DIGITAL LIBRARY: DESIGN AND SECURITY CONSIDERATIONS

Stanislav Mikulecký
Faculty of Informatics and Management, University of Hradec Králové, Víta Nejedlého 573, Hradec Králové, Czech Republic

Keywords: digital library, DILLEO.

Abstract: In the last two years a digital library of learning objects (named with an acronym DILLEO) has been implemented by the staff and students of the University of Hradec Králové. The intent of this paper aims mainly to introduce the features of the library and to point out some of the core design and security issues dealt with during the implementation process.

1 INTRODUCTION

Teachers at our university who develop e-learning support for the students would like to have some “building blocks” available for the course development. The students also ask for easy and convenient accessibility of all study materials, which are now split over the university intranet (or accessible only within courses being supported by learning management system, and that only when the student is enrolled in the course). The conclusion is that the teachers and students lack a repository where they can submit their materials, find related material, create new content, and, collectively improve both the quantity and quality of digital teaching resources. Therefore it was decided to build DILLEO - a digital library of learning objects.

Following paper sections summarize some interesting aspects of building such digital library. Borrowing RUP (Pollice, 2003) methodology dictionary, the most interesting FURPS+[1] requirements and provided solutions are dealt with, with the emphasis to the F(unctionality) and S(ecurity) parts.

2 FEATURES: A NEVER-ENDING STORY

Roughly speaking, the DILLEO is the collection of learning objects organized in the tree-like structure of topics with library features around it. Library features include mainly (but not only) the way to specify access rights to the contained objects, and groups of objects and easy-to-use object management system. In comparison to other similar library systems, DILLEO provides a complete web-based management and a generic multilingualism – user interface and object metadata can be specified in theoretically unlimited number of different languages, which makes library accessible to wide range of potential users.

The functionality the library provides to the user is dependent on the role the user is in; the user in librarian role will probably expect something different than ordinary registered user.

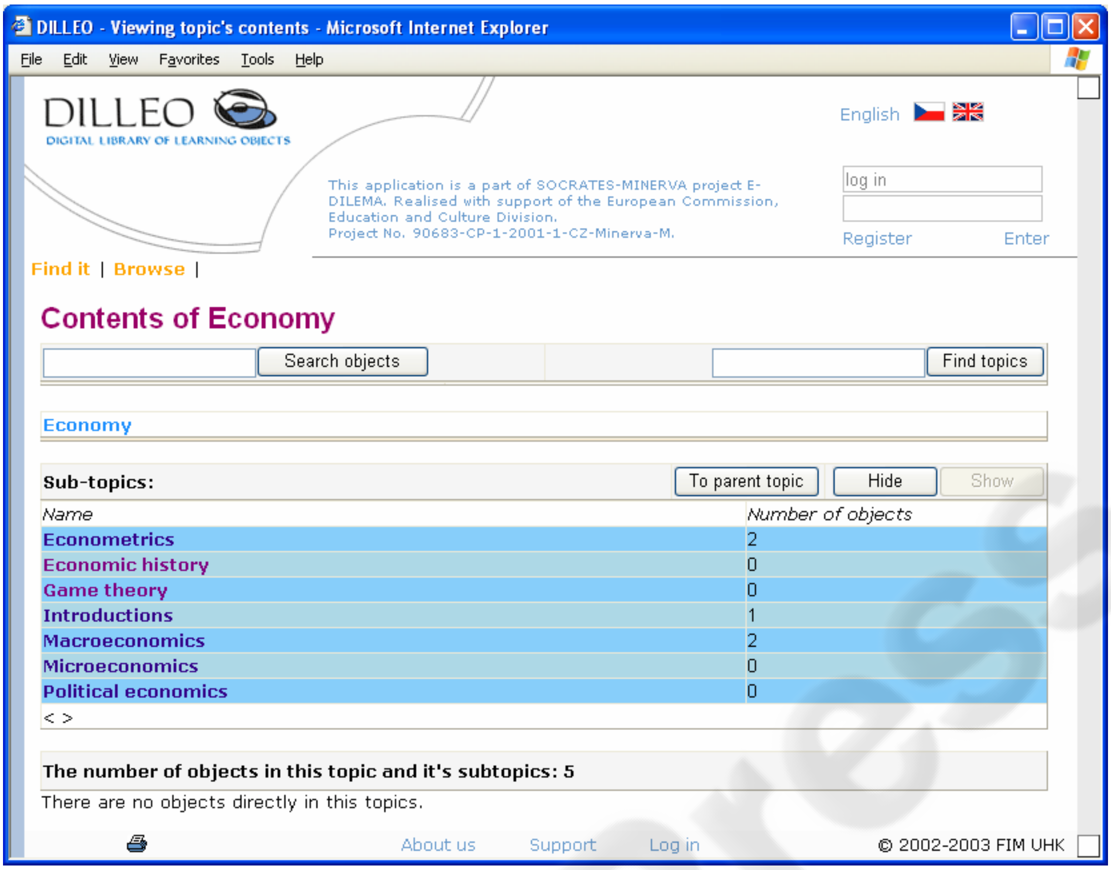
There are four main types of users (or roles) in the DILLEO system, so the functionality is described from four different points of view:

