



**HAL**  
open science

## Software Interoperability Problems and E-Learning

Ingo Dahn

► **To cite this version:**

Ingo Dahn. Software Interoperability Problems and E-Learning. 2nd International LeGE-WG Workshop on e-Learning and Grid Technologies: A Fundamental Challenge for Europe, 2003, Paris, France. pp.6. hal-00190455

**HAL Id: hal-00190455**

**<https://telearn.hal.science/hal-00190455>**

Submitted on 23 Nov 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Software Interoperability Problems and E-Learning

Ingo Dahn  
University Koblenz-Landau, Knowledge Media Institute,  
Universitätsstr. 1, D-56070 Koblenz  
*dahn@uni-koblenz.de*

## Abstract

**Grid applications are special cases of networking applications. In order to investigate potential applications of Grid technologies to e-learning we discuss in the following some current applications of network technologies in e-learning as they have occurred at the University Koblenz-Landau. At the appropriate points we shall indicate demands that our e-learning applications pose to the underlying network services and mention, how they might benefit from resource sharing as potentially offered by Grid services.**

*Keywords: Personalization, Textbooks, Slicing Book Technology, Learning Management Systems, Software Interoperability.*

## 1. E-LEARNING AT THE UNIVERSITY KOBLENZ-LANDAU

The Knowledge Media Institute Koblenz is a central institute of the University that coordinates e-learning activities in Koblenz. It carries out research projects, pilots e-learning technologies and provides services for other departments.

However, courses at the university are held in two different locations – Koblenz and Landau – which are 160 km away from each other. For the university it is therefore convenient, to run the e-learning platform simultaneously at two sites, eventually even more distributed to servers at the institutes that are responsible for the individual courses.

In fact, the problem of exchanging material between different e-learning servers is even more severe in universities where e-learning is growing “from below” initialized by engaged but uncoordinated work in different departments even using different platforms for their students. The Grid may become a framework that allows students to locate, access and share the necessary learning resources wherever they are.

Within its research activities the Knowledge Media Institute in Koblenz is involved in the European project Trial-Solution (IST-1999-11397) and in the project In2Math funded by the German Ministry of Education and Research. It pilots the e-learning platforms WebCT and Hyperwave ELS.

The work described below builds on results from these projects.

## 2. LEARNING MANAGEMENT SYSTEMS

Sharing heterogeneous resources is a major feature supported by Grid technologies. In the following we describe that share resources of three different kinds.

- A Learning Management System (LMS)
- A personalization system for teaching materials
- Computationally demanding services

All these are used concurrently by a number of students that form a learning community.

The LMS WebCT [4] is a web based system that provides a unified access to teaching materials, to communication and collaboration tools for several courses held in Koblenz. This platform gives also tutors the possibility to publish and grade tests online. WebCT is one of the most widely used LMS systems.

For the following it is worth noting that LMS are fed with content from tutors (course designers) or by importing courses from commercial vendors. Also students can upload content to restricted areas.

Given that qualified tutors want to shape their courses by themselves, material uploaded by the tutors is by far the most common source. This, as well as the current prices of course packs, implies that LMS will be in most cases fed only with a restricted amount of content. Making a network connection with a large content repository, as described below, can change this situation and pose new challenges to the development of LMS.

Students working with an LMS are exposed to the content uploaded by the tutor. There is no possibility for the learner to adapt the presented content to his needs and there are no possibilities to generate the content dynamically for a particular student in a particular learning situation. These possibilities may be provided by other systems as described below. Then there arises the challenge to let these systems and the LMS interoperate.

Learning platforms allow tutors to restrict access to certain teaching materials for certain groups of students, for example for students who have not passed a certain exam yet. To preserve this possibility, each component of a distributed LMS would have to use the same access policy. As far as this policy is based on grades achieved, the information on these grades must be synchronized for each student between the various components.

This information is highly sensitive and privacy issues have to be taken into account. In fact, grading students is one of the major applications of current LMS. These systems automate the assessment of answers to simple tests (like multiple choice), but they also provide the possibility that a human tutor assesses free-text answers of the students submitted electronically or on paper. In the latter case the LMS can provide statistics and can give the individual student a concrete indication of his position within a course. We shall describe further applications of this feature that can be achieved through networking in the next section.

