any case, it is important to design an environment which combines complementary activities, such as active-passive, creative-reactive and directed-explanatory.

Matching the instructional approach with student learning objectives is yet another problem. Each student's motivation to learn is different. Moreover, society and his/her social role are factors that directly influence what, how and when he/she learns. And other important issues bear upon student learning such as age, sex, educational level, prior training, ethnic background, cultural heritage, level of initial motivation, personality and physical abilities (Barker, 1993). The learning style of each student can suddenly change, for example due to his/her emotions, the type of courseware being used or the current stage of a system. Consequently, the environment must be personalized so that the learning method is adapted to each student's style abilities (Allinson and Hammond, 1990; Barker, 1993). Some experiments have been based on this idea; an example is AnatomTutor (Beaumont and Brusilovsky, 1995), a hypermedia learning system for medicine that is adapted to the user's needs using artificial-intelligence techniques.

A further issue in developing hypermedia systems is the design of the user interface. In fact, a good user interface can help to solve many of the problems stated above. In this respect, the use of metaphors and stories may be appropriate. Metaphors provide a way of encapsulating system facilities, minimizing the cognitive overhead and maximizing system transparency and use (Vaananen, 1995). For example, the book metaphor allows a familiar learning space to be created; this kind of metaphor has already been implemented in the *ABC Book for Early Learners* (Barker and Giller, 1990), a hypermedia system for learning the alphabet. Equally, the intrinsic characteristics of stories make them a powerful mechanism for solving some of the hypermedia problems, in particular disorientation and system control. Stories also serve as a natural context where students can acquire and relate knowledge. For instance, The *Jasper* series (Barron and Kantor, 1993) uses video tapes of a complex story to show

mathematical concepts, and *CyberBunch* (Chijn and Plass, 1995) puts together the story and book metaphors to help readers understand German texts.

The problems mentioned above, and the advantages of using a book metaphor and a story have been taken into account in developing the model presented in the next section.

3. Designing hypermedia learning systems

This section presents a model, based on the book metaphor and the idea of a story, which inherits a number of characteristics from hypermedia and electronic-book models.

3.1. Hypermedia models

According to the *Diccionario de la Lengua Española* (1992), a model is a theoretical scheme of a complex system or reality that facilitates its comprehension and the study of its behaviour(. In particular, a hypermedia model gives unambiguous definitions of the elements and relationships needed to represent any application which uses this technology. A number of models have been developed, such as HAM, Dexter and Labyrinth, which allow designers logically to describe hypermedia applications.

The HAM hypermedia model (Campbell and Goodman, 1988; Delisle and Schwartz, 1996) points up some key features such as version control, the use of filters to retrieve information, and data control access.

The Dexter model (Halasz and Schwartz, 1990; 1994) introduces the concept of anchor and separation between nodes and their information. This model has been improved in order to take account of collaborative learning environments (Min and Rada, 1993) and the use of multimedia information (Hardman *et al*, 1993; 1994).

Labyrinth (Díaz, 1995) is a model for the design of collaborative hypermedia systems that defines seven elements (nodes, links, contents, anchors, users, events and attributes) and a set of operations to express both static structure and dynamic behaviour. The model separates the structure (i.e. information holders or nodes) from the contents (i.e. information pieces), providing information-sharing by reference instead of by copy. This feature also makes possible the assignation of attributes and dynamic behaviour to particular content-items using events, which are independent elements. It generalizes the link definition, allowing distinct kinds of links to be modelled (e.g. bi-directional links, calculated or virtual links, and conditional links). In addition, Labyrinth includes mechanisms for controlling access to a hyperdocument by users, be they individuals or groups: constraints can be put on the ability to edit and personalize it, and a control version can be defined, either over the whole hyperdocument or over particular elements of it.

However, all such hypermedia models are very general and do not take into account the specific characteristics (elements as well as functionalities) of learning environments.

3.2. Electronic-book models

Since the book metaphor appears to be most suitable for presenting electronic books (Benest and Duric, 1990; Barker, 1992; Catenazzi, and Sommaruga, 1994), some features of electronic-book models must be considered.

Electronic books present many features which make them close to hypertext systems. However, hypertext models are based on the classical definition of hypertext structures (nodes and links) rather than on the concept of pages and page components which characterize electronic books. For this reason, it is important to consider models which have been explicitly defined for electronic books. Barker (1992) presents a set of three high-level models, including a conceptual model, a design model, and a fabrication model. The conceptual model, intended for the end-user, is composed of a series of pages of reactive and dynamic information, which support two primary functions: book control and information display. The design and fabrication models are intended for designers and producers of electronic books. The design model includes the formulation of the end-user interfaces, the book and page structure, the content of the book, and the nature of the reader services. The fabrication model describes the relationships between the various stages of system development, from specification of the content and structure of the book, to the final book distribution.

