The concept for a flexible authoring and learning environment

Jochen Wittmann, D.P.F. Möller, University of Hamburg
Corresponding Author: J. Wittmann
University of Hamburg, Department Computer Science, Chair Computer Engineering
Vogt-Kölln-Str. 30, D-22527 Hamburg, Germany
Phone: ++49 (0)40 / 42883-2437, Fax: ++49 (0)40 / 42883-2551
e-mail: wittmann@informatik.uni-hamburg.de

Abstract

This paper describes the concept for a flexible authoring and learning environment. It starts defining the requirements for the important flexibility of learning systems in respect to the current background of the learner, the dynamic ordering in which content is presented to the learner – also with consideration of the individual progress a learner makes – and the opportunity for the learner to proceed on many different paths in an exploratory way of learning.

To reach these demands a concept for a system architecture is proposed. The basic idea is to split the content into smaller units than usual, the so called 'content units'. Only doing so, the needed flexibility for individually adapted content, for an efficient handling of larger sets of content units, and the reusability for economic authoring can be assured.

On the other hand however, if the stored units get smaller than usual, a growing effort has to be spent to synthesize new content pages which are suited for the current learning situation and the current individual learner. It is the task to find approbiate content units which build a semantic senseful page unit. That is the reason why all relations between content units have to be explicitely stored by the system. This metadata set for each content unit includes technical information as well as semantical one. It splits in the parts

- technical information
- didactical information
- semantical information.

Following from this, the author has the task to provide these information for each content unit and to connect new content units to the already existing ones by reasonable relations. Afterwards the system has to evaluate the inter-related content units when a content page is asked by an user during his learning session.

The paper demonstrates the proceeding during the authoring and the learning phase by giving some details about the user classes with their roles and rights and by specifying those program modules which support the author in editing content units and in defining the relations between them. The following user classes are explained:

1. learning persons,
2. teachers,
3. authors,
4. system administrators,

To provide software support in managing the content units with their relations two program modules are necessary:

1. content unit designer:
   the content unit designer is an editor tailored especially to the demands of editing learning material for the learning system. It is based on a set of given templates and a given set of element types to fill the pages.
2. relation manager:
   The set of relations is extendable by the authors and thus can be individually adapted to cover special semantic restrictions and rules coming up from the context and/or background a content unit or even a complete course has. The specification and the administration of the relations is done graphically by linking graphically represented objects by labelled lines.

1 Introduction

Existing learning systems suffer from different deficits (q.v. M.J. Rosenberg [Ros2001]). First of all the learning often is not authentic. The presented learning material is irrelevant or inappropriate for the learner. The learners background can't be taken into account, because the programs lack the flexibility to adopt themselves to the learners specialist terms or can't dynamically integrate examples that are taken from the learners surroundings. If an author wants to reuse content within another context he is often forced to "copy and paste" the material. In other cases the content is out of date or simply incorrect because new facts can only be introduced by working through every single page.

A major problem for existing learning environments is the "One-size-doesn't-fit-all"-problem. The existing programs can't find the right pace and the right content for the particular learner. Presenting information in multiple contexts and multiple points of view is a fundamental postulation of pedagogues (q.v. Ch. Grune[Grun2000], J. Gerstenmeier & H. Mandl [Gers1995]). A simple solution might be to offer multiple modes but this often raises the costs to prohibitive levels.

Most programs do not support a good way to search for specific information. So if the learner needs a specific knowledge after he has finished his course, he often can't find that bit of knowledge without working a sequence or worse the whole course through once more. In this case the learning system is useless after the initial use.

2 The demands for flexibility

On the following pages the paper sketches a solution for all the problems specified in section 1. With regard to the acquainted papers of our working group (LIT) five demands on the flexibility of learning systems are motivated [Kope2000]. After that we will introduce the idea of a new software architecture for a learning and authoring environment.

Demand 1: Adaptability of the course content in accordance to the background the learner comes from. Course material written once should be adaptable to learners coming from very different application areas by using their terminology, by organizing and labelling data in the way they are accustomed and by using examples coming from their own field of interest.