- not-registred user
- registred user
- librarian
- administrator

Not-registered user is a default type of user. Not-registered (and therefore anonymous) are all visitors of DILLEO library until they obtain username and password in the registration process. Not-registered user can search and view public objects, which can be just the small fraction of whole number of available objects in the library. Every not-registered user can undergo a simple registration process, in which he/she requests to become a registered user.

[1] FURPS+ - the acronym used by RUP during requirements-collecting phase of information system evolution. The IS requirements are divided into following groups: F – functionality, U – usability, R – reliability, P – performance, S – security, + - technological constraints.

Mikulecký S. (2004).
DIGITAL LIBRARY: DESIGN AND SECURITY CONSIDERATIONS.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 295-300


DILLEO - Viewing topic's contents - Microsoft Internet Explorer
File Edit View Favorites Tools Help
DILLEO
DIGITAL LIBRARY OF LEARNING OBJECTS
English
log in
Register
Enter
Find it | Browse |
Search objects
Find topics
To parent topic
Hide
Show
About us
Support
Log in

This application is a part of SOCRATES-MINERVA project E-DILEMA. Realised with support of the European Commission, Education and Culture Division.
Project No. 90683-CP-1-2001-1-CZ-Minerva-M.

Contents of Economy

Economy

Sub-topics:

Name	Number of objects
Econometrics	2
Economic history	0
Game theory	0
Introductions	1
Macroeconomics	2
Microeconomics	0
Political economics	0

< >

The number of objects in this topic and it's subtopics: 5

There are no objects directly in this topics.

© 2002-2003 FIM UHK

Figure 1: Example of DILLEO user interface

Registered users successfully went through the registration process and own username and password for accessing the library. Above all rights of non-registered user he/she earned right to search and retrieve all learning objects (with the exception of some objects with specifically assigned rights) in the library and also the right to submit new learning objects into the library.

Librarian has a higher rights for specified group of learning objects (usually one topic and its subtopics), as defined by the system administrator. In these topics he/she is responsible for accepting new objects submitted by registered users and making them available for registered and/or not-registered users. Librarian has right to move and delete learning objects in a topic and he/she is also responsible for proper object metadata management.

System administrator is responsible for continuous operation of the whole DILLEO system. He/she manages all users, roles and their access rights and also has unlimited access to all parts of the system. He/she can change all system settings (for example primary language of communication).

For better imagination of DILLEO functionality, it is advised to read the article (Mikulecký, 2003) or visit the library on the internet (DILLEO). The screenshot of up-to-date DILLEO user interface can be seen in figure 1.

The functionality introduced above had been divided into smaller more or less atomic parts, while each of them was implemented independently. Such parts are used to be called use cases. The list of core use cases is provided in the following enumeration.

- Main page
- About page
- Login
- New user registration
- Topics & objects browser
- View object detail
- File download
- Simple object search
- Advanced object search
- Top 10 objects
- Alphabetical list of objects
- Submit an object
- Web-service gate
- Edit an object
- Edit access rights to object
- Delete an object
- Accept new objects

- Add a new topic
- Edit a topic
- Edit access rights to topic
- Users list
- User detail & editation
- Find user
- Roles list
- Change user details
- Role detail & editation
- Users in role
- Edit rights to use case
- Access statistics
- View application log

Each use case is provided with the name of its main file (an access point) and the code – this information is used by the access rights resolving module.

As you can see from the list, the library construction process is not a task for one afternoon. Each functionality depicted in the table has from several hundred to several thousand lines of program code (and thus took from several tens to several hundreds hours to implement). Of course, DILLEO hasn’t been implemented by a single person – it is collective work of a number of people, including university students. For more information about implementation process please see the section 4.

Although DILLEO has provided wide range of features, it hasn’t been completely finished yet. As in any other complex information system, there are still areas in which it can be improved and optimized. New requirements vary from the minor user interface modifications to the changes with the impact to the system internal architecture. All requirements become registered and are going to be dealt with in the following iterations of implementation process.