These models are high-level models which describe, from a conceptual point of view, several aspects of electronic-book production, design and use. A more complete and formal model for electronic books is the hyperbook model (Catenazzi and Sommaruga, 1994), in which an electronic book is seen as an interactive and dynamic system, i.e. a system which can evolve from one state to another. The hyperbook model is defined in terms of structural and functional components. The structural components reflect the book subdivision into pages, and the page subdivision into elements such as text or figures. The functional aspect is indispensable for describing the use of a dynamic and interactive system. In particular, a number of operators, which represent reader services (e.g. orientation, navigation, personalization, history and searching), allow a user to change the system state. This simple model is intuitive, general and easy to extend.

3.3 A model for hypermedia learning systems

The model presented below can be considered as a basis for the development of hypermedia learning systems. It offers a common framework where designers will be able to develop their systems, focusing on the educational material and on the design of help facilities, and ignoring inherent hypermedia problems.

The EBNF (Extended Backus-Naur Form) notation (Sethi, 1989) has been used for the definition of the model. Items ending with the suffix 'id' are used to identify particular attributes of elements. Some elements are not completely specified, since their definition depends on the development of a specific application, and they are therefore formalized for a concrete example in Section 4 below. Table 1 shows the different symbols and their meaning.

Symbol	*Description*	*Example*
< >	Begin and end symbols that enclose no terminal elements.	<item>
<< >>	Begin and end symbols that enclose terminal elements.	<<item_id>>
::=	Decomposition symbol, where the left part is made up of the elements in the right part.	<item>::= <item1><item2>
(	Exclusive symbol (OR).	((item_id>> | <item>
{ }	Repetition symbol; the elements enclosed are repeated from 0 to N times.	{ <item> }
( )	Associative symbol.	(<<item_id>> I <item>)
[ ]	Optionality symbol: the elements enclosed are optional.	[<item>]

Table 1: EBNF notation

The learning environment is defined as a *library,* with which several *users* interact, and a series of *peripheral elements* which support the learning process. The *contents* are independent elements, so that the same content can be used in different environment elements to promote relations among knowledge domains (Page, 1991). *Events* are relevant facts that occur in the environment. They are defined by the tutor or by the system and allow, for example, contents to be synchronized or stimuli to be presented to the students. Moreover, the environment components are inter-related. Thus, the learning environment is defined by:

```
<learning_environment>: <library> (<user> {<user>})
   {<peripheral_element>} (<content> {<content>}) {<events>}
   <relation>
```

This definition supposes that the environment is composed of a library which has at least one user and one content, and where peripheral elements and events can be included. Finally, a set of relations among these elements is defined.

Elements definition

The central axis in the environment are the *books* that are in the library, composed of a *story* and a set of *trainings.* The use of the *book* allows a natural environment to be created where the student knows where he/she is, where he/she can go, and what he/she can do, since he/she is familiar with physical books. The *library* definition is:

```
<library> ::= <book> {<book>}

<book> ::= ((book_id(( <story> (<training> {<training>})
```

The *story* must be sequential, beginning with *the front cover,* followed by several content *pages,* and ending with the *back cover.* According to Haynes's (1990) recommendations, intellectual-property information must be included in order to preserve the book *copyright.* Each content *page* can have tied a set of *exercises* for each *training,* which are oriented towards acquiring a particular knowledge or skill. In addition, *bookmarks* can be added in the content and the *copyright pages.* The *story* definition is:

```
<story> ::= <front_cover> (<page> {<page>}) <copyright>
   <back_cover>

<front_cover> ::= <<page_id>>

<page> ::= <<page_id>> {<<training_id>> (<<exercise_id>>
   {<<exercise_id>>})} <<bookmark>>

<<bookmark>> ::= "true" ( "false"

<copyright> ::= <<page_id>> <<bookmark>>

<back_cover> ::= <<page_id>>
```

