Deciding on the Architecture of the Concept Map Based Knowledge Assessment System

Romans Lukashenko, Marks Vilkelis, Alla Anohina

Abstract: *The paper is dedicated to the problem of the selection of an optimal client-server architecture for the implementation of the Web-based learning system. Constituent parts, advantages and disadvantages of nowadays prevalent client-server architectures, namely two-tier, three-tier and N-tier architectures, are described. Experience of the authors in the development of the concept map based knowledge assessment system using two- and three-tier client-server architectures is discussed as well.*

Key words: *Client-Server Architecture, Web-based educational application, J2EE platform.*

INTRODUCTION

The Internet and World Wide Web provide unprecedented opportunities for carrying out online learning and distance education [5] in the form of Web-based educational applications implemented on the basis of the client-server architecture. Nowadays there are several types of client-server architectures, namely application can be built as two-, three- and N-tier systems. The selection of the type of the architecture is affected by such parameters as number of simultaneous connections, network traffic load, security level, overall system performance, maintenance costs, development complexity, upgradeability and scalability options. Thus, the problem of the selection of the most suitable type of the architecture for a Web-based application is crucial. The detailed investigation and comparison of available architectures should be done before deciding on a particular architecture for future implementation.

The remainder of this paper is organized as follows. In the next section two-, three- and N-tier architectures are described and compared. Then the concept map based knowledge assessment system implemented as a Web-based educational application is presented. After that our experience in the development of two- and three-tier architectures is discussed. Finally, some conclusions are given.

TWO-TIER AND THREE-TIER CLIENT-SERVER ARCHITECTURES

A client-server application, as a rule, is composed of two types of nodes: a client and a server. The client is a software program which sends data requests to one or more servers. In turn, the server is an application which accepts client's requests, processes them and returns the requested information to the client. Nowadays, the interaction between the client and the server is often supported by an additional node, namely middleware [11].

A client-server application consists of different layers (tiers) which refer to the number of executable components into which the application is partitioned [11]. The tiers are entirely logical in nature. Their physical implementation may vary considerably: everything compiled into one executable file, a single tier spread across multiple statically- or dynamically-linked libraries, tiers distributed amongst separate networked computers [12].

Generally, there are several tiers that represent different logic in an application [10]:

- *A presentation tier* shows how information is presented to clients, for example, how web pages interact with clients.
- *A business tier* is composed of a collection of objects and methods that differ from business to business. This tier is related to the aspect how to represent business logic which should be performed in a certain way in the system.
- *A data tier* ensures the persistence and security of data. It holds application's business data and automates the storing, searching, and retrieving of them. The data tier usually comprises one or several database management systems.

Typically there are three types of tiered architectures: two-tier, three-tier and N-tier architecture. In the two-tier client-server architecture (Fig. 1) the functionality of an application is partitioned into two tiers. One of them implements both the presentation logic and the business logic. This tier executes on a client's machine and requests data from the second tier, which usually resides on the server's machine where the application's data are stored [11, 12, 13]. The last mentioned tier implements the data access logic.

In the three-tier client-server architecture (Fig. 2) the application's code is partitioned into three tiers: the presentation tier, the business tier and the data tier. In the deployment, or physical partitioning, of the application the business logic is separated from both the presentation logic and the data access logic. The business logic is deployed on its own

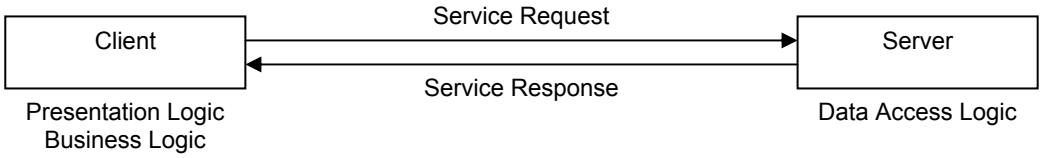

Fig. 1. The two-tier architecture (adapted from [13])

server (middle tier server) or on the same server as the database. The middle tier adds queuing, scheduling and prioritization of work in progress to this architecture [11, 12, 13]. It also provides the additional infrastructure for the better Web security because the server controls access and resources to guarantee that only those clients who have the appropriate permissions may access and change data. This tier uses access control lists to protect services, queues, or events from an unauthorized access. The mentioned security mechanisms are not available in the two-tier architecture [3].

