

HILDE : A Generic Platform for Building Hypermedia Training Applications¹

A. Tsalgatidou, D. Plevria, M. Anastasiou, M. Hatzopoulos
Dept. of Informatics, University of Athens, TYPA Buildings
Panepistimiopolis, Ilisia GR-157 71 Athens
e-mail: {afrodite, grad0102, grad0073, mike} @ di.uoa.gr

Abstract HILDE is a generic platform that aims to support the development of a wide range of educational applications that use multimedia assets such as video, image and sound. It supports reusability and provides flexibility by integrating a number of tools to support the tasks and the activities of four kinds of roles, namely: the MetaAuthoring role, the Authoring role, the trainee role and the tutor role. The tasks of the MetaAuthoring role are concerned with the organization and storage of the knowledge of a scientific domain. The Authoring role tasks concern the creation of a specific application. The trainee role encapsulates the activities of an individual trainee who attends a session of lessons either locally or remotely while his/her progress is automatically monitored and guided with intelligent educational techniques. A supervisor optionally monitors the trainees' progress and advises them, if needed. HILDE can meet the requirements of many and different organizations, since the corresponding Repository schema is general and open. The provided facilities enable the access of the stored information by other software environments while ample graphical utilities facilitate the users. Special care has been taken for speed and efficiency.

1. Introduction

Research in multimedia training applications has mainly focused on developing applications whose characteristics meet the particular needs of the domain being studied, emphasizing on the specialised features that characterise this domain (e.g music [7], history [1] etc). The main goal of such applications is to attractively represent such features and to enrich the environment so as the trainee gets a complete idea of the subject being taught. These applications are also strictly restricted as far as the taught subject is concerned, while there is the need for a more flexible environment, that supports reusability and that could be used for the development of various applications covering a wide range of domains.

In this paper, we present an environment, called HILDE¹, for building hypermedia training applications. One of the main features of this environment is its genericity, in the sense that it supports the development of a wide range of educational hypermedia applications. HILDE provides a number of flexible and easy to use tools, which can be used by a domain expert to transform his/her domain knowledge into a number of interactive sessions of lessons to be attended by trainees of various levels.

The rest of the paper is organised as follows: Section 2 presents the HILDE architecture and discusses the user roles supported and the various parts/tools of HILDE. Section 3 describes the main tools that have been developed and integrated in the HILDE environment outlining their functionality. Section 4 presents related work, and section 5 outlines results to which we have concluded.

¹ Proceedings of EMMSEC '97, Florence, Italy, 3-5 Nov. 1997, published in "Advances in Information Technologies: The Business Challenge", J.-Y. Roger, B . Stanford-Smith and P. T. Kidd (eds.), pp. 469-476, IOS Press, 1998

2. The HILDE Architecture

The HILDE architecture is divided into two parts: The *Authoring Environment* and the *Delivery Environment*. This environment aims to support the activities of four kinds of roles: the *MetaAuthoring* role, the *Authoring* role, the *Trainee* role, and the *Tutor/Supervisor* role.

The MetaAuthoring role concerns the organization and storage of the knowledge of a scientific domain. The Authoring role has to do with the formation of educational sessions of lessons and the definition of the way the information will be presented, thus creating the final training application. A Trainee uses this training application in order to attend a session of lessons. The attendance takes place either locally or remotely, and is monitored and guided automatically with intelligent educational techniques [5]. A Tutor may monitor the trainees' progress by inspecting the appropriately stored information, and advise them, if needed.

2.1. The Authoring Environment

The Authoring Environment (fig.1) is used by a user, who can undertake either the MetaAuthoring role or the Authoring role. This environment consists of two parts: the “*Authoring in-the-large*” part which supports the MetaAuthoring role activities, i.e. the creation and storage of generic applications, and the “*Authoring in-the-small*” part which supports the Authoring role activities, i.e. the construction of a specific training application.