This means flexibility of the content of a course in dependence on the application area and the background of the learners (q.v. S. Mällinen [Mäll2001]).

Demand 2: Very often the Computer Based Training (CBT) material is very elaborated but it is very inflexible concerning the sequence a learner has to follow working through. Naturally, this fact leads to a lack of flexibility if a course has to be applied to a new field of application or shall be used only partially. In these cases, the content has to be restructured which implies the need to copy the pages with the content the course author needs and giving them an alternative sequence. This functionality is given by most of the learning systems, the question however is, whether the so restructured pages keep consistent concerning their content. Often the content will have to be rewritten to avoid inconsistencies and false or missing links.

Formulated as a demand on a learning system: A really modular content management on page level is necessary, giving the opportunity to arrange the content in the sense of a real kit without any redefinition of the relations and interdependencies between the page contents.

Demand 3: One of the most tedious situations for the learning person is the fact that the material is not well adapted to the level of knowledge this person has. If the contents are too easy the learners will become bored and their concentration on the page will decrease rapidly, if the contents are too difficult the learner does not feel enough progress in his session and will be demotivated as well.

Therefore the demand is to provide flexibility concerning the level the learners enter the system and to adapt the standards and the severity of the content dynamically in dependence on the interactions and the progress a learner makes during a learning session.

Demand 4: The fourth level of flexibility comes up when we have a look to the course of a learning session and the interactions between learner and CBT system. Very often, some content is read and worked through repeatedly (e.g. it is obvious to make an example before the theory- and to redo it after the theory is understood) or the user is redirected to the material on special pages more than once. In all these cases it is better for the learner to find some material and content which is different to the example and/or the data he has seen at his last page call. If the system shows always the same example, the same data, he will not be able to understand the common link between the example data and will not find the abstract sense of the

examples. However, if the system changes examples and data automatically with each page call, the user has the chance to switch through different examples and therewith through different sights to understand the explaining part of the page which naturally stays static while the example changes dynamically.

Therefore demand four postulates a dynamical and automatic change of page content in relation to the position of the page call within the course of the learning session.

Demand 5: A completely new aspect on the structure of a CBT system is introduced by the fifth demand. Didactical investigations show that best learning results are achieved by the concept of exploratory learning. This term means that the system gives the learner stimulation to work through the content by his own exploration on an individual path. There are different situations which allow to manipulate data, abstract formulas, and graphical representations to see and understand the influences of the manipulation on the different representation of the content or the learning target and the interdependencies between these representations. This idea is explained in much more detail in the papers of R. Schulmeister [Schu2001] and can not be treated in its full depths here.

For the architecture of a learning and authoring system however the demand for the possibility to achieve an exploratory learning process leads to the flexibility level five. It extends the fourth which concentrates on the modular exchange of content elements on a page during a session to the flexibility concerning even a single page call: The elements on a page should be flexible linked to each other to allow reactions on user interaction in the sense of exploratory learning.

For the architecture of a CBT system follows from these demands that there are two alternatives for a software designer to fulfil them by an implementation of a software system. Firstly, to elaborate all the possible variants on all the possible ways through the content explicitly, or, secondly, to inherit the flexibility from the objects and the structure of the CBT system.

The existing software systems do not suffice the demands for flexibility (see also [Schu2003]). Therefore a new approach is necessary. The following chapter will sketch the ideas for this new software framework.

3 A proposal for a new system structure

The architecture of the system which is developed in our group is explained by

Figure **1**. It depicts the interrelationships between users, objects, and program modules. The explanation is followed by the four classes of users and their interactions and concentrates on the generation and the display of learning modules. All the functionalities which common learning environments offer but do not influence to the flexibility demands mentioned above (such as chat, user administration, etc.) can be added easily and shall neither be discussed in this picture nor in this paper. However, the here shown functionalities seem to be the most demanding part of the system and in this range of functionality the new architecture differs most from existing learning environments.