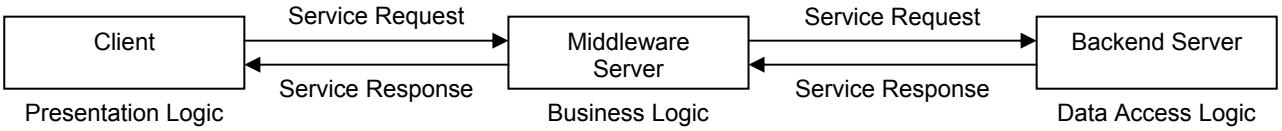

Fig. 2. The three-tier architecture (adapted from [13])

N-tier systems are those systems which have more than three tiers in their architecture. This type of the architecture theoretically provides even greater performance capabilities and maintainability. However, a n-tier system is inefficient in practice in most cases, expensive and more complex [10, 11].

Table 1 summarizes advantages and drawbacks of the mentioned types of client-server architectures [3, 8, 9, 11, 12, 13] which show that the three-tier application provides greater opportunities for scalability and upgradeability than the two-tier system. The three-tier architecture also offers affordable manageability combined with good performance and security. Thus the development of client-server applications, inter alia Web-based educational systems should be oriented towards the three-tier architecture [11, 12].

In the next sections we will share our experience in the development of a concept map based knowledge assessment system using the two-tier and the three-tier client-server architecture.

A SHORT DESCRIPTION OF THE CONCEPT MAP BASED KNOWLEDGE ASSESSMENT SYSTEM

The last three years we have been developing the concept map based knowledge assessment system [1, 2]. It allows the teacher to assess learner's knowledge regularly, that is, at each stage of the study course, and to use assessment results for the analysis and the improvement of learning content and teaching methods. At the same time the

learner can use the system for knowledge self-assessment in order to control and to keep track of his/her own learning progress.

Table 1: The comparison of two-tier and three-tier client-server architectures

Architecture	Advantages	Disadvantages
Two-tiered	• Lots of high level development tools, easy to develop • Less network traffic between the client and the server	• The client is too heavy (thick, fat client), i.e., too many software on the user machine • Number of users is limited by 100 users connected simultaneously • Difficult and expensive to upgrade: software copies on each client should be updated
Three-tiered	• A lot less software on the client (thin, lite client) • Easier and cheaper to maintain and upgrade: only one software copy on the server should be updated • Increased performance, flexibility, maintainability, reusability, and scalability – each tier can be scaled and upgraded independently • Much more secure	• Complex structure, the development environment is more difficult to use

The developed system has three discriminative features in comparison with other systems based on concept maps. Firstly, the system uses the new algorithm that compares the teacher's and the learner's concept maps and is sensitive to the arrangement and coherence of concepts. Secondly, possibility to change the degree of task difficulty is included and allows more accurate assessment of the knowledge level of a learner. Thirdly, the system supports systematic knowledge assessment and allows the teacher to extend the initially created concept map for the new stage of assessment.

The system has been experimentally evaluated in 7 learning courses. Students positively evaluated the chosen approach to the knowledge assessment, as well as the functionality and the user interface of the system.

THE TWO-TIER ARCHITECTURE OF THE SYSTEM

The concept map based knowledge assessment system initially was built as a Web-based client-server application on the basis of the J2EE platform by using the following technologies: Borland JBuilder 9.0., JGraph, PostgreSQL DBMS 8.1.3. and JDBC drivers. The platform was chosen taking into account the following considerations. Nowadays the most well-known approaches for the implementation of client-server applications are J2EE and .NET. J2EE is available on the market already for many years while .NET appeared only several years ago. Discussing costs needed for the development of applications it is necessary to stress that J2EE is free, as well as there are free application servers. In order to use .NET we need to buy a Visual Studio .NET IDE (Integrated Development Environment) and IIS Web server. Further, J2EE is more opened because anyone can implement a J2EE application server that meets the specification. In turn, .NET as a product of the Microsoft Corporation depends on other Microsoft products. Moreover J2EE is more widely used technology for the realization of client-server systems [13].

