Conclusion

The described above technology of remote organization of educational process is successfully developed at the Bank Faculty of the Kiev National Economic University. The implementation of this technology is carried out in parts and in the future it is planned inclusion of the educational programs of the satellite TV. This technology is used for preparation of the specialists of the various forms of training. However, it is observed greatest efficiency of the technology in the student's preparation process of the correspondence form of training, because such students have not stable possibilities of direct communication with the teacher and this technology makes such communication more active.

Bibliography

[Bondarenko, 2006] Victor Bondarenko Mobile Communication Technology as a Tool of Educational Process. Proceedings of International Conference "Modern (e-) Learning" July 1-5, 2006, Varna (Bulgaria). FOI-Commerce, Sofia, 2006, pp.229-232.

[Bondarenko, 2005-1] В.Є.Бондаренко Методологія побудови мультимедійних підручників. Удосконалення змісту та форм організації навчального процесу відповідно до міжнародних стандартів. Т2, Збірн. Матер.Науково-методичної конференції 2-4 лютого 2005 р., К.:, КНЕУ, 2005, с. 56-58.

[Bondarenko, 2004] В.Є. Бондаренко Інструментальний комплекс для формування адаптивних систем контролю знань. "Методичні та практичні аспекти застосування та розвитку системи контролю знань в університеті "Збірник матеріалів науково-методичної конференції. К.: КНЕУ, 2004, с. 173-176.

[Bondarenko, 2005-2] В.Є. Бондаренко Дистанційний тренажерний комплекс для навчання роботі з системою EXCEL і тестування отриманих навичок. Образование и виртуальность. Вып. 9.Харьков, 2005, с. 60-64.

Author's Information

Viktor Bondarenko – Kiev National Economic University; Pobeda ave., 54, Kiev-047, Ukraine, 03047; e-mail: victorbondarenko@rambler.ru

IMPLEMENTATION OF DISTANT LEARNING PORTALS BASED ON CASE-TECHNOLOGY METAS

Ludmila Lyadova, Alexey Urezalov, Andrey Khlyzov

***Abstract**: The paper describes an approach to the development of software aimed at the creation of distant learning portals integrated with education support and educational institution management systems. The software being developed is based on CASE-technology METAS which is used for the creation of adaptive distributed information systems. This technology allows to dynamically adjust the portal's structure and portal's functionality enhancements.*

***Keywords**: portal, learning portal, distant learning, testing, information system, CASE-technology.*

***ACM Classification Keywords**: D.2 Software Engineering: D.2.2 Design Tools and Techniques – Computer-aided software engineering (CASE); H.3 Information Storage and Retrieval: H.3.5 Online Information Services – Web-based services; K.3 Computers and Education: K.3.1 Computer Uses in Education – Distance learning.*

Introduction

Because of a widespread occurrence of the Internet during the late decade its users became able to quickly and easily access the desired information. It is well known that demand determines supply. That's why with the growth of users' needs there appeared many internet resources. It also applies to educational resources. Many educational portals were created lately. They include, for example, the Federal Portal «Russian Education» (www.edu.ru) and the Russian Educational Portal (www.educentral.ru). Besides there are educational institutions' portals which are certainly playing now an important role in the educational system. It means that they provide not only the delivery of important on-line information for students and teachers of these institutions, but also abilities for outside users to be registered at the portal and to perform operations with its resources. It should be particularly meant that many universities provide opportunities for distant learning. This paper presents technology which allows to simplify and automate the creation of such portals.

Portal, the Concept and Functions

In order to increase the efficiency of educational institutions it is necessary to implement a certain technological solution for integrating information in the educational field and creating an adaptable environment which would provide opportunities for remote workplaces. These workplaces are intended to allow employees, teachers and students not only to get access to the information resources, but also to perform typical operations concerned with their day-to-day activity. Such solutions are usually implemented in Web-portals which provide unified access to both public and nonpublic organizations' resources through the Internet.

A portal provides *1) an integration of applications and data and users access to them through the unified interface in accordance with the rules prescribed for them; 2) a workplace personalization; 3) publishing, distributing and presentation of information; 4) data categorization, exhaustive and relevant information search; 5) business-processes' management and collective work; 6) feedback and development*.