Trainings are used as a learning aid and cover the topics considered relevant by the tutor (e.g. mathematics, history). Each one contains *exercises* for a particular *category*. For example, in a *training* of mathematics, there can be exercises of the Real Numbers *category*. These *exercises* will fundamentally formulate questions that students will answer using different *strategies*. *Strategies* give the student the possibility of solving the same *exercise* in different ways, since the *presentation* (i.e. how contents are shown), the *interaction* (i.e. how the system will respond to the student interaction) and the *assessment* (i.e. how to determine if the answer is correct) can be modified from one *strategy* to another by the tutor or an intelligent system. Each *strategy* has a *simulation* that acts as a stimulus and indicates how the *exercise* must be solved. The training composition is:

```
<training> ::= <<training_id>> (<exercise> {<exercise>})

<exercise> ::= <<exercise_id>> (<<category_id>>
   {<<category_id>>}) (<strategy> {<strategy>})

<strategy> ::= <<strategy_id>> <<simulation_id>> <resolution>

<resolution> ::= <presentation> <interaction> <assessment>

<presentation> ::= <<event_id>> {<<event_id>>}

<interaction> ::= <<event_id>> {<<event_id>>}

<assessment> ::= <<event_id>> {<<event_id>>}
```

Figure 1 resumes the library structure in the model.

User is included in the model as a component that must be instanced according to the objectives and characteristics of the concrete system. This element allows the interaction with the system for each person to be established.

```
<user> ::= <<user_id>> <user_information>

<user_information> ::= Depends on the concrete system.
```

Peripheral elements allow a number of additional activities to be included and can be used in any *book* in the *library*. They are classified into *external elements* and *tools*, according to the activities they support using the *taxo_id* attribute. The first category includes those that mainly support explanatory activities. *External elements* can be, for example, reference books, dictionaries, and help manuals. The information transfer from/to the book is made by using those *tools* which support active and creative activities and allow the user to develop a personal framework. Some examples of *tools* are notebooks, and calculators. The different *peripheral elements* are classified using the attribute *class_id*.

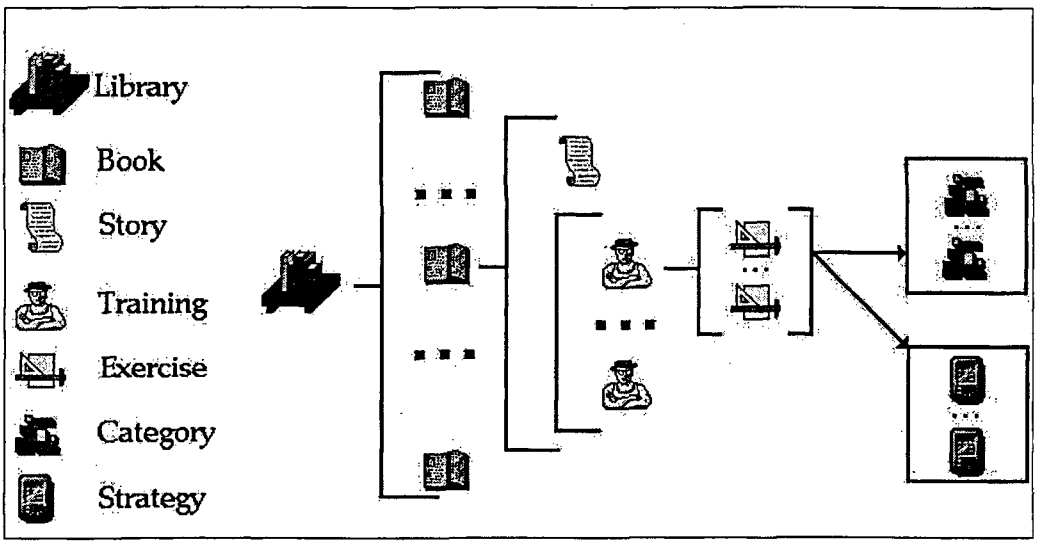

Figure 1: Library structure

```
<peripheral_element> ::= <<peripheral_element_id>>
   <<taxo_id>> <<class_id>> <peripheral_element_info>

<<taxo_id>> ::= "external_element" | "tool"

<peripheral_element_info> ::= Depends on the concrete
   peripheral element.
```

The *contents* refer to particular information included in the environment. Each one is defined by a type and a length specified in the spatial as well as in the temporal axis.

```
<content> ::= <<content_id>> <<type_id>> [<<data>>] <length>

<<data>> ::= string of bits

<length> ::= [<<duration>>] [<<rect>>]

<<duration>> ::= Length in the temporal axis.

<<Rect>> ::= Length in the spatial axis.
```

Events are included in the model to represent reactive and dynamic behaviour of the multimedia information. In addition, *events* can be used to model any other kind of *conditional actions,* for example to activate a process depending on the system status. An *event* is defined as:

```
<event> ::= <<event_id>> <<condition>> {<<action>>}

<<condition>> ::= Logical expression.

<<action>> ::= Script.
```