During the development of the system the main attention was devoted to its overall conception and functional capabilities and less one to the architecture of the system and protection of data. Figure 3 displays the architecture of the system from the point of view of the used technologies. The learner accesses the application by using any browser, for example, Internet Explorer or Opera. The client application is downloaded from the server via Web Start and installed on the user's machine (the client side). When the learner

interacts with the system data exchange between the database on the server side and the client application is carrying out. The application connects to the server directly and communicates with it using the JDBC driver and SQL or PL/SQL commands. Thus, the data tier is implemented by the server while the client side implements the presentation and the business logic tiers. It is the typical two-tier architecture.

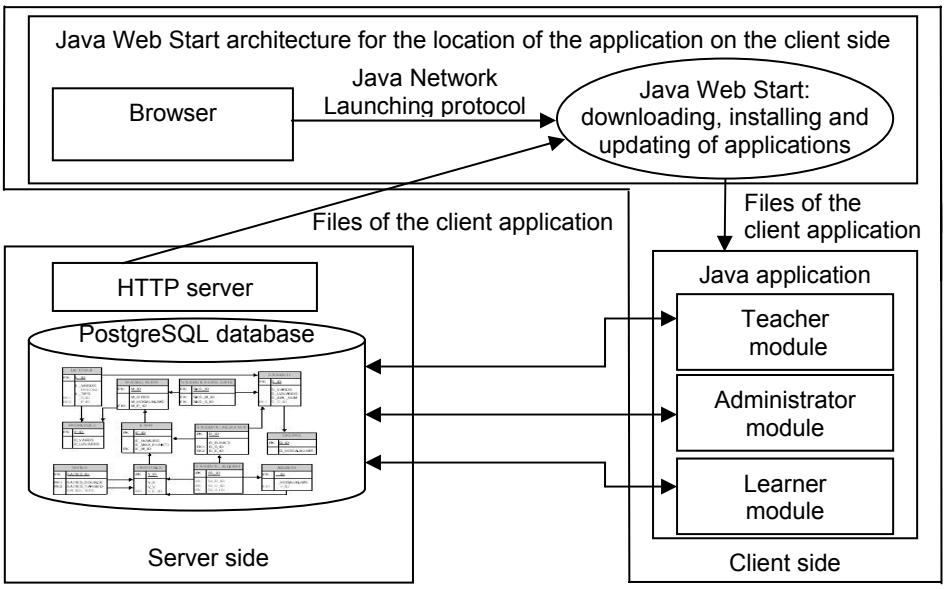

Fig.3. The two-tier architecture of the concept map based knowledge assessment system

The following problems have been revealed by using the described architecture. Firstly, the application is not secure enough because the database is opened for any outside connections. To get access to the database, it is necessary to input correct address, user name and password, but it is not a big problem, because all application code, including database connection data, is accessible on the client side after launching Web Start. Secondly, SQL queries are not much flexible and readable, thus, it is not easy to refactor some logic or data access methods.

The problems motivated us to reconsider and to modify the architecture of the system by making the transition from the two-tier architecture to the three-tier architecture which is described in the next section.

THE THREE-TIER ARCHITECTURE OF THE SYSTEM

The new (three-tier) architecture of the concept map based knowledge assessment system is displayed in Figure 4. It has three conceptual elements: the database server, the client application and the application server. The last one is a new component in comparison with the previously described old architecture of the system. The architecture is implemented by using the following technologies: Eclipse 3.2, Apache Tomcat 6.0, PostgreSQL DBMS 8.1.3, JDBC drivers, Hibernate, VLDocking, JGoodies, and JGraph.