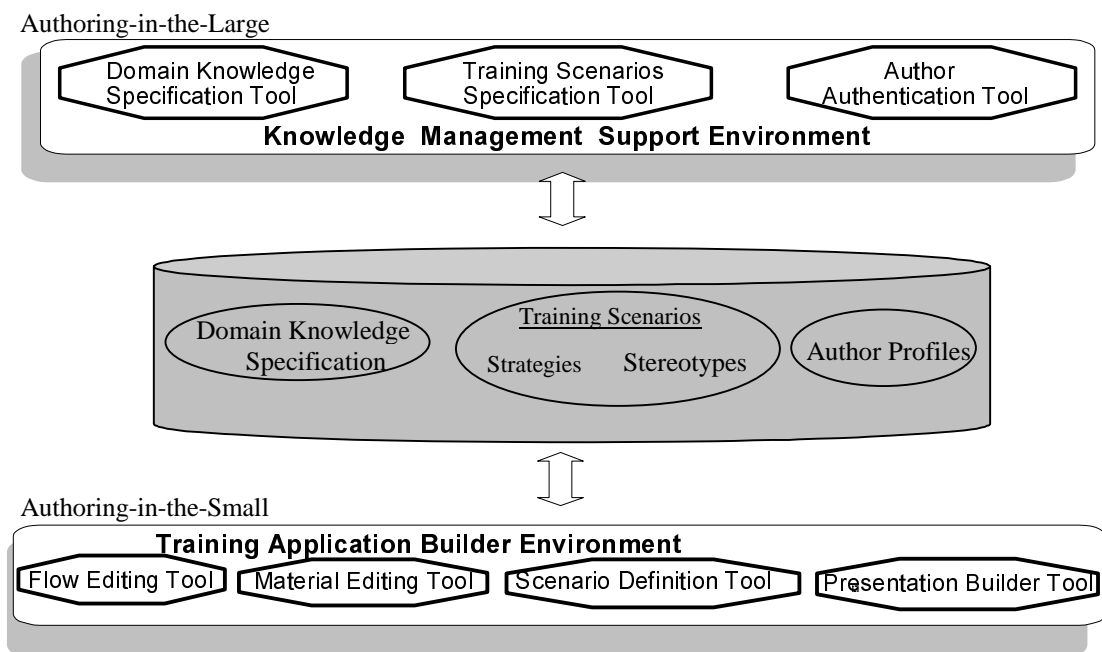


FIGURE 1. The Authoring Environment

2.1.1. Authoring in-the-large

The Authoring in-the-large part comprises the Application Repository and the Knowledge Management Support Environment (KMSE). The Application Repository is a database where generic knowledge about a scientific domain is stored. The KMSE provides a user with the needed tools to act as a meta-author².

The Application Repository consists of three parts, as follows:

- the *domain knowledge specification* part which stores the training applications' data and their links to multimedia assets
- the suggested *training scenarios* part which include trainees' stereotypes and alternative strategies that define various ways of information display. A training scenario adjusts the level of training according to a user's expertise on domain concepts and
- the *author profiles* part which include elementary information about the users of the *Knowledge Management Support Environment*.

The Knowledge Management Support Environment (KMSE) environment supports the interaction of a meta-author with the Application Repository providing the following three tools:

- the *domain knowledge specification tool* which supports the introduction, classification and modification of knowledge items of various scientific domains
- the *training scenarios specification tool* which supports the suggestion of stereotypes and strategies and
- the *author authentication tool* which supports the introduction of author names and passwords.

2.1.2. Authoring in-the-small

The Training Application Builder Environment (TABE) aims at the generation of a session of lessons which composites the target application by using the generic application specifications stored in the Application Repository. The tools that allow a user to specify the functionality and the interface of a final application are:

- the *presentation builder tool* which enables the design of the presentation of the various items on a screen,
- the *flow editing tool* which supports the selection of lessons and the definition of their sequence in a training session,
- the *material editing tool* which aids the formation of theories, questions, answers and of the multimedia assets that compose a lesson,
- the *scenario definition tool* which supports the selection of a specific scenario that will be adapted by the final application.

2.2. The Delivery Environment

The actual users of the delivery environment are trainees and tutors (see fig. 2). This environment not only assists the trainees' practice but also monitors their progress and adjusts the application interface according to the trainees' profiles and their progress.