The elements presented above are inter-related in the following way:

```
<relation> ::= <rUser> <rContent> <rEvent> <rLink>
```

There are some decisions that must be taken for each student: the *contents* to be presented (rUC relation); the *peripheral elements* that can be used (rUP relation); the *exercises* that can be done (rUT relation); and the *events* that can be enabled (rUE relation). These relationships are specified as follows:

```
<rUser> ::= {<rUC>} {<rUP>} {<rUT>} {<rUE>}

<rUC> ::= <<user_id>> <<content_id>> <<licence>>

<rUP> ::= <<user_id>> <<peripheral_element_id>> <<licence>>

<rUT> ::= <<user_id>> <<book_id>> [<<page_id>>]
   <<training_id>> [<<exercise id>>] <<licence>>

<rUE> ::= <<user_id>> <<event_id>> <<licence>>

<<licence>> ::= "true | "false"
```

Each *content* should be located at any place in the *story pages* (rSC relation). Nevertheless, the location of the *contents* in the *exercises* will depend on the *strategy* used (rTC relation).

```
<rContent> ::= {<rSC>} {<rTC>}

<rSC> ::= <<book_id>> <<page_id>> <<content_id>> <position>

<rTC> ::= <<book_id>> <<training_id>> <<exercise_id>>
   <<strategy_id>> <<content_id>> <position>

<position> ::= [<<space>>] [<<time>>]

<<space>> ::= Position in the spatial axis.

<<Time>> ::= Position in the temporal axis.
```

Each event can be tied to different model elements, in particular to *pages* (rSE relation), *exercises* (rTE relation) and *contents* (rCE relation).

```
<rEvent> ::= {<rSE>} {<rTE>} {<rCE>}

<rSE> ::= <<book_id>> [<<page_id>>] <<event_id>>

<rTE> ::= <<book_id>> <<training_id>> [<<exercise_id>>]
   <<event_id>>

<rCE> ::= <<content_id>> <<event_id>>
```

A basic characteristic of a hypermedia system is the existence of links. In the model, links are established between two *pages* of a *book* (relation rSS), two *contents* (rCC relation), a *content* and a *page* of a *book* (rOS relation); and a *content* and a *peripheral element* (rEP relation). All these relationships are included in rLinks

```
<rLinks> :: = {<rSS>} {<rCC>} {<rCS>} {<rCP>}

<rSS> ::= <<book_id>> <<page_id>> <<page_id>>

<rCC> ::= <<content_id>> <<content_id>>

<rCS>::= <<content_id>> <<book_id>> <<page_id>>

<rCP>::= <<content_id>> <<peripheral_element_id>>
```

The model also offers a set of basic functions. Table 2 lists navigation functions. Tables 3, 4, 5, 6 and 7 include functions that can be applied to the model elements.

GoLibrary	Go to the library from everywhere.
GoBook	Go to a particular book from the library.
GoFrontCover	Go to the front cover of the current book.
GoBackCover	Go to the back cover of the current book.
GoPage	Go to a particular page of the current book.
GoNextPage	Go to the next page of the current book.
GoPrevPage	Go to the previous page of the current book.
GoTrain	Go to the training list of the current book.
GoCatTrain	Go to the category list of a particular training.
GoPageExer	Go to the exercise list of a particular page.
GoExerPage	Return from the exercise to the last page activated.
GoToLink	Activate a link.
GoBookMark	Activate a bookmark.

Table 2: Navigation operations

CreateLink	Create a link.
DeleteLink	Delete a link.
CreateBookMark	Create a bookmark.
DeleteBookMark	Delete a bookmark.

Table 3: Operations on the links

ActivateSimulation	Activate the simulation for an exercise and a strategy.
ActivatePresentation	Activate the presentation for an exercise and a strategy
ManageInteraction	Activate the interaction for an exercise and a strategy
CheckAnswer	Activate the assessment for an exercise and a strategy.

Table 4: Operations on the exercises

ShowContent	Show a content.
HideContent	Hide a content.

Table 5: Operations on the contents

RunEvent	Run the actions of an event.
StopEvent	Stop the action execution of an event.
PauseEvent	Make a pause in the action execution of an event.

Table 6: Operations on events

ActivatePeripheral	Activate a peripheral element.
ClosePeripheral	Close a peripheral element

Table 7: Operations on peripheral elements

4. Using the model: the design of CESAR

CESAR (Aedo *et al*, 1995) was designed starting from the model presented in the previous section. It is a hypermedia learning environment which aims to help hearing-impaired children to acquire the necessary skills in sign and written languages. This system will initiate the deaf child to the story structure and will provide him or her with the necessary experience by using stories. In the system description, we include those parts of the model which have been left open because they are dependent on the specific system and on its functionalities.