Apache Tomcat is chosen as an application server. It is a container of servlets. A servlet is a small Java interface, which runs within a Web server. Servlets receive clients' requests and respond to them, usually across HyperText Transfer Protocol. There is a basic implementation of this interface (for example HttpServlet), but we can extend it by creating our own event handlers and data transformation for a concrete business logic.

The working principles of the architecture are based on the Model-View-Controller (MVC) pattern [7]. The user still downloads the client application via Web Start which is supported by Apache Tomcat as well. The application on the client side implements both conceptual layers: views (buttons, text fields, panels and other visual components of the

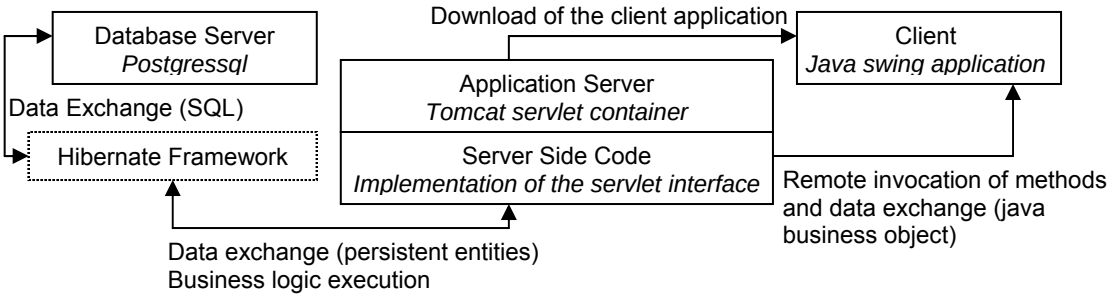

Fig. 4. The three-tier architecture of the concept map based knowledge assessment system

user interface) and models, which manage views by relating them to data. All user's actions in the user interface (such as pressing of buttons, entering text in a text field, etc.) are events. The client application handles different events and calls services of the application server. Called services are executed remotely on the application server, which acts as a controller. The client knows only signatures of the accessible server methods. Implementations of methods are stored on the application server and the client side does not know anything about them. So direct connections between the database and the client application are eliminated, what prevents the system from an unauthorized access.

The application server receives remote calls from the client and redirects them to the appropriate servlet. The information about a servlet is included in the remote call. The servlet handles a call and launches the appropriate method needed for the communication with the database or for the execution of business logic. The application server does not use SQL queries to perform data manipulations. Instead of that, we have Java object oriented framework, namely Hibernate. Hibernate is a high performance object/relational persistence and query framework. It allows programmers to develop persistent classes following object-oriented paradigm, including associations, inheritance, polymorphism, composition and collections. Hibernate provides opportunities to create queries by using HQL (SQL extension), native SQL, or object-oriented criteria [4]. The server asks Hibernate to save, update or delete some objects. Hibernate generates appropriate SQL scripts and interacts with the database.

The described three-tiered architecture allows us to solve the problems identified in the previous section. The application is secure enough because the database server is opened for connections only from one computer, in our case, from the application server Apache Tomcat. No one from outside can get access to data. Refactoring of logic or data access methods is easier due to the usage of Hibernate.

CONCLUSIONS AND FUTURE WORK

The paper focuses on the transition from the two-tier architecture of the concept map based knowledge assessment system to the three-tiered application. The transition is motivated by the identification of advantages and drawbacks of both types of the mentioned architectures, as well as by the revealed problem of the two-tier architecture, namely unprotected data. The three-tier architecture solves the problem of data security by introducing the application server between the client application and the database server. The application server works as a controller by granting data access rights only to those clients who have the appropriate permissions.

In the future we are planning to continue the improvement of the architecture of the concept map based knowledge assessment system. It can be related to the integration of a security server, for example, Lightweight Directory Access Protocol [6]. At the moment all information about users, their roles and privileges are stored in the database. It is not a good approach, because security belongs to one logical layer, but data management and database to another.