The resulting hypermedia training system consists of:

- the *Object Base* that stores the *specification* of a training application and the *training scenarios* to be used

- the *Interface* module that supports the presentation of a session of lesson and aids the practice of a trainee
- the *Individual User Profiles* module that contains information about trainees, such as name, password, progress, individual stereotype, and point of interruption of the session
- the *Trainee Assessment* module which automatically evaluates the progress of a trainee and updates her/his stereotype in the Individual User Profiles module
- the *Customise Training Scenario* module, that customises the application’s strategy according to a selected application’s scenario as well as according to the new stereotypes being produced by the Trainee Assessment module.

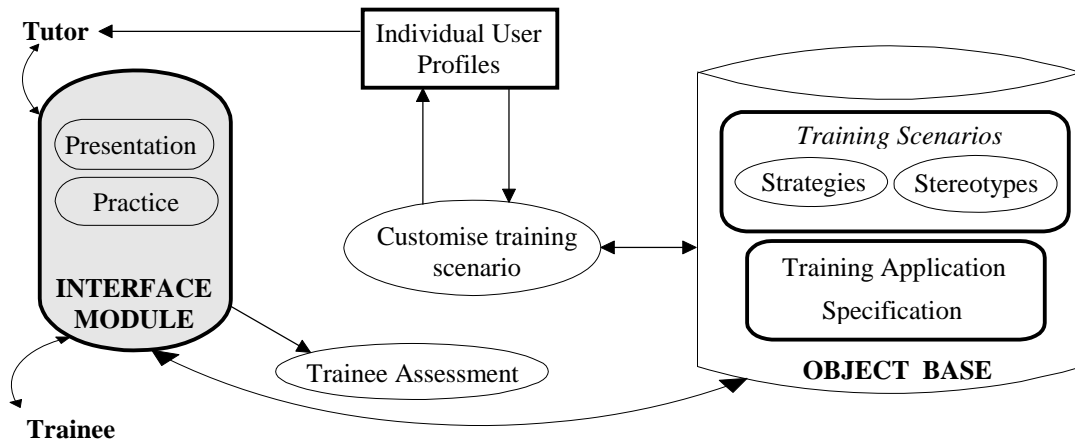


FIGURE 2. The Delivery Environment

3. HILDE Tools

The main HILDE tools are the ones offered by KMSE and TABE. We use the collective names MetaAuthoring tool and Authoring tool in order to refer to the tools offered respectively by KMSE and by TABE. The functionality of these tools is described in the following.

3.1 MetaAuthoring Tool

The MetaAuthoring tool supports the activities of a meta-author who aims mainly to suggest training *sessions of lessons* rich enough to support the educational process. More specifically, this tool aims to facilitate the organisation and storage of knowledge of general application domains, in order to form a database with ample training material and multimedia assets. The graphical environment of the tool helps a meta-author to insert, edit, update or copy various elements that form knowledge, as they are stored in appropriate fields of the database. The tool is implemented in Visual Basic 4.0, and cooperates with the Repository that is a database implemented in Oracle Server 7.0.

Thus, a meta-author, by using this tool, organizes the knowledge of general application domains into *sciences* (e.g. “Computer Science”) which are then divided into *science fields*, (e.g. “Signal Processing”) and then into *educational units* (e.g. “Medical Image Processing”).

Each educational unit is related to a number of *lessons* (e.g. “Introduction”) which consist of *theories* (e.g. “Basics on Medical Image Quality”), containing the theoretical

background of the lesson, and some exercises in the form of *questions* and *answers*. Theories and questions with their answers can be created independently to any other educational module (lesson, educational unit, etc). Each lesson is associated with multimedia assets (i.e. video, images & audio), that form the *accompanying training material* of the lesson.

