Advances in Databases and Information Systems

13th East-European Conference, ADBIS 2009, Associated Workshops and Doctoral Consortium, Riga, Latvia, September 7-10, 2009

Local Proceedings

Edited by

Janis Grundspenkis
Riga Technical University, Latvia

Marite Kirikova
Riga Technical University, Latvia

Yannis Manolopoulos
Aristotle University, Thessaloniki, Greece

Tadeusz Morzy
Poznan University of Technology, Poland

Leonids Novickis
Riga Technical University, Latvia

Gottfried Vossen
University of Muenster, Germany

Riga Technical University
2009

Technical Evolution of the Concept Map Based Intelligent Knowledge Assessment System

Marks Vilkelis[1], Romans Lukasenko[1], Alla Anohina[1]

[1] Department of Systems Theory and Design, Riga Technical University, Kalku Str. 1, LV-1658 Riga, Latvia
MarkVilkel@inbox.lv, LREXPress@inbox.lv, All.Anohina@rtu.lv

Abstract. The paper describes evolution of the concept map based knowledge assessment system developed at the Department of Systems Theory and Design of Riga Technical University. Three generations of the system are presented focusing on technical aspects of system's implementation. The first version of the system was implemented as a two-tier client-server application. In the second version all functionality was moved onto more scalable and secured three-tier architecture. Refactoring of GUI engine was made in the last prototype to get the client side application running on the multi-threading environment for faster and smoother displaying of UI components.

Keywords: Knowledge assessment, concept map, client-server architecture, graphical user interface, Swing, AWT.

1 Introduction

Knowledge assessment is an integral part of the learning process. However, it is a very time and effort consuming activity in the traditional learning process, because it demands from the teacher to prepare assessment tasks or questions, to conduct assessment activities, to check and evaluate students' works, to provide feedback. This is the reason for the development of computer-assisted assessment systems. The mentioned systems are used, on the one hand, to detect students' knowledge and skills, but, on the other hand, to regulate teaching and learning process on the basis of informative and tutoring feedback generated automatically by the system.

This paper presents the concept map based knowledge assessment system (KAS) developed by researchers from the Department of Systems Theory and Design of the Faculty of Computer Science and Information Technology of Riga Technical University. The system is based on concept maps representing knowledge in form of a graph which nodes correspond to concepts in a domain, but arcs indicate relationships between concepts. The developed system supports both assessment made by a teacher and students' self-assessment in order to keep track of person's learning progress.

The discriminative feature of the KAS is support of process oriented learning, in which a teacher divides a study course into several stages and carries out knowledge assessment at the end of each stage. Thus, systematic assessment is provided that, in its turn, allows not only assessing of a current knowledge level of each individual

student, but also changing of teaching methods and learning content timely in order to achieve a desirable knowledge level and to promote qualitative teaching and learning process.

The mentioned system has already reached the certain level of maturity and has been used successfully in practice. Therefore, this paper is focused on its evolution considering technical aspects of system's implementation. The main aim of the paper is to recap authors' experience in developing of an e-assessment system and to point out factors which should be taken into account by other developers when implementing highly interactive web-based educational systems.

The remainder of the paper is organized as follows. In the next section a brief description of working principles of KAS is given. Section 3 provides the description of all three versions of the system and presents a comparison of all prototypes. Finally, the conclusions are given.

2 A Short Description of the KAS

The main reason which motivated us to develop a new computer-based KAS is to ensure systematic knowledge assessment. Systematic assessment allows a student to tracks his learning progress and allows a teacher to make necessary corrections into learning content and teaching methods thus providing more qualitative teaching.

As it was mentioned in Introduction concept maps are used as an assessment tool in the KAS. They represent knowledge in form of a graph which nodes correspond to concepts in a domain, but arcs indicate relationships between concepts. Concept mapping stimulates students to articulate and externalize their actual state of knowledge. Concept maps allow assessment of the higher order cognitive levels according to Bloom's taxonomy [1] and support the checking of the students' understanding of interconnectedness of concepts mastered during a study course, instead of a degree of memorization of separate facts.