There are good reasons to establish a connection from within an LMS to other systems outside the university, for example to servers of publishers that offer relevant teaching material. The following example connecting an e-learning server with a publishing server has been implemented in Koblenz.

### **3. CONNECTING A LEARNING MANAGEMENT SYSTEM WITH A TEXTBOOK PERSONALIZATION SERVICE**

A major problem of e-learning is the availability of high quality content in appropriate forms. The production of special multimedia courses is very expensive since it requires competencies beyond the expertise of most authors. Not only that the content must be available in an appropriate format, it must moreover meet the needs of a particular course. Ready-made course packs are only very seldom fully appropriate. We consider this as a main reason why e-learning content is only seldom reused outside the organization which produced it. Especially at universities, where highly qualified teaching personnel are in charge of the teaching process, possibilities to select, modify and adapt material from other sources are an important need.

Courses frequently reuse teaching material, notably existing textbooks. The parts of a given textbook that are needed for a course vary greatly from course to course. Even different students may prefer material from different books and the level of detail needed by the same student may vary depending on whether he is at the beginning of his study or shortly before an exam.

Slicing Book Technology [1] serves these varying needs for reuse by slicing existing teaching materials into small semantic units and composing new documents from these units on the fly, tailored to the learner's current needs.

In order to compose meaningful documents, a server uses metadata describing the content and conditions for reuse of these semantic units. It also takes into account information about the user, especially about his acquired knowledge, in order to reduce the size of the delivered document.

Within the project Trial-Solution, funded in part by the European Commission within the 5<sup>th</sup> Framework Programme, a server has been set up with a library of 20,000 semantic units taken from 5,000 pages of textbooks on undergraduate mathematics. It is accessed by students from several universities in Europe to generate personalized documents. For example, students may request documents from several sources that provide guidance for solving a particular exercise. These documents can be configured such that they take the students from the facts they know already to the learning objectives without detours.

However textbook material, even if it contains exercises, lacks the interactivity that is necessary to provide a valid assessment of the learners' knowledge. Therefore the Trial-Solution Delivery Tool relies on the user to provide information on his knowledge.

On the other hand, computer based training systems and e-learning platforms provide assessment tools, but they lack the wide variety of contents that is available in textbooks.

#### **4. MAKING A STANDARD WORK**

We investigated a way in which these particular strengths of Textbook Personalization and e-Learning could be combined using network services. This was implemented<sup>1</sup> as an online communication between the Trial-Solution Delivery Tool Version 1.0 provided by Slicing Information Technology Berlin and the e-Learning Platform WebCT Campus Edition Version 3.8 running at a server at the University in Koblenz.

It is easy to access a web based system like the Trial-Solution Delivery Tool from within an e-learning platform by embedding a link to its login page within a course. A single-sign-on system for these servers, ideally also connected to a central student administration system of the university, would provide greater comfort here, allowing the students to use always that server that can serve their needs best. This requires that the relevant material is available and up-to-date on all addressed servers. Automated updating and synchronization procedures have to be installed for this purpose.

The Trial-Solution Delivery Tool lets the teacher compose the documents he needs for a particular purpose using advanced search and advisory services. He can request these documents from the server as pdf files which can then be presented within the course material. For example he may select particular exercises from the book for an exam and present in several static documents

1. the selected exercises
2. a specific exam preparation material leading from the presumed knowledge of the students as directly as possible to the selected exercises
3. for each exercise the prerequisite knowledge needed for the solution of this particular exercise

Since all these documents can be produced by the Trial-Solution Delivery Tool dynamically on the fly for each student individually, providing them as static documents is clearly below what would be possible using advanced network services.

As a second step, the tutor may compose the documents as personal books in the Delivery Tool but chose to export only his selection of exercises instead of the ready-made pdf files. Through

---

<sup>1</sup> Implementation was provided by Philipp Letschert and Pablo Lachmann (University Koblenz-Landau, Knowledge Media Institute) and Karsten Tabelow (Slicing Information Technology GmbH, Berlin)

the e-learning platform he can publish these selections to the students which can then import them, eventually with additional annotations from the teacher, as their personal books into the Delivery Tool. They can further adapt this selection and then generate their individual pdf document. This requires more steps but leads to more personalized documents.