Each book in the library consists of two parts: the story and the training. A combination of different media (text, image, and video) is used to represent the book contents in both parts. Therefore, the content definition is completed by instancing the type of information that can be included:

```
<<type_id>> ::= "image" | "text" | "video"
```

The story is based on the book metaphor and consists of a sequence of pages which can be classified according to three different types. The first one is intended to cover the information contained in the presentation pages of the story: front and back cover. A front cover page is seen in Figure 2, where the video shows a narrator signing the story. The second one corresponds to those pages which contain information about the book production and copyright, for example the name of the publisher. The last one contains the story pages. In each page the story is displayed in text, graphic and video forms.

CESAR has currently one training part, addressed towards learning sign and written language. Starting from the model presented in the previous section, from the

Figure 2: Front-cover page

conversations with teachers who are involved in hearing-impaired children's education, and from the teaching methodology defined in Aedo *et al* (1994), the global structure of the training was designed. The training consists of a number of exercises which belong to the following categories:

- Vocabulary category, which allows the child to extend his/her vocabulary with the terms used in a concrete story;
- Understanding Questions category, intended to facilitate the child in the task of analysing and understanding the story contents, either through video, or through the written text;
- Regulators category, which motivates the child toward learning and interiorizing the syntactic structure of the language, by using questions which contain terms such as Who, Where, When, What, Which, Why, and How;
- Values, Motives, and Consequences category, which allows the child to realize the existence of different social values;
- Narrative Structure category, which aims to create a logical internal speech to help a deaf child relate his/her experience, and interiorize the story contents;
- Expressive Elements category, which is useful for enriching information and communication, independently from the language used to communicate;

- Grammatical Elements category, in which, by using the formal structure of the book, different kinds of exercise are defined to help a child incorporate these morpho-syntactic elements in his/her colloquial language.

By instancing the general model, the result is:

```
<<training_id>> ::= "linguistic competence"

<<category_id>> ::= "vocabulary" | "understanding_questions" |

"regulators" | "values_motives_conseq" |

"narrative_structure" | "expressive_elements" |

"grammatical_elements"
```

More than 50 different strategies have been designed according to the particular contents to be presented to the child, and to his/her specific needs. In Figure 3, an exercise in the Narrative Structure category is shown. The exercise consists of three different images that the child has to organize, following the same sequence as they appear in the story.