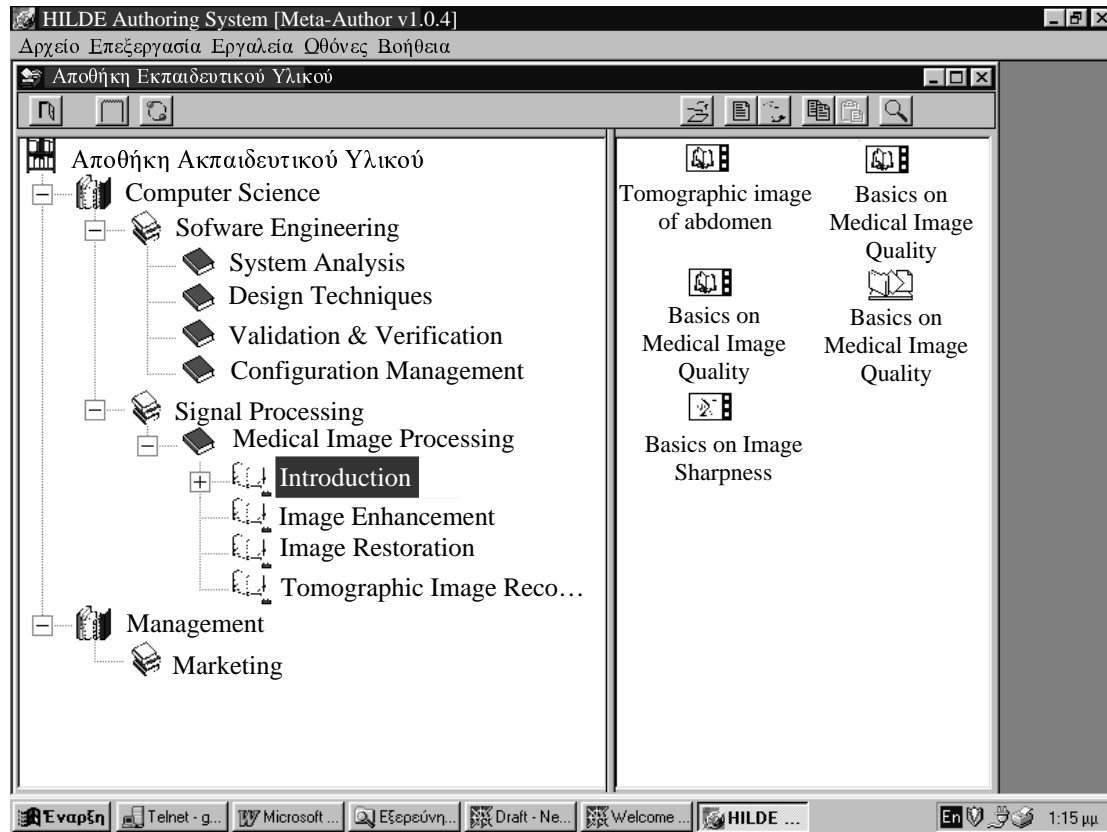


FIGURE 3. Screen illustrating a repository hierarchy with sciences, science fields, educational units and lessons and the items associated with the lesson “Introduction”.

Fig. 3 is snapshot of the creation of a lesson called “Introduction” and shows the icons corresponding to the various items associated with this lesson so far. These items are: a theory called “Basics on Medical Image Quality” (in three alternative presentation forms), a question named “Question on image sharpness” and an image called “tomographic image of abdomen”.

Theories are characterized by their *description*, *type*, *text* and *name*. The *type* of a theory denotes the level of details presented for the taught subject. Thus a theory can be *concise*, *normal* or *detailed*. The *text* of a theory is stored in a separate file with a name analogous to the unique ID of the theory. A meta-author can relate a theory to a specific lesson by just grabbing the theory icon and attaching it to the lesson icon.

Similarly to theories, questions are characterized by their *description*, *name*, *text*, *type* as well as a *level of difficulty*. The *level of difficulty* is expressed as a number 1, 2 or 3. This characteristic is needed in order to reject the difficult questions when an “easier” version of the lesson is to be presented. Each answer is associated with a *correctness* number ranging from 0 to 100. More than one answer can be correct for each question, thus allowing for example, two answers to have correctness 90 & 100 respectively. *Diagnostic messages* are presented to the trainee, depending on the chosen answer.

The multimedia assets that accompany each lesson are videos, sounds (audio), animation items or images related to the taught subject. For each multimedia asset, there is a description of its contents, which is an indexed field in the Repository and is used for rapid searching in the database of a multimedia asset.

A meta-author also suggests some *training scenarios* that can be followed during a training session. These scenarios are the basis for adjusting the response of the system to the progress of a trainee. A training scenario comprises two concepts: the *educational strategy* that determines a combination of the type of theory presentation and the number of questions to be presented to the trainee, and *stereotypes*, which are “beginner”, “good” and “advanced”. The educational strategies introduced by a meta-author are actually suggestions to the author of a final application.