At a third level, we tried to import content from the Delivery Tool into WebCT – not as a collection of separate static pdf files but as a course module. This gives the course designer the possibility to apply the personalization features of WebCT, for example to hide certain content from students before they have passed some test.

A favourable condition for this import was, that WebCT as well as the Trial-Solution Tools support the export/import of content packages formatted according to the open IMS Content Packaging Specification. Nevertheless we encountered a number of difficulties.

The first problem was that the Delivery Tool uses content packages which have the content as LaTeX source code. This had to be converted by an additional tool into pdf.

A second problem was that the current version of WebCT does not allow the import of content modules. Instead it supports only the import of complete courses. We circumvented this by creating a course with an empty content module within WebCT, exporting this course as an IMS content package, replacing the empty module by a content package from the Trial-Solution project and finally re-importing the resulting course.

In fact, this did not work that easy, since it turned out that the Trial-Solution project and WebCT use the IMS Content Package Specification in a different way. Trial-Solution aggregates the slices of a book as IMS item elements organized by an IMS organization element which is a refined table of content. WebCT, on the other hand, aggregates content modules by using nested IMS manifests. Some XSLT programming was necessary to convert the Trial-Solution XML file that describes the organization of the content of the book to be imported into the form expected by WebCT.

Having the book successfully imported into WebCT we found that it took several minutes to display the table of content from WebCT in a browser. The reason was, that the amount of information transferred from WebCT to the browser was very large since WebCT tried to display the tables of content completely unfolded – we created two courses with the sliced books [5] and [2] having 1,300 respectively 3,000 entries in the content modules. Since WebCT is a relatively open system we found a way to let WebCT display the table of content as a folded tree which ensured an admissible working speed.

A result of this experiment was the finding that such a fine grained slicing (5 slices per page in the average) is not useful when static documents are presented as in the WebCT content modules. However it increases flexibility very much when larger documents are to be created dynamically from the slices.

A further step is the import of some of the meta information available in the Trial-Solution tools to WebCT. For example the navigation titles used in the Trial-Solution Delivery Tool were used also for navigation in the corresponding WebCT content module. This confronted us with the problem that the German umlauts in these titles were not correctly handled by WebCT. We experimented with several encodings but could not find a satisfactory solution until now.

A thesaurus of key phrases is available for search in the Delivery Tool. Each book in the Trial-Solution Library uses several hundreds of these key phrases. In this tool the reader can enter a part of a key phrase and he will get in return a list of all key phrases that contain this part. He will also get a table of content containing the places where appropriate content was found. This thesaurus was imported into WebCT as an index with its entries linked to the corresponding parts of the imported book.

## **INTEROPERABILITY OF ASSESSMENT AND PERSONALIZATION**

A more ambitious task than the import of Trial-Solution materials into WebCT is, to provide a closer integration of both systems. The Trial-Solution Delivery Tool relies on the input from the student himself in order to build an internal model of the student's knowledge. Making a connection with the assessment facilities of WebCT could improve the user model considerably by getting the tutors involved.

Fortunately, WebCT is a relatively open system. Besides providing open APIs for basic functions, it is possible for the administrator to access many data also directly. Many desirable extensions of WebCT – like those described in this paper – need only read access to these data and can therefore be implemented without a risk of interference with the standard functions of WebCT. This makes WebCT suited as a framework for the investigation of new e-learning services.

The connection between the Trial-Solution Delivery Tool and WebCT has been realised in the following form. Exercises from a textbook have been imported into WebCT by a tutor. In our case it was not possible to evaluate the correctness of solutions provided by the students automatically. Instead, a tutor would assess and grade these solutions within WebCT.

Then the tutor can transfer the assessment results for the students from the course over the Internet to the Trial-Solution Delivery Tool. The transfer software accessed the user data within WebCT, infers that the learner knows certain slices or topics and sends this information to the Delivery Tool. That system will check whether the students have a valid account. If this is confirmed, their user models are accordingly updated. For example it will be inferred that the student must have understood other subjects too. This will lead to shorter documents when these students log into the Trial-Solution personalization system next time and request documents for a certain purpose since slices that are inferred to be known will be omitted.