One of the main functions of a portal which distinguishes it from a "traditional" Web-site is an integration of data, being received from different sources, and applications, automating typical operations performed by users through the Web-interface. Another non-trivial task is a feedback support which is the key to the portal's development and its adaptation to user needs and changing conditions. Solving these tasks implies implementing appropriate mechanisms which would provide, in particular, data restructuring and user interface adjustment, functionality enhancements and the possibility for integration with external systems. This paper describes the technology suggested by authors for the development of portals satisfying mentioned requirements.

Standards and Technologies

CASE-technology METAS

The software described in this paper is based on CASE-technology METAS which provides information systems (IS) developers with means for data restructuring, user interface generation and adjustment, attaching components created by third-party developers, reporting, business-processes and security management [1].

Systems created with METAS are based on interpretation of multilevel metadata which describe the data domain model (this includes data allocated at the database of IS, their structure and operations concerned with their processing). The logical level of the data model allows users to manipulate data in terms of data domain entities. The logical level is reflected on the physical level which defines data presentation in terms of database (DB) tables of the system. Restructuring components provide the ability to dynamically change data model of IS DB. User interface (the main window of application and the object explorer in the form of a tree designed to navigate through data domain objects in accordance with existing relations between them, data input-edit forms) is generated automatically from the defined data model. Moreover, there is a capability to adjust interface to user needs. Users of the system can work with IS database (DB) through both Windows- and Web-interface. Reporting components offer a capability to create and execute database queries and develop document templates generated by the system. One can also place documents obtained from different sources in different formats to the DB. The functionality of IS can be enhanced also by means of attaching in the process of running the system new components (for example controls or business-operations) which can be designed by third-party developers and by means of business-processes management.

IMS Q&TI

Knowledge control is one of the most important components of every system used to support educational process. Testing is a traditional approach to control the knowledge of students. As a result of the wide and long-term application of tests there are many materials today which can be and must be reused or transferred from one system to another. That's why we need to standardize the way to represent and store tests for maintaining resources interoperability and their compatibility with different testing systems.

The most popular test standard today is IMS Q&TI [2, 3]. This standard is accepted by all major testing system vendors. Almost all testing systems use this specification to exchange the test material and some of them use it to store tests. In Russia this standard is accepted as the main standard for test storing. The main features of it are the separation of material's description and representation, the support for test hierarchies, hints, different ways for processing answers, an enhanced set of test settings and a lot of question types.

The Architecture of Information Systems Based on Technology METAS

The technology METAS is aimed at the creation of distributed information systems which integrate geographically remote subsystems (or domains). These subsystems can interact in different modes by using the BizTalk Server technology. Every domain is represented by application which has client-server architecture and works with a local database. Remote users can get access to IS resources through a Web-interface. A general architecture of a subsystem and a pattern of user access to IS resources through a Web-interface are shown on Fig. 1.