2.1 The user classes

To start, let us explain the four different user classes which represent the persons who work with the system:

5. learning persons,

   are the persons who are registered for one or more courses and work within the learning system.

6. teachers,

   are the persons who manage a course concerning its content and its organisation.

7. authors,

   are the persons who write new content in the form of new learning modules and who set the new material into relation to already existing content units.

8. system administrators,

   are the persons who have complete access to all the data and objects stored in the system to maintain them and to solve technical problems.

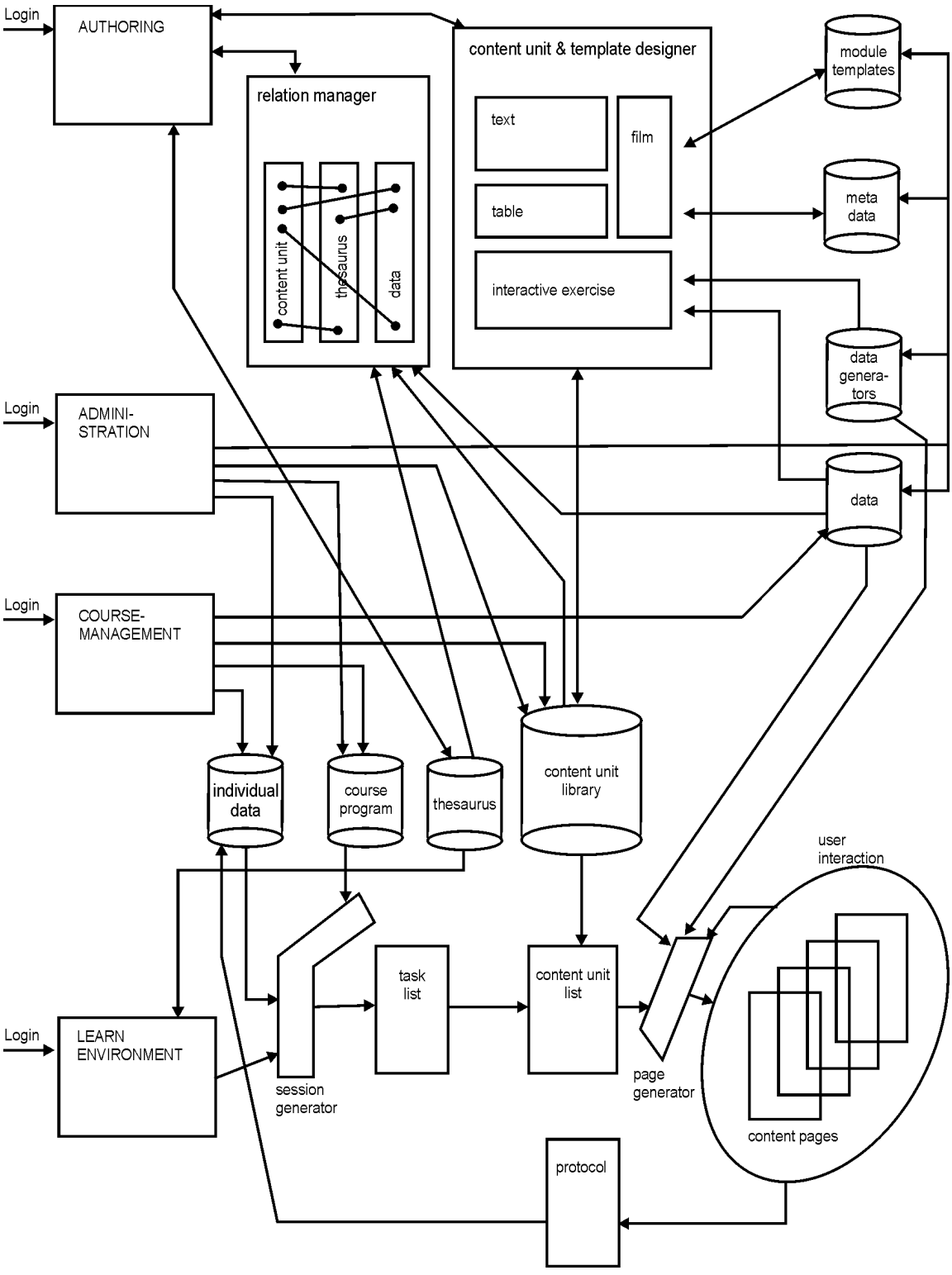

Figure 1: System architecture overview

The tasks of the system administrators are evident. They have full access to all data stored in the system. However, if the learning environment is operating, they will not have to interfere. That's why this class needs no further explanation in this context.

The teaching persons have two kind of roles: On the one hand they have administrative functions on the other hand they are online and/or offline contact person for the learners. From the view of software technology the administrative work (e.g. to build up a course from already existing modules, to create and maintain a list of participants of the course, etc.) will be supported by a database system. This part of the system is very conventional and does not differ from existing learning environments. The same observation can be made for the interactions with the others persons registered by the learning system. The corresponding software modules can be found ready made and can be integrated easily without any complication in the relation to the content management system. Therefore, this user group can be handled quite conventional and profits of the modules and relations the authors brought into the system.

For the user classes teacher and learner the steps during a session with the new system shall be explained in more detail:

The learner enters the learning environment which offers the well known features as chat, forums, etc. and starts a learning session. To do this, a program module called session generator generates a task list with the topics still to work through. Generating the tasks the program of the course he logged in and some individual data of the learner such as learning level and/or recent progress within the course are taken into account.

The task list gives the content and the learning target for the learning session in an abstract form which has to be transformed to a set of learning modules of the learning system.

These learning modules are highly parametrized dynamically and therefore have to be filled dynamically by adding content units, e.g. text, pictures or examples, which correspond to the learning progress, the background, and the course of the person who logged in. This is the task of the so called page generator.