3 SECURITY: HIDDEN BUT IMPORTANT

This chapter reveals important decisions that are connected with the overall security of the DILLEO library.

Our digital library is an application connected to internet and thus accessible by any internet user. All library functions including administration are technically accessible from any computer on the net. Therefore there is a strong need for secure mechanism how to distinguish users from each other and provide a mechanism to reveal only the functions that are relevant for that user. The security is guaranteed on two different levels:

- **general security**, referred also as infrastructure security provides the common HW and SW infrastructure for securing the internet application
- **application security**, referred also as internal security, which provides authorization to functionalities inside the application.

3.1 General Security

As the basic protection element the communication encryption (using HTTPS/SSL protocols) has been used. Digital library is always accessed from the web, every user, including librarians and administrators, logs into library by entering his/her credentials – username and password, which become during the authentication process sent from the client computer to the server. By eavesdropping on the communication and extracting the credential information a malicious user would gain a complete access to the library with the access rights of the communicating user. Thanks to the communication encryption, such attack is near to impossible.

Usernames and passwords are stored in the library’s persistent data structures – in database tables of MS SQL Server 2000. The server’s security is guaranteed by the security policies of the institution – in our case University of Hradec Králové. Rights to fully access data on the server are given only to network administrators and the special “virtual” user the library application uses for the connection to database server, so the attacker shouldn’t be able to read the library data, unless he or she breaks into one of these accounts or exploits some of the possible security holes in the operation system. However, the library does not rely on such premises and enhances the security even further – all passwords in the database are not stored in the readable plaintext, but in the hash form. For hashing the MD5 one-way digest algorithm has been used. The reconstruction of the plaintext password from the hashed value is very hard (one has to try all the possible combinations using brute-force). Even if the attacker gets as far as he/she is able to read the data in the database, it wouldn’t be much help for getting the credentials for logging-in into the library system.

3.2 Application Security

From the business point of view, the most important security element is the authorization to objects. When submitting object to library, the author (or distributor) often doesn’t want it to be accessible for download for all internet users. Usually, the author specifies something like "I want to share this only with the university staff". Therefore, the library has to provide means for grouping the users and allowing specifying their authorization level for any library object.

This business idea was realized by implementation of role based authorization system, which provides means for authorization to:

- *object*
- *topic*
- *functionality (use-case)*

As stated in the figure 2 all access rights in the system are defined for the role. The role in the library context has similar meaning as the users group in the Windows system. The cardinality of role-to-user relationship is n to n – each user may appear in one or more roles and an arbitrary number of users may be in each role.

The authorization rights for library objects and topics principle is inspired by the access rights to files and folders as implemented in the Windows OS, slightly simplified and modified to meet the needs of the digital library.

For the given role in the system, one may define authorization level for any object and topic. Possible levels of authorization are:

- *Access denied* – user cannot access the object or topic at all.
- *Read* – user may read the metadata of the object, or list objects and subtopics in the topic.
- *Download* – user may download the object, same as read for topic.
- *Write* – user is allowed to make changes to object or topic, even remove it from the library.

For the sake of implementation simplicity, the levels are inclusive – e.g. if one has write rights, he/she may also download and read.

The authorization for the object is quite straightforward – access rights of given role are defined by the authorization record. When accessing the object, the authorization module decides whether it can be accessed. The decision is based on the authorization record. If the record for the given object and role was not found, default authorization level is used.

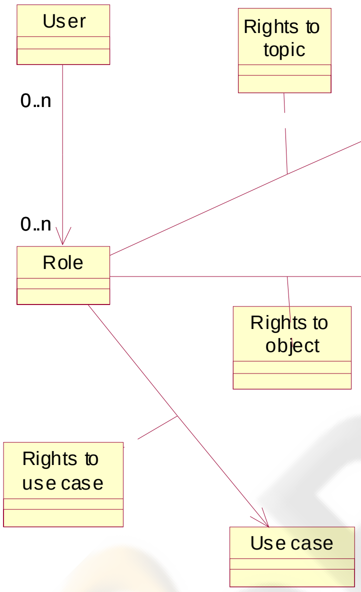


Figure 2: Authorization module overview

In the case of topic, this is a bit complicated. As were already mentioned, the topics are organized in a tree-like structure. The authorization level to given topic therefore affects the levels to any sub-topics. Therefore when accessing the topic, the system has to recursively check the authorization records for parent topics, until it finds relevant record. If the record was not found, the default authorization level is used.

Apart from access rights to objects and topics, the library defines also rights to run specific system use cases. Such is accomplished that some use cases may run only privileged users – for example it is not desirable to allow non-administrator to access the use case for administering users.