3.2 The Authoring tool

The basic goal of the Authoring tool is to guide an author³ to form an educational scenario using data from the Repository in order to build an executable educational application which can be either in tbk format (toolbook application) or html format (internet application). An executable educational application consists of a number of sessions of lessons on a specific educational domain. A lesson, as it was described above, comprises theories, questions on these theories and multimedia material such as images, video and audio, organised in such a way that fits each trainee’s stereotype.

Trainees’ behavior is captured over training sessions and their stereotypes are assessed. The level of training is then adjusted automatically by determining the educational material and the way it will be presented to the trainee during the next lesson. Trainee stereotypes and educational strategies compose scenarios which have been introduced already and stored in the Repository by a meta-author. The Authoring tool, through its graphical environment, facilitates an author either to select material from the Repository or to introduce new material which will be eventually stored in the object base of the delivery environment.

One of the first tasks of an author is to decide which strategy will be incorporated in an educational application. This can be either selected from the Repository or introduced by the author. Then, the author selects lessons’ theories in a way that also determines their flow in the application. Theory presentation may take place in three alternative forms which serve diverse groups of users who may require different modes of interaction with the system according to their role and level of knowledge. After theory selection, an author designs the presentation of the final application. In fact, a number of alternative pre-constructed forms are available. The presentation can be enhanced with the use of multimedia material from the Repository in order to speed up the assimilation of concepts by trainees and make the application more interesting or just for aesthetic reasons.

Finally, an author chooses questions of various levels of difficulty and designs screens for the presentation of the selected theories. The final presentation of the selected theory and questions is determined automatically in real time by the incorporated strategy according to the type of individual trainees.

4. Related Work

Work related to HILDE can be classified in three categories: hypertext models and methodologies, hypermedia authoring environments and specialised hypermedia & multimedia training environments and applications.

HDM [2] (Hypertext Design Model) and the Dexter Model [3] are two examples of work of the first category. HDM is a general-purpose model for hypertext development that suggests the notions of “Authoring in-the-large” and “Authoring-in-the-small”. HILDE has borrowed these two terms from the HDM methodology and uses them in a similar generic notion for the authoring of generic as well as of specific multimedia applications. The Dexter Model [3] aims to provide a principled basis for comparing systems as well as for developing interchange interoperability standards and it comprises three layers: the storage layer, the run time layer and the within-component layer. The Dexter Model is a system-oriented model in the sense that it aims to capture abstractions found in hypertext systems rather than applications as it is the case with HDM and HILDE.

Examples of work of the second category are Multicard [6], which provides a hypermedia toolkit and an interactive authoring/navigation tool for the creation and manipulation of hypermedia structures, SEPIA [8] which is a hypermedia authoring system that addresses the issues of hypermedia documents sharing and multimedia functionality integration and MTDE [9], which provides a framework for assigning hypermedia functionality to multimedia assets, and suggests the construction of an asset Repository and integrated tools. The main difference between HILDE and these systems is that the latter do not specifically address the production of hypermedia training applications. Another similar platform is HTAS [10] which can be considered as the predecessor of HILDE, since the HTAS proposed architecture for the authoring and the delivery environment are analogous to the ones proposed by HILDE. In addition, HILDE implements particular educational strategies and provides specific ways and flexible possibilities to the authors to form lessons and assign to them specific characteristics which are useful in a training process.

Examples of specialised hypermedia training environments and applications can be found in MuG [1], in the PsyCLE [4] project and in the PianoForte system [7]. Environments of this kind offer to a trainee a ready-to-use teaching environment for a specific knowledge domain, however, they neither enable the domain-expert to intervene and compose lessons nor they promote reusability.

HILDE incorporates characteristics from the above three categories of related work and goes further as its main goal is to provide a generic and flexible environment for developing any kind of training application supporting the notion of reusability.

5. Conclusions

The HILDE project ends to an integrated environment based on a generic model, which supports reusability and offers flexibility. The genericity of the model, the flexibility of the offered tools and of the environment result in specific advantages such as: enhanced productivity of authors, improved courseware production, fast delivery of new courses and minimised production cost, individualisation of courses by taking into account the trainee stereotypes, effective and efficient access to the training material, easy navigation through the lessons and training material due to the low complexity, and distance learning through the World-Wide Web.