Afterwards a set of fully parametrized pages which relate to each other consistently is available for processing during the session. However, even than the flexibility demands require adaptability caused by user interactions. For example, the page generator will have to be re-called to offer changed data sets within a repeatedly called example or an exercise has to be updated due to the reached learning level. Therefore, all the interactions and learning steps the user makes are logged for online or offline evaluation and added to the individual data-set of the learner.

The most demanding task for the system is the way new content can be integrated. In the terms of the system architecture the question must be answered how the content unit library can be augmented with new content. The user class which is allowed to do this is the class of the authors. Authors will get a special user interface to interact with the system and are supported by two quite independently working software components. The first component is the content unit designer, which focuses on editing the content units. The other component is the so called relation manager, which supports the author linking the edited content units to existing material.

2.2 The content unit designer

The content unit designer is an editor tailored especially to the demands of editing learning material for the learning system. It is based on a set of given templates and a given set of element types to fill the pages. The use of templates has two reasons: First should the resulting content be displayed in an uniform layout and secondly and here is a great difference to other systems do the templates imply semantic restrictions the author has to hold. For example the structure of an interactive example fits only to the corresponding template and can not be edited using the template for an explaining text.

The restrictions on a given set of elements on the content pages are caused by semantic reasons as well. Only if the type and the usage of a media element used on a content page are known the system can achieve a proper display. In addition, the system can automatically substitute content elements. This dynamical behaviour is necessary to present the learning material in different ways. So if a learner doesn't understand the presented information he can redo the unit and gets a different entrance to the learning target.

The content units used on the content pages are texts, pictures, films, graphs, formulas, tables and so on. A special class of elements are the data elements. They can be filled either by access to the database of the system where the example data lies or by using a data generator which generates the data to display at the time of the page call dynamically.

Thus new material can be edited and persistently brought into the systems content unit database as a new content unit using such media elements within a semantic restricted module template.