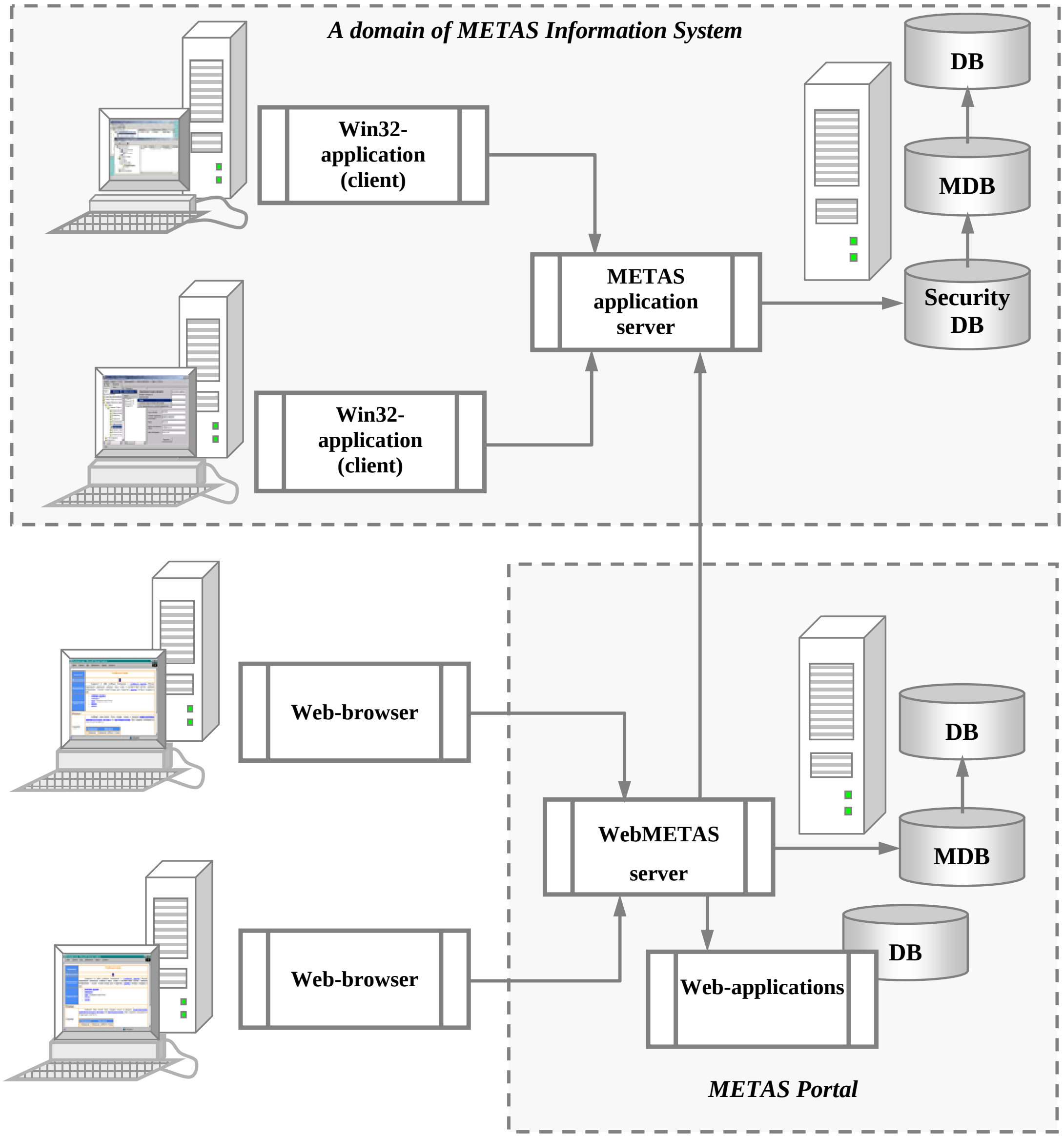

Figure 1. A general architecture of interaction between METAS components

The technology METAS enables to create applications which allow local users to work with them through Windows-interface and remote users through Web-interface. User interface is generated automatically from information stored in metadata database (MDB) of the information system. One of the METAS features is the ability to connect to different database management systems (DBMS) for which ODBC drivers are provided. The database access is carried out through the ADO.NET technology. An application specially developed in compliance with the BizTalk Server technology requirements allows to organize interaction and data transfer between subsystems (or domains) in different modes and using different protocols. In the process of subsystems' integration the METAS technology allows not only to transfer DB data but also to replicate all changes made to MDB. This provides the possibility to distribute changes in data structure, user interface settings and functionality enhancements. These capabilities are implemented in the replication component of CASE-system METAS.

The designed architecture allows to integrate stand-alone running applications which have different purposes and to combine models developed for different data domains. These features serve as the framework for creating integrated information systems for educational institutions consolidating applications designed for the management of these institutions and learning process [4].

EduMETAS, a Tool for Automating the Process of Creating Distant Learning Portals

A portal usually includes a forum, news, document publishing, links, search, authentication and other modules. Multilevel data model enables users to equally design and generate all these modules in METAS-system. On a logical level the data domain is described as a set of entities which represent data domain objects, their attributes and relations between objects. All operations are performed exactly with objects of this level. So, the logical level allows abstracting from the structures and peculiarities of data storage in relational database tables. A presentation level is used for describing the interface of a user which works with a system implemented on a METAS-technology basis.