The developed KAS allows a teacher to assess student's knowledge regularly, that is, at each stage of a study course, and to use assessment results for the analysis and the improvement of learning content and teaching methods. At the same time a student can use the system for knowledge self-assessment in order to control and to keep track of his/her own learning progress.

The system is used in the following way. The teacher creates one or more concept maps for a study course. The process of the creation of a concept map consists from the specification of relevant concepts and relationships among them. Teacher's created concept maps serve as a standard against which students' concept maps are compared. During knowledge assessment a student solves a concept-map based task using initially given concepts and linking phrases. Maps created by students show how well they understand the learning material. After the student has submitted his/her solution, the system compares the concept maps of the student and the teacher, identify student errors, calculates the student's score and generates feedback which is delivered back to the student.

The developed system has three discriminative features in comparison with other systems based on concept maps. Firstly, the system uses a new algorithm that

compares the teacher's and the student'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 student. Thirdly, the system supports systematic knowledge assessment and allows the teacher to extend an initially created concept map for the new stage of assessment [2].

The system has been experimentally evaluated in 7 study courses with the participation of 256 students. Results of submitted questionnaires after the use of the KAS show that students positively evaluated the chosen approach to knowledge assessment, as well as the functionality and the user interface of the system.

The next section presents the history of the development of the KAS on the basis of evolution of the system from technical point of view. The description of evolution of functional aspects of the system can be found in [3]. It is necessary to point out that the fifth prototype of the system considering its functionality is under development at the moment. However, from the technical point of view it is possible to identify only three versions of the system: the first one was developed in 2005, the second one appeared in 2008 and the last version is being developed now.

3 Implementation of the KAS

The development of the KAS started in 2005 and four projects funded by the Latvian Ministry of Education and Science and Riga Technical University have been finished since that time and one more project is running at present.

3.1 The First Version of the System

The development of the first version mainly was focused on the implementation of the overall conception and basic functional capabilities of the system moving aside questions related to physical implementation and data security. The architecture of the system included three modules: the teacher module, the learner module and the administrator module. The following tools were chosen: Borland JBuilder 9.0, JGraph, PostgreSQL DBMS 8.0.3 and JDBC driver for PostgreSQL.

The two-tier client-server architecture was used (Fig. 1) in the implementation of the first version of the system. A student accesses the application using any browser. The client application is downloaded from the server via WebStart and installed on the user's machine (the client side). Data exchange between the database on the server side and the client application is carried out when the student interacts with the KAS. The application connects the server directly and communicates using the JDBC driver and SQL or PL/SQL commands.

During experimental evaluation of the system several serious bottlenecks were discovered in system's work. Firstly, the application was not secure enough because the database was open for any outside connections. To get access to the database, it was necessary to input a correct address, a user name and a password, and it was easy to do because all application code, including database connection data, was accessible on the client side after launching Web Start.

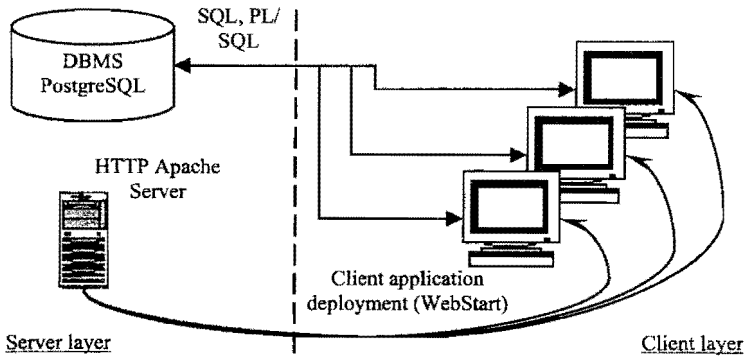

Fig. 1. The two-tier client-server architecture

Secondly, users experienced difficulties working with the system when the number of simultaneous connections was growing decreasing system's overall performance. The most dramatic decrease was observed when the number of simultaneous connections reached 50 users. In this situation some students faced denial of services.

The mentioned problems motivated a team of developers to reconsider and to modify the architecture of the system by performing transition from the two-tier architecture to the three-tier architecture which is described in the next section.