REFERENCES

[1] Anohina, A., J. Grundspenkis. Prototype of Multiagent Knowledge Assessment System for Support of Process Oriented Learning. Proceedings of the 7th International Baltic Conference on Databases and Information Systems, July 3-6, 2006, Vilnius, Lithuania, pp. 211-219.

[2] Anohina, A., D. Pozdnakovs, J. Grundspenkis. Changing the Degree of Task Difficulty in Concept Map Based Assessment System. Proceedings of the IADIS International Conference "e-Learning 2007", July 6-8, 2007, Lisbon, Portugal, pp. 443-450.

[3] Client Server Architecture, Key Concepts. DISCOM. Available online at: http://mazsola.iit.uni-miskolc.hu/tempus/discom/doc/db/tema02.pdf (Last visited: 25.03.08)

[4] Introduction to Hibernate. Red Hat Middleware, LLC. Available online at: http://www.hibernate.org/hib_docs/reference/en/html/tutorial.html (Last visited: 31.03.08)

[5] Liaw, S. S., H. M. Huang. Exploring Web-based Learning for Distance Education: Client-Server Architecture Approach. International Hawaii Education Conference, Hawaii, USA, January 07-10, 2003. Available online at: http://www.hiceducation.org/Edu_Proceedings/Shu-sheng%20Liaw.pdf (Last visited: 25.03.08)

[6] Marshall, B. Introduction to LDAP. Available online at: http://quark.humbug.org.au/publications/ldap/ldap_tut.html (Last visited: 31.03.08)

[7] Model-View-Controller Pattern. eNode, Inc., 2002. Available online at: http://www.enode.com/x/markup/tutorial/mvc.html (Last visited: 31.03.08)

[8] Schussel, G. A Generalized Way Of Thinking About N-Tier Client/Server Architectures. DCI's Database and Client/Server World, March, 2005. Available online at: http://www.catstravel.com/files/GB7620050322_3-TIER.pdf (Last visited: 25.03.08)

[9] Schussel, G. Client/Server: Past, Present and Future. Available online at: http://www.dciexpo.com/geos/dbsejava.htm (Last visited: 25.03.08)

[10] Sun, H. Web Client-Server Architecture To Support Advanced Text Search. Master thesis, August, 2005. Available online at: http://www.dcs.shef.ac.uk/intranet/teaching/projects/archive/msc2005/pdf/m4hs.pdf (Last visited: 25.03.08)

[11] Three-tier Client-server Architecture: A Targeted Applications Environment. Office of the Governor, State of Utah, December, 1998. Available online at: http://www.governor.state.ut.us/CIO/Docs/ClientServerSF.pdf (Last visited: 25.03.08)

[12] Two and Three Tier Architecture. Rai Foundation Colleges. Available online at: http://www.rocw.raifoundation.org/computing/BTech-IT/DataBaseManageSys-3.202/lecture-notes/Lecture-20.pdf (Last visited: 17.03.08)

[13] Woo, J. The Comparison of J2EE and .NET for e-Business. Proceedings of the 2005 International Conference on E-Business, Enterprise Information Systems, E-Government, and Outsourcing, Las Vegas, Nevada, USA, June 20-23, 2005. Available online at: http://www.calstatela.edu/faculty/jwoo5/publications/hipic-techreport-102912003.pdf (Last visited: 25.03.08)

ABOUT THE AUTHORS

Assistant, Romans Lukashenko, M.sc.ing., Faculty of Computer Science and Information Technology, Riga Technical University, Phone: +371 67 089 529, E-mail: lrexpress@inbox.lv

Student, Marks Vilkelis, B.sc.ing., Faculty of Computer Science and Information Technology, Riga Technical University, Phone: +371 67 089 529, E-mail: markvilkel@inbox.lv

Lecturer, Alla Anohina, Dr.sc.ing., Faculty of Computer Science and Information Technology, Riga Technical University, Phone: +371 67 089 595, E-mail: alla.anohina@rtu.lv