The software reviewed in this paper operates via interaction with the logical level of the METAS data model. The administrator of a portal can design the structure of it and its modules by means of restructuring components which are developed as a Windows-application. This structure is used as a framework for generating the portal. To solve this task we developed a component responsible for Web-presentation of information through objects' data input-edit forms. We also implemented components which secure the portal's data, allow to create and maintain portal user accounts, user groups and their permissions for accessing and modifying portal resources.

By using METAS's universal adjustment components we were able to create Web-application providing flexible means for dynamic restructuring of the portal and extending its functionality [5, 6]. This allows the portal to be adjusted to satisfy the changing conditions and user needs during its operation (without having to recompile it).

The portal's information resources are described in terms of data domain objects. Every object is associated with some type. An object type is specified by its attribute set, Web-presentation and user permissions for accessing objects of that type.

A user can create, modify or delete objects in compliance with his/her permissions. Permissions can be set both for individual users and user groups.

A user can navigate resources of the portal through the site map in the form of portal objects' tree view the structure of which is determined by metadata. Every site's section has a corresponding object in the tree view. When a user selects tree's node (or some section of the portal) a Web edit form or document is displayed. Editing information about the objects implies modifying its attributes. Every attribute is shown on the Web edit form through some control. For each attribute the layout and the behavior of the corresponding control is set. Manipulation with object's attributes is executed in a common way. On the other side one can extend the portal's functionality through enriching the set of available attribute types and corresponding controls for manipulating them. That's one of the ways for providing the necessary flexibility. For example, it becomes possible to work with data which have sophisticated organization (files of different types, hierarchical classifiers, etc.). For increasing efficiency of user's work a web form can contain controls associated with referenced objects' attributes or lookups on parent and child objects.

Extending the functionality of a portal is possible not only by introducing new types of controls. It can also be made by supporting business-operations. Business-operation is a non-typical operation performed on objects (operations for creating, modifying and deleting objects are not considered as business-operations) which is specific for a particular application and its data domain. Such operations can be implemented by third-party developers and attached to portal during its operation.

So, a developer of a portal is provided with all means for designing its structure and content and performing necessary operation on it.

It was mentioned that a subsystem for knowledge remote control and user testing is a mandatory part of distant learning portals. This subsystem is implemented by attaching the application which implements business-operations on objects which represent tests and users of the portal (experts, students and administrators which will be described further in this paper).

Now we will focus on the testing subsystem of the distant learning portal EduMETAS.

Testing Subsystem’s Architecture

The testing subsystem must work with at least three user categories:

1. “Expert” – a user who creates tests for controlling student’s knowledge of the data domain through test building components. Expert should be a person with teaching and practical experience in the data domain of the test being created (a teacher).

2. “Student” – a user (usually a student) who examines his knowledge in a particular data domain. Initial requirements for a student include minimal skills of working with a computer through Windows- or Web-interface.

3. “Administrator” – a person who maintains and sets up all parts of the distant testing subsystem and is responsible for its operability (a developer).

For every user category appropriate permissions are defined. Obviously an “expert” should be able to create tests, modify and delete them, to set assessment criteria, to view and analyze test results. A “student” should not have access to operations of modifying and viewing the test. He (she) must be able only to pass the test by choosing the test’s type and the particular test and view the results. “Administrator” should have access to all data and their database descriptions for maintaining the system in the operable condition.

Besides, users from different categories should have access to different functions: a “student” needs only to view test’s question in a reasonable view and to select or enter answers; an “expert” needs a tool to create tests and set up the appearance of their questions, to build database queries to put out the results of the testing; an “administrator” should have tools for controlling users’ work with the system.

According to IMS Q&TI standard a test has a hierarchical structure. Every test consists of sections, every section contains questions and questions include information depending on the question’s type. A test can contain the following question types: “choice”, “multiple choice”, “string response” (when a user should enter the text), “”numeric response” (when a user should enter some number), “match objects” (when objects from several lists have to be grouped together) , “fill-in-blank” (when a user should fill numbers or text in blanks in the question’s text). In addition to standard question types implemented in the system new question and task types can be created via restructuring and business-operation attaching tools.