3.2 The Second Version of the System

The main goal of the implementation of new three-tier architecture was to make the system more secure providing protection of students' data and their results. The architecture shown in Fig. 2 has three conceptual elements: 1) a data layer, which is represented by Data Base Management System (DBMS) PostgreSQL; 2) an application logics layer, which is composed of two parts: the application server and the server side code running on it (a special persistence and query framework is used to communicate with the DBMS); and 3) the representation layer or graphical user interface (GUI) [4]. The new version of the KAS was implemented using the following technologies: Eclipse 3.2, Apache Tomcat 6.0, PostgreSQL DBMS 8.1.3, JDBC drivers, Hibernate, VLDocking, JGoodies and JGraph.

Experimental evaluation of the prototype showed that implemented three-tier architecture helped to solve the problems identified in the first version of the system. The application was secure enough because the database server was opened for connections only from one computer, in our case, from the application server Apache Tomcat. No one from outside could get access to data. In addition, system's performance was increased significantly and students did not experienced denial of services any more.

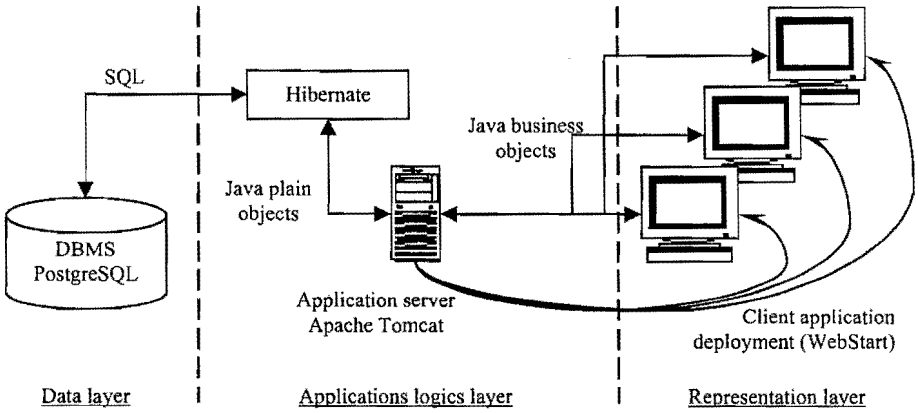

Fig. 2. The three-tier client-server architecture

Nevertheless, some issues regarding slow performance were still mentioned. Deeper investigation of the problem has shown that a GUI engine that is responsible for drawing and refreshing graphical objects on user's monitor is working slow and refactoring need to be done in order to make the engine work faster. Modifications implemented in the GUI engine are described in details in the next section.

3.3 The Third Version of the System

The greater part of changes and improvements in the previous versions of the system were related to the architecture and logics of the application server. From the point of view of the GUI engine the client side was minimally refactored in the second prototype comparing with the first one. So the main area of improvements in the third version of the system, apart from new functionality, is GUI core.

Let's briefly describe the most common problems related to GUI in client-server applications such as the KAS. As it was mentioned in [2] GUI of the developed system is based on Java open source graphical library Swing. Swing working principles obey the single-thread rule – once a Swing component has been realized, all code that might affect or depend on the state of that component should be executed in the single event-dispatching thread. That means, that all graphical components are being displayed and handling of their events occurs one by one in the single thread organized as a queue. Taking into account that the KAS is a client-server application, it is obviously that the client part needs to perform non-event-driven GUI work after the GUI is visible to users [5, 6]. There is one more important issue. The client GUI of the KAS must be updated as a result of non-AWT (Abstract Window Toolkit) events. For example, suppose that the client part can get requests from the application server code running on different machine [7]. These requests can come at any time, and they result in one of the server's methods being invoked in some unknown thread. How can that method update the GUI? By executing the GUI update code in the event-dispatching thread [8, 9, 10].

Let's examine the following scenario. A user opens a concept map editing window and presses a button "Load Concept Map". AWT event-dispatching thread generates an event related to pressing of the button and a corresponding remote service invocation request is sent to the server side. In the previous KAS versions GUI was not updated exactly after sending of such a request, because every action was executed in one GUI thread, so it was necessary to wait while the server performs a corresponding method and returns an "answer" back to the client. Only after that GUI could be refreshed using data returned from the server. In case of long server method invocation, GUI used to hang up.