2.3 The relation manager

One important step is missing even if the content unit is completed. It has to be linked to the other content units of the kit semantically. This task has to be done by hand of the author but can be supported by the relation manager. This program module gives the author the opportunity to define relations between

- new and existing content units
- new and existing content units and their media elements
- new and existing media elements without their representation in a particular course
- content units and the thesaurus of the system
- content units and media elements to courses
- content units and media content to application areas
- etc.

The set of relations is extendable by the authors and thus can be individually adapted to cover special semantic restrictions and rules coming up from the context and/or background a content unit or even a complete course has. The specification and the administration of the relations is done graphically by linking graphically represented objects by labelled lines. Examples for possible relations could be: “is subordinate term to”, “contains example to”, “needs data from”, “is explained in”, etc.) The huge set of relations can be managed concisely by restricting and filtering the object classes and relations involved. Thus, despite the enormous amount of relations the system holds the author can work with a concise subset and do the important work of linking the content elements semantically with justifiable effort.

3 Conclusion

The system concept explained here on a very top level under respect to the page restriction for this paper shall be concluded by the following assumptions:

1. To efficiently build reusable CBT systems the demands for flexibility have to be considered.
2. Existing systems fulfil these demands only partially. Especially there is a lack of consistency between authoring systems and learning environments. Therefore users of existing systems reach the flexibility only by explicitly programmed content pages which are hardly reusable.
3. To obtain flexibility the internal structure of a content page has to be taken into account and the relations between the content units have to be specified and administrated.
4. The system concept proposes an architecture which is based on the described content units which represent the content elements of a page. All flexibility and dynamic behaviour results from a set of user defined relations which are administrated by a special graphical program module within the system.
5. The presented architecture for an authoring-, learning- and content management system in this paper can easily be extended with standard features provided by software systems of the competitors.

This leads to an integrated, complete and efficient CBT system.

4 References

[Gers1995] Gerstenmaier, Jochen & Mandl, Heinz: Wissenserwerb unter konstruktivistischer Perspektive. From: Zeitschrift für Pädagogik, volume 41, issue 6, p. 867-888; 1995

[Grun2000] Grune, Christian: Lernen in Computernetzen. Analyse didaktischer Konzepte für vernetzte Lernumgebungen; KoPäd Verlag, München; 2000

[Kope2000] Koper, R.: From change to renewal: Educational technology foundations of electronic learning Environments; Educational Technology Expertise Center, Open University of the Netherlands, p. 8-15; 2000

[Mäll2001] Mällinen, Sisko: Teacher effectiveness and online learning; From: Stephenson, John (Edt.): Teaching & Learning Online. Pedagogies for New Technologies; Kogan Page Limited, London; 2001

[Schu2001] Schulmeister, Rolf: Virtuelle Universität, Virtuelles Lernen; Oldenbourg Verlag, München; 2001

[Schu2003] Schulmeister, Rolf: Lernplattformen für das virtuelle Lernen. Evaluation und Didaktik Oldenbourg Verlag, München 2003

[Rose2001] Rosenberg, Marc J.: E-Learning. Strategies for Delivering Knowledge in the Digital Age; McGraw-Hill, New York; 2001

[Witt2002] Wittmann, J. (2002): Learning Systems for Environmental Informatics; EnviroInfo Vienna; 2002

Filename: AuthoringLearning_163-Wittmann - Jochen Wittmann, D.P.F. Möller, University of Hamburg
Directory: G:\session III_fullpapers
Template: C:\Documents and Settings\yonatan\Application Data\Microsoft\Templates\Normal.dot
Title: CONFERENCE PUBLICATIONS
Subject:
Author: bpotot
Keywords:
Comments:
Creation Date: 9/1/2004 10:14 AM
Change Number: 3
Last Saved On: 9/1/2004 10:23 AM
Last Saved By: Nega Alemayehu
Total Editing Time: 2 Minutes
Last Printed On: 9/22/2004 6:04 AM
As of Last Complete Printing
Number of Pages: 7
Number of Words: 3,171 (approx.)
Number of Characters: 18,076 (approx.)