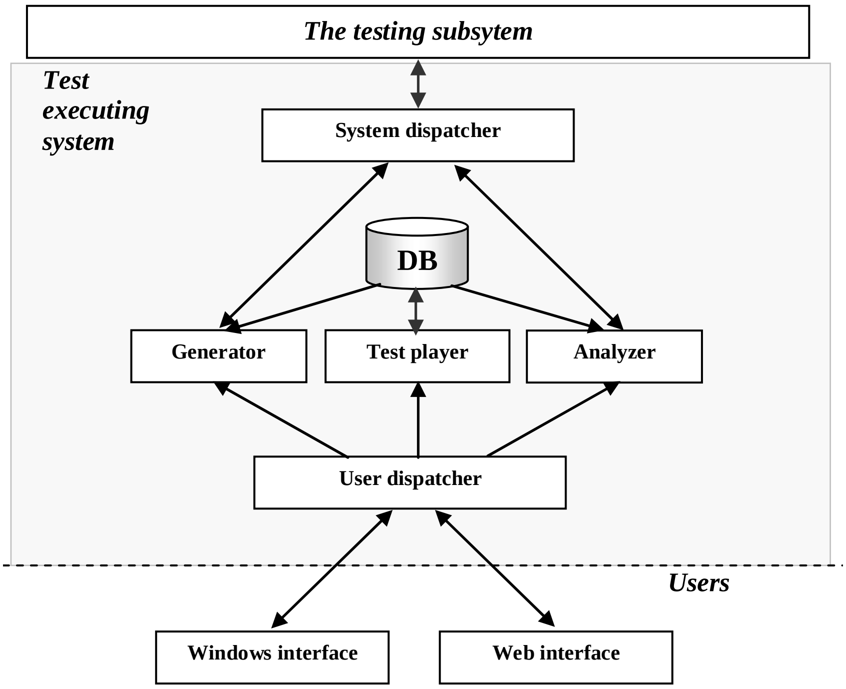

Figure 2. A general architecture of the test executing system

The testing component is implemented as a system-independent module [7] which performs test’s running and results’ check. This module has several functions such as preparing the data for testing, running the test, results’ check, user interface and interface with the testing subsystem to which it is connected.

The test executing system includes several modules. Its architecture is shown on Fig. 2.

The test executing system consists of five main modules: system dispatcher, user dispatcher, generator, test player and analyzer. It uses local database and that provides the possibility to integrate with various testing systems which support IMS Q&TI standard. The interaction is performed on a basis of

XML.

Now we'll take a closer look at every of this modules.

The main purpose of *the user dispatcher* is to redirect user requests to other modules. A user interacts through Web- or Windows-interface exactly with this module by sending requests. These requests should be processed by different system's modules. That's why when the user dispatcher receives a request for accessing data or executing some operation it analyzes the request and redirects it to the appropriate system's module which implements the requested function. The dispatcher maintains the user's session and when it receives the response from this module it sends the result of the operation to the user.

The system dispatcher is developed to redirect requests from the testing system into which the test player is integrates to other modules. Depending on request's parameters it redirects the request to the generator or the analyzer. When it receives request's result from the test executing system the system dispatcher returns it as a response to the testing system. When the test's result is requested the system dispatcher redirects it to the analyzer. If the data are passed for storing to the DB (in XML format), the request is redirected to the generator.

The main functions of *the generator* are to make a list of users in the database, to make a list and descriptions of tests, to set an order of questions for testing a particular user. Requests for these functions are received by the generator from the user or the system dispatcher. A command from the system dispatcher should have an XML-string as a parameter whish must contain a list of all users who can pass tests, a list of all tests with questions in the IMS Q&TI 1.01 standard and the mapping of users and tests (which tests are accessible for each particular user). If there is no such mapping, it is considered that all tests are available for all users.

Commands from the user dispatcher can contain following sets of parameters:

- no parameters (in that case the response contains a list of users from the users table of the database);
- user ID (in that case the response contains a list of tests available for the user with the specified ID)
- user ID and test ID (in that case it is considered that the user chose a test to pass and a sequence of question is generated for that user depending on the type of constructing a list of questions for the selected test).