## **5. CONNECTING PERSONALIZED TEXTBOOKS WITH INTERACTIVE SERVICES**

Textbooks and their personalized versions provide information for the reader. However modern electronic formats like HTML and pdf can carry embedded elements that allow the integration of interactivity. Such a combination of personalized pdf documents with web based interactive systems was investigated within the In2Math project.

This project uses the SIT Reader from Slicing Information Technology for personalization and presentation. The author embeds in his LaTeX source of the book special commands which contain as a parameter the name of a program that can be executed on the server where the SIT Reader server is running. The SIT Reader will compose a special document source for the reader. However before generating the pdf it will scan the source for these special commands and will execute their parameter programs. These programs can access the particular user model. They are expected to produce a piece of document source that replaces the special command under consideration.

Only when all these special commands have been evaluated, the pdf document is generated for the reader. In this way, parts of the document have been produced by the embedded programs at runtime.

As an application of this, each learner can get randomly generated exercises adapted to his skills. A number of these service programs have been implemented in Koblenz by a group of students under the supervision of Peter Baumgartner to extend the sliced book [3].

A specially interesting service program (from the point of view of Grid technology) takes a set of formulas entered by the learner into a form in the pdf document and passes it on to an automated theorem prover running somewhere in the internet on a third machine. This automated prover will, when successful, produce a proof of a consequence from the formulas the user has provided. Locating and accessing this prover on a server with free capacity may be well supported by Grid technology.

The prover will produce a proof protocol and send it back to the calling service program. That program will format the proof nicely in LaTeX and the SIT-Reader will embed this in a new version of the document for the learner.

The theorem prover used in this application is available only in a few places. It is a research system which must be maintained by experts and it has a high demand on computational resources. Nevertheless, this example shows how such systems can be used over the net to generate interactively personalized teaching materials.

## 6. CONCLUSION

We have shown practical examples how two different systems can provide e-learning services for each other over the net. In our case, the e-learning platform was hosted in Koblenz while the Trial-Solution Delivery Tool was provided by the company Slicing Information Technology GmbH in Berlin. On a larger scale, the teaching material would be hosted at various sites, run by the publishers and third parties offering their course materials. Of special interest would be to connect the course material servers of various universities with each other. Then course materials could become valuable shareable resources that augment each other.

A brokerage system will find the appropriate material for the course or augmenting material for the individual student. It may – as we have shown in the last example – also locate and access computationally demanding resources in the web.

Accessing this material and these computational resources could be mediated through a Grid service. This service would log the user onto the Grid and negotiate conditions for the reuse of the material. It might contact several servers to find the most convenient offer.

In order to adapt to the user's needs, the server may not only select the material but also adapt it in a certain way, for example by changing some notation to the one used in the student's course.

As we have seen, the cooperation of different e-learning systems requires the exchange of confidential data, namely information about the users' current interests and about their actual knowledge. This makes a secure environment with trusted connection necessary. Note that not only the institutions providing the services must trust each other but also the individual users must be aware of the systems involved and need trust that the information that is provided is secure. In our case the tutor that transfers the student data needs a general permission from the students for this action. A Grid environment can be an important contribution to establish this trust.

## REFERENCES

- 1 I. Dahn: Automatic Textbook Construction and Web Delivery in the 21st Century. (2001) J. Struct. *Learning and Intelligent Systems*, vol. **14**, pp. 401-414, Taylor & Francis
- 2 C. Gellrich, C. & Gellrich, R: (2001) Lehr- und Übungsbuch für Fachhochschulen (Band 1): Arithmetik, Algebra, Mengen- und Funktionenlehre. Verlag Harri Deutsch, Frankfurt
- 3 Furbach, U. Logic for Computer Scientists (2003) University Koblenz-Landau, Koblenz
- 4 <http://www.webct.com>
- 5 H. Wolter, I. Dahn. (2000) Analysis Individuell. Springer-Verlag, Heidelberg