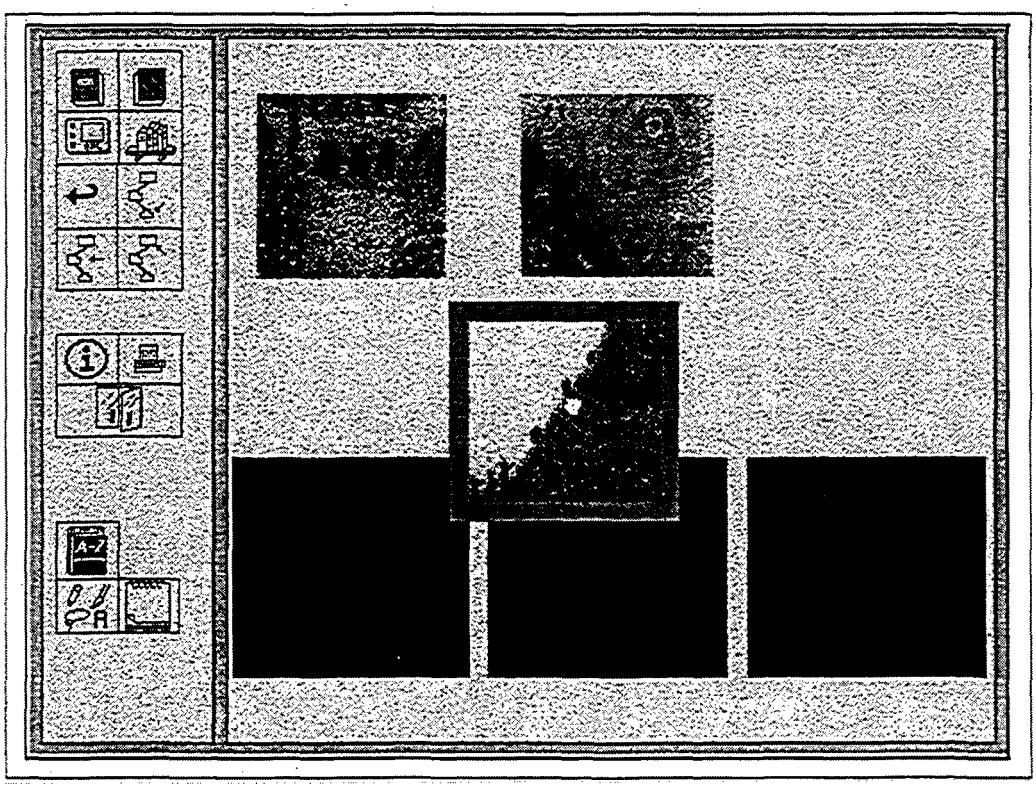

Figure 3: An exercise in the Narrative Structure category

Learning style takes into account three different levels of the learning process, considering the specific problems of a child in any of these levels, and presenting information depending on the specific level. The three levels are as follows.

1. The initial level, where the child is still acquiring the phonemes of a language, and where he/she is not able to read and does not know sign language. Children who belong to this level are between four and seven years old. Sign language, supported by images, is given priority.

2. The intermediate level, where the process of acquiring the phonemes of the language has been completed, and the child is in a phase where he/she is extending his/her vocabulary and creating simple linguistic structures. The child can read, although he/she has difficulties in understanding, and is starting to know sign language. Children who belong to this level are between seven and nine years old. Written text, supported by sign language, is given priority.

3. The highest level, where the child has acquired more linguistic and reading skills but needs to improve and strengthen his/her linguistic structures. Children who belong to this level are between 9 and 12 years old. The sign-language text, supported by explanatory activities, is given priority.

The child's characteristics have been formalized, maintaining information about his/her name and learning level, and about the operations he/she is allowed to accomplish with respect to the narrator and the bookmarks. The result of this formalization is:

```
<user_information> ::= <personal_info> <<level>> <command>

<personal_info>::= <<first_name>> <<last_name>>

<<first_name>> ::= String of characters.

<<last_name>> ::= String of characters.

<<level>> ::= "initial" ( "intermediate" ( "high"

<command> ::= <narration> <bookmark>

<narration> ::= <repeat_narration> <stop_narration>

<repeat_narration>: : = <<licence>>

<stop_narration> ::= <<licence>>

<bookmark> ::= <<licence>>
```

CESAR takes into account the peripheral elements considered in the model. There are two kinds of tool to accomplish active and creative activities, which extend the book metaphor to include the desktop metaphor: the personal notebook and the drawing-tools box.

The child can write, draw and create animations in the personal notebook, which is unique to each child. He/she can create his/her own information starting from the book contents, thus favouring the learning process and inciting him/her to use his/her imagination.

```
<<taxo_id>> ::= "tool"

<<class_id>> ::= "notebook"

<peripheral_element_info>::= <<note_id>> {<<note_id>>}
```

In Table 8, the operations which are available in the notebook are listed.

GoNextNote	Go to the next page of the notebook.
GoPrevNote	Go to the previous page of the notebook.

Table 8: The notebook operations

The drawing-tools box offers a number of mechanisms for creating and changing information in both the personal notebook and in the book. These mechanisms can be defined as actions which are performed if particular conditions are fulfilled, i.e. they can be represented using the concept of events previously formalized.

```
<<taxo_id>> ::= "tool"

<<class_id>> ::= "drawing_box"

<peripheral_element_info> ::= <browser> | <select> | <eraser>
   | <pencil> | <write> | <line> | <copy> | <paste>

<browser> ::= <<event_id>> {<<event id>>}

<select> ::= <<event_id>> {<<event id>>}

<eraser> ::= <<event_id>> {<<event id>>}

<pencil> ::= <<event_id>> {<<event id>>}

<write>::= <<event_id>> {<<event id>>}

<arrow> ::= <<event_id>> {<<event id>>}

<copy> ::= <<event_id>> {<<event id>>}

<paste> ::= <<event_id>> {<<event id>>}
```

In Table 9, the operations which are available in the drawing-tools box are listed.

BrowserMode	Used to go back to the normal mode, where the child can use the functionalities of the environment.
SelectObject	Allows the child to select an object shown on the screen, e.g. a portion of text.
RemoveObject	Allows the child to erase the selected object.
DrawPixel	Allows the child to draw using the pencil.
DrawText	Allows the child to write using the keyboard.
DrawLine	Allows the child to draw a line.
CopyObject	Allows the child to copy the selected object.
PasteObject	Allows the child to paste a copied object in any location of the screen.