The main purpose of the test player is to store to the database the user's answer to the question and to return a new question. Parameters of the request to this module must include the user ID and the question ID. One more parameter is optional. This is the user's answer to the question (in XML format). If the third parameter is missing it is considered that this is a request for a new question. In that case the text of the question is retrieved from the questions table and the description of the user is retrieved from the users table. This description is used to perform some verifications such as the time of passing the test (whether the user exceeded the time limit for the test), the sequence of user responses for the question (whether the user can answer the questions in an arbitrary order), etc. If all verifications are successful the text of the new question (in XML format) is returned as the answer to the dispatcher's request. Otherwise the cause of rejecting the request is returned. If the third parameter is specified (so it is the request for storing the user's answer) then the user's description is extracted from the users table to performs such verifications as the possibility to answer the question again (i.e. correct the previously submitted answer), etc. If all verifications are successful then the value passed as the third parameter is stored in the database. The returned result contains an empty XML document if the request is completely accomplished or an XML describing the cause of rejecting the request if storing the answer failed.

The analyzer has two main functions. They are an examination of the test results and an output of these results. The request for the examination of the results is sent by the user's dispatcher. Request parameters must include the user ID which is used to select answers and check-up the results for the chosen test. Results for each question are stored in the database table. The result of fulfilling the request are the number of correct answers and/or the list of "question number - result of question's checking-up" pairs.

At present Microsoft Access or Microsoft SQL Server (or just Express Edition) can be used as the database for testing subsystem. The results can be stored in Microsoft Office documents.

Conclusion

The paper presents an approach to developing educational portals by means of CASE-technology METAS which is used to create dynamically adjustable distributed information systems and by means of testing components developed by ANO "Institute of computing" and Perm State University's Computer Science Department's employees. These components can be used to create portals which implement all functionality necessary for different user categories. It is possible thanks to system's dynamic adaptation and functionality extension capabilities. The technology allows working with different relational database management systems and integrating heterogeneous information systems. Besides users can both work using Windows applications or gain access to the portal's resources through the Web interface of the system.

Bibliography

[1] Лядова Л.Н., Рыжков С.А. CASE-технология METAS // Математика программных систем: Межвуз. сб. науч. тр. / Перм. ун-т. Пермь, 2003. С. 4-19.

[2] IMS (IMS Global Learning Consortium) – American Project [Web Resource]: http://www.imsproject.org.

[3] IMS Q&TI Specification [Web Resource]: http://www.imsproject.org/question.

[4] Лядова Л.Н. Архитектура информационной системы «Образование Пермской области» // Математика программных систем: Межвуз. сб. науч. тр. / Перм. ун-т. Пермь, 2002. С. 25-35.

[5] Рыжкова Е.А., Хлызов А.В. Реализация удаленного доступа к ресурсам информационной системы «Образование Пермской области» // Информатика в школе: Тезисы докладов X юбилейной областной научно-методической конференции 9-10 января 2006 г. «Рождественские чтения» / Перм. ун-т. Пермь, 2006. С.86-88.

[6] Хлызов А.В. Разработка средств создания порталов с использованием многоуровневых метаданных // Технологии Microsoft в теории и практике программирования: Тез. докл. конференции-конкурса / Нижегородский ун-т. Нижний Новгород, 2006. С.310-313.

[7] Урезалов А.В. Разработка средств тестового контроля на основе CASE-технологии METAS // Технологии Microsoft в теории и практике программирования: Тез. докл. Конференции-конкурса / Нижегородский ун-т. Нижний Новгород, 2006. С.303-306.

Authors' Information

Ludmila Lyadova *– Head of Department, Computer Science Department, Perm State University, Bukirev St., 15, Perm, 614990, Russia; e-mail: cs-psu@mail.ru.*

Alexey Urezalov *– University Undergraduate, Computer Science Department, Perm State University, Bukirev St., 15, Perm, 614990, Russia; e-mail: uralkama@mail.ru.*

Andrey Khlyzov *– University Undergraduate, Computer Science Department, Perm State University, Bukirev St., 15, Perm, 614990, Russia; e-mail: andrew_khlyzov@mail.ru.*