Authorization level for user being in more than one role is computed by the union of the levels for each role he/she belongs into – i.e. the least restrictive authorization level becomes used. Thanks to the fact that user can be in more the one role, it is possible to maintain such users that are for example librarians of one topic and just ordinary registered users for the rest.

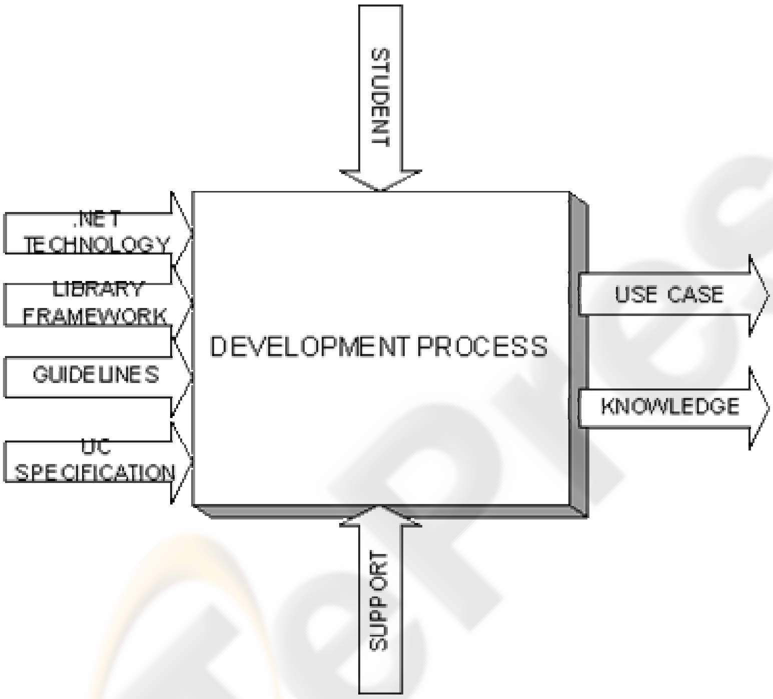


Figure 3: Implementation process from the student's point of view

4 DEVELOPMENT: THE CORE OF EVERYTHING

The physical design and implementation of e-library is not an easy task to do. Moreover, while designing and implementing DILLEO, we were limited by the students' involvement in the project. The students at our faculty usually don't have much practical experience with programming or working in a team, and they usually don't have much time left for such activities, so we had to think up and then provide the way for them to work on the project, without endangering the quality and security of the final product. Therefore we divided library into the large number of almost completely independent modules.

For each module, we provided:

- Detailed description of the functionality as well as the solution of potential technological problems, so that the student didn't have to reinvent the wheel. This also lowered the potential implementation and security problems.
- "Building bricks" – the complete e-library programming framework, solving the implementation of security and basic functionality so that the students solve mainly business specific problems (the F part of FURPS) and don't have to bear in mind the URPS requirements.
- The implementation guidelines – the set of "howto" documents providing the recommended solution of common implementation problems.
- Active support. Every student had an opportunity to come to regular weakly library workshop, as well as use tools provided by WebCT (discussion, chat, whiteboard, mail) to consult their problems.
- Final integration. Each student developed his/her use case individually and autonomously. After finishing it, software architect of the library was

responsible for checking the source code and integration with the rest of the library.

The implementation process from the student’s point of view is summarized in figure 3.

The student’s cooperation has brought advantages to both sides – students gained valuable experience, which has already been utilized in the “real” life by some of them and the overall implementation process of digital library went much faster. Moreover, it removed the implementation tasks burden out of the software architect, so that he was able to spend more time with analysis and design, which positively affected the quality of the library.

5 CONCLUSION

There are many interesting issues regarding digital libraries. In this article were covered the most interesting ones we came across during the development of the DILLEO digital library.

The DILLEO library has been implemented as a part of the E-DILEMA project No 90683-CP-1-2001-1-MINERVA-M (E-DILEMA).

REFERENCES

Mikulecký, S., 2003. DILLEO: The Digital Library of Learning Objects. In: DEL 2003 “Developments in eLearning” Intl. Conference Proc., Czech Tech. University, Prague, pp. 67-79

Pollice G., Augustine L., Lowe C., Madhur J., 2003. Software Engineering for Small Projects: A RUP-centric Approach. Addison-Wesley. 1st edition

DILLEO library instance (http://e-dilema.uhk.cz/dilleo)

E-DILEMA project web pages (http://e-dilema.uhk.cz)