Table 9: The operations provided by the drawing-tools box

The dictionary is an external element which supports explanatory activities and helps the child to advance in the language-learning process. The dictionary consists of words with different meanings, each of which includes a description and a sample situation. This information is available both in written and sign language. In addition, each word-entry contains its labial form. The dictionary can be also used as an independent object. Its use is recommended to the children in the intermediate and highest levels. The operations available in the dictionary are listed in Table 10.

```
<<taxo_id>> ::= "external_element"

<<class_id>> ::= "dictionary"

<peripheral_element_info> ::= <entry> {<entry>}

<entry> ::= <<entry_id>> <<level>> <interpretation>
   <labial_form>

<interpretation> ::= <interpretation_id> (<article>
   {<article>})
(<meaning> {<meaning>}) {<related entry>}

<article> ::= <<content_id>>

<meaning> ::= <definition> <example>

<definition> ::= <<content_id>>

<example> ::= <<content_id>>

<related_entry> ::= <<entry_id>>

<labial_form> ::= <<content_id>>
```

NextTerm	Go to the next term following the alphabetic sequence of the dictionary.
PreyTerm	Go to the previous term following the alphabetic sequence of the dictionary.
SearchTerm	Look for a particular term in the dictionary.
ShowLabialForm	Show the labial form of the selected term.
CloseLabialForm	Close the labial form if it has been activated.
ShowSignedForm	Show the sign form associated to a definition or a meaning.
CloseSignedForm	Close the sign form associated to a definition or a meaning.

Table 10: The operations available in the dictionary

CESAR was evaluated using an expert technique called 'jogthrough' (Rowley and Rhoades, 1992) which can achieve very good results. In particular, the evaluation involved the system functionalities (the ease of use, the navigation tools, etc.), and the learning tools provided by the system (the design of the exercises, the contextualization of the learning process, etc.).

5. Conclusions

The hypermedia learning-environment model presented in this paper can be used as a basis for designing hypermedia learning systems for a number of reasons.

Firstly, this model is adaptable. It is possible, for instance, to change the page contents, and thus, substantially, the story, maintaining the same logic scheme. It could also be possible to achieve different learning objectives with the same story by changing the training.

Secondly, the use of the book concept in the model allows the main hypermedia problems mentioned in Section 2 above to be taken into account and solved in the system-design phase. Our experience the development and evaluation of CESAR confirmed that the use of the story allows its sequentiality to be employed as a natural way of navigating. Moreover, a student takes advantage of working with a known object, the book, in which the services are not limited by the system designer but by the environment itself.

Thirdly, the training allows the child to acquire, assimilate and associate knowledge and ideas, by doing a number of exercises designed to achieve a specific objective. In CESAR evaluation (Aedo *et al*, 1996), participants confirmed that training is an indispensable process which supports learning and allows the required competence to be achieved. They argue that this is useful not only for learning the written and learning language, but also for acquiring knowledge of a subject.

Fourthly, the inclusion in the model of peripheral elements allows the system designers to create contextualized help which complements the learning process. CESAR evaluation confirmed that it is profitable to use a dictionary which allows a hearing-impaired child to acquire new words and concepts, and to employ tools for helping the child develop new knowledge and ideas.

Finally, the model highlights the need for adapting the environment to the learning style of a child. In a specific system, this adaptation can be accomplished by using techniques such as expert systems or frame-oriented systems. These techniques will allow the instructional objective of the system to get close to the learning objective of the child, adapting the contents to the user.

Acknowledgements

Thanks to Nuria Torra, Pilar Miranda, María Martín, and Rafael Calzada for their help in realizing CESAR. Some of the content has been used with the authorization of Ediciones S.M. Some shareware was obtained via the Internet: thanks to Immedium Design.

References

Aedo, I., Martín, M., Miranda, P., Panetsos, F. and Torra, N. (1994), 'A teaching methodology for hearing impaired using hypermedia and computer animation', *Journal of Computing in Childhood Education,* 5 (3/4), 353–69.

Aedo, I., Catenazzi, N. and Calzada, R. (1995), 'Electronic stories for deaf children in a hypermedia environment (CESAR). in Maurer, H. (ed), *Proceedings of ED-MEDIA 95, Graz (Austria)*, n.p., 63–8.

Aedo, I., Catenazzi, N. and Díaz, P. (1996), 'The evaluation of a hypermedia learning environment: the CESAR experience', *Journal of Educational Multimedia and Hypermedia,* 5 (1), 49–72.

Allinson, L. and Hammond, N. (1990), 'Learning support environments: rationale and evaluation', *Computers and Education,* 15 (1–3), 137–43.