We believe that the HILDE approach contributes to the effective, quick and cost effective production of hypermedia training applications. This is especially important for organisations which today face the challenge of reducing training costs while on the other hand continuing training and education seem to be necessary for the creation of a competitive advantage of the organisation.

Future work will mainly concentrate on two issues. We will investigate degrees of freedom which can be given to a trainee as regards navigation in the hypernetwork, taking into account that full navigation freedom may create disorientation problems as well as problems in the training process and the trainee may feel completely confused. At the same time we will investigate whether the fact that an author selects one of the alternative pre-constructed forms of presentation of the final application restricts his/her imagination and creativity hindering in this way the production of innovative applications. In this respect, we have to take into account that this approach will not have a negative effect to the performance and the quality of the interface of the final application.

Endnotes

1. HILDE is a three-year national project partly and indirectly CEC-funded, which aims at the development of a flexible and integrated environment for building hypermedia training applications. The consortium is composed from 3 Universities and 7 private companies (training institutes, s/w companies and industrial users). Project management is realised by 01-Pliroforiki S.A.
2. In the rest of the paper the term “meta-author” will refer to the user of the tools offered by the KMSE.
3. In the sequel, the term “author” will refer to the user of the Authoring tool.

References

- [1] U. Eco, Hypermedia for Teaching and Learning : A Multimedia Guide to the History of European Civilization (MuG). In: D. Lucarella, J. Nanard, M. Nanard, P. Paolini (ed.), Proceeding of the ACM Conference on Hypertext. ACM Press, Milano, Dec. 1992, pp. 288.
- [2] F. Garzotto & P. Paolini, HDM - Model -Based Approach to Hypertext Application Design, *ACM Transactions on Information Systems*, **11** (1), Jan. 1993, pp. 1-26.
- [3] F. Halasz & M. Schwartz, The Dexter Hypertext Reference Model, *Comm. of ACM*, **37** (2), Feb.1994, pp. 30-39.
- [4] A. Hammond, J. McKendree, W. Reader, A. Trapp, The PsyCLE Project: Educational Multimedia for Conceptual Understanding. In: Proceedings ACM Multimedia 95. ISBN: 0-201-87774-0. ACM Press, San Francisco, Nov. 1995, pp. 447-456.
- [5] M. Panagiotou, Students’ cognitive and knowledge quality model Ph.D. Thesis, University of Athens, Dept. of Informatics, 1996.
- [6] A. Rizk, L. Sauter, Multicard : An open hypermedia System. In: D. Lucarella, J. Nanard, M. Nanard, P. Paolini (ed.), Proceeding of the ACM Conference on Hypertext. ACM Press, Milano, Dec. 1992, pp. 4-10.
- [7] S. W. Smoliar, J. A. Waterworth, P. R. Kellock, pianoFORTE: A System for Piano Education Beyond Notation Literacy. In: Proceedings ACM Multimedia 96. ISBN: 0-201-87774-0. ACM Press, San Francisco, Nov. 1995, pp. 457-465.
- [8] N. Streiz, J. Haake, J. Hannemann, A. Lemke, W. Schuler, H. Schütt, M. Thüning, SEPIA: a Cooperative Hypermedia Authoring Environment. In: D. Lucarella, J. Nanard, M. Nanard, P. Paolini (ed.), Proceeding of the ACM Conference on Hypertext. ACM Press, Milano, Dec. 1992, pp. 11-22.
- [9] A. Tsalgatidou, C. Halatsis, M. Spiliopoulou, M. Hatzopoulos, A Multimedia Title Development Environment (MTDE). *Information Processing & Management*, **31** (1), 1995, pp. 101-112..
- [10] A. Tsalgatidou, Z. Palaskas, Halatsis, M. Hatzopoulos, An Integrated and Customisable support system for building Hypermedia Training Applications. In: D. Karagiannis (ed.), Proceedings of the 5th Conf. on Applications of Databases & Expert Systems. SPRINGER-VERLAG, 1994, pp. 540-549.

ACKNOWLEDGEMENT

We would like to thank the rest of the members of the HILDE group for their contribution to this work and particularly the project manager M. Panagiotou for his fruitful contribution to cognitive research on intelligent educational techniques.