Barker, P. (1992), 'Electronic books and libraries of the future', *The Electronic Library,* 10 (3), 139–49.

Barker, P. (1993), *Exploring* Hypermedia, London: Kogan Page.

Barker, P. and Giller, S. (1990), 'An electronic book for early learners: a CD-ROM design exercise. *The CTISS File,* 10, 13–18.

Barron, B. and Kantor, R.J. (1993), 'Tools to enhance math education: the Jasper Series', *Communications of the ACM,* 36 (5), 52–4.

Beaumont, I. and Brusilovsky, P. (1995), 'Adaptative educational hypermedia from ideas to real systems' in Maurer, H. (ed), *Proceedings of ED-MEDIA 95, Graz (Austria)*, n.p., 99-102.

Beltran, T. (1993), 'Educational hypermedia: from theory to practice' in Maurer, H. (ed), *Proceedings of ED-MEDIA 93, Orlando, Florida*, n.p. 68–75.

Benest, I. D. and Duric, D. (1990), 'Some design issues in the automated office metaphor' in *Proceedings of the European X Window System User Group, Guildford, UK*, Cambridge:: European X User Group, 56–69.

Campbell. B. and Goodman, J.M. (1988), HAM: a general purpose hypertext abstract machine', *Communications of the ACM,* 31 (7), 856–61.

Catenazzi, N. and Sommaruga, L. (1994), 'Hyperbook: a formal model for electronic books', *Journal of Documentation,* 50 (4), 46–62.

Chijn, D.M. and Plass, J. (1995), 'Project CyberBunch: a hypermedia approach to computer-assisted language learning', *Journal of Educational Multimedia and Hypermedia,* 4 (1), 95–116.

Delisle, N.M. and Schwartz, M.D. (1996), 'Neptune: a hypertext system for CAD applications' in: *ACM, Proceedings of the International Conference on Management of Data*, New York: ACM, 132–43.

Díaz, P. (1995), *Modelo abstracto para el desarrollo de sistemas hipermediales seguros,* Ph.D. thesis, Universidad Politécnica de Madrid.

Diccionario de la Lengua Española (1992), Madrid: Real Academia de la Lengua Española.

Haga, H. and Nishino, M. (1995), 'Guidelines for designing hypermedia teaching materials' in Maurer, H. (ed), *Proceedings of ED-MEDIA 95, Graz (Austria)*, n.p. 283–88.

Halasz, F.G. and Schwartz, M. (1990), 'The Dexter hypertext reference model' in *.ACM, Proceedings on World Conference on Hypertext*, New York: ACM, 95–133.

Halasz, F.G. and Schwartz, M. (1994), 'The Dexter hypertext reference model', *Communications of the ACM.* 37 (2), 30–9.

Hardman, L., Bulterman, D. and Van Rossum. G. (1993), 'The Amsterdam hypermedia model: extending hypertext to support real multimedia', *Hypermedia,* 5 (1), 47–69.

Hardman. L., Bulterman, D. and Van Rossum, G. (1994), 'The Amsterdam hypermedia model: adding time and context to the Dexter model', *Communications of the ACM,* 37 (2), 50–62.

Haynes, S.L. (1990), 'Intellectual property and licensing concerns' in Berk. E. and Devlin, J. (eds), *Hypertext/Hypermedia Handbook.,* New York: McGraw-Hill, 227–42.

Min, Z. and Rada, R. (1993), 'The Dexter-Groupware model for collaborative authoring' in Maurer, H. (ed), *Proceedings of ED-MEDIA 93, Orlando, Florida*, n.p. 383–90.

Page, C.R. (1991), 'The Nuffield interactive book system', *Educational and Training Technology International*, 28 (4), 334–40.

Reader, W. and Hammond, N. (1994), 'Computer-based tools to support learning from hypertext: concept mapping and beyond', *Computers and Education,* 22 (1–2), 99–106.

Rowley, D.E. and Rhoades, D.G. (1992), 'The cognitive jogthrough: a fast-paced user interface evaluation procedure' in *ACM, Proceedings of Computer Human Interaction*, n.p. 389–95.

Sethi, R. (1989), *Programming Languages. Concepts amid Constructs,* Reading MA: Addison-Wesley.

Vaananen, K. (1995). 'Metaphor-based user-interfaces for hyperspaces in: Schuler, W., Hannemann, J. and Streiz, N. (eds), *Designing User-Interfaces for Hypermedia*, Berlin: Springer Verlag,, 68–78.

Viau, R. and Larivée, J. (1992), 'Learning tools with hypertext: an experiment', *Computers and Education,* 20 (1), 11–16.