In the third KAS version the client AWT event-dispatching thread does not wait any more for a server response, but it continues its execution. GUI working principles described so far is shown in Fig. 3.

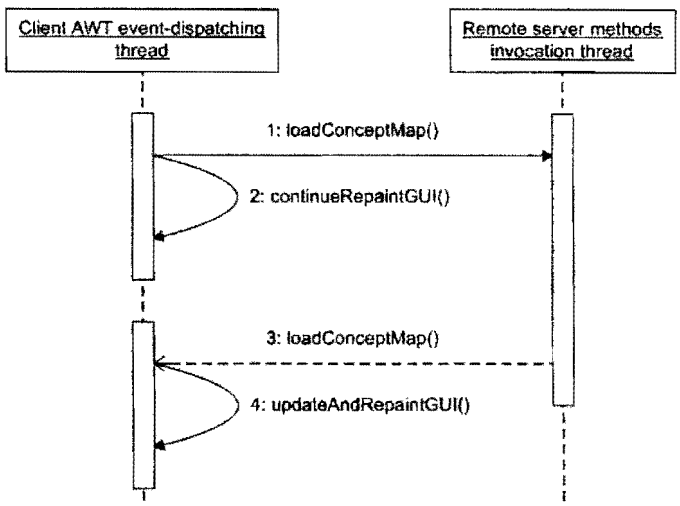

Fig. 3. An example of the KAS GUI working principles

The figure above shows that the client GUI thread does not depend on server methods invocation thread any more. However, there is one important problem. While the server is busy, but the client thread is alive and continues its operation, the user could press the same button once again, so a new event will be generated and another copy of the same service request will be sent to the server. So two or more the same methods could be simultaneously executed on the server side and this situation might cause unpredictable consequences. To prevent such a possibility a GUI lock mechanism was invited, which working principles are very simple: while the client side is waiting and in the same time continuing refreshing GUI, all AWT input events are artificially consumed. So a user could not affect the current application's window. After the response of the server is received, the GUI is refreshed, using data returned by the server, and it is unlocked for the next user's actions.

The new GUI engine provides smoother and faster graphical components repainting. Of course, GUI painting speed depends on server response time, because it is impossible to "draw" the data, which is not arrived on the client side yet. However, the application hanging problem was solved.

3.4 Summary

Evolution of technical aspects of the KAS is summarized in Table 1.

Table 1. General technical features of KAS's versions.

KAS's version	General features of the version	Tools used for the implementation of the version	Drawbacks of the version identified during evaluation
I	Simple two-tier client-server architecture	- Borland JBuilder 9.0 - JGraph - PostgreSQL DBMS 8.0.3 - HTTP Apache Server - Web Start - JDBC driver	- Low secureness - Low performance - The architecture is not very flexible (it's impossible to implement more advanced functionality, file sharing and transferring between users, for example)
II	Scalable and secured three-tier client-server architecture including the application server (with possibilities to implement more advanced and flexible functionality)	- Eclipse 3.2 - JGraph - JGoodies - Hibernate - VLDocking - PostgreSQL DBMS 8.1.3 - Apache Tomcat 6.0 - Web Start - JDBC driver	- Not smooth GUI refreshing - GUI hanging problem while waiting for server response
III	Client graphical user interface engine with multi-thread environment	- Eclipse 3.2 - JGraph - JGoodies - Hibernate - VLDocking - PostgreSQL DBMS 8.1.3 - Apache Tomcat 6.0 - Web Start - JDBC driver	*[Not evaluated yet]*

The table 1 shows that the first version of KAS was implemented as two-tier architecture [2]. After the practical evaluation of the first version of KAS it became clear that there is no future for the system evolution, because of low security, performance issues and scalability problems of two-tier client server architecture. Therefore, a decision was made to move the system onto three-tier architecture.

The second version of the system became more secure and flexible for adding new functionality. Performance of the system was increased significantly as well. However, a very important drawback was detected – GUI refreshing and hanging issues while waiting for application server responses. In order to solve this problem a GUI engine was refactored and migrated onto multithread environment in the last version of the system. It is expected that this improvement will bring KAS UI alive while server performs long operations. The last version of the system was not evaluated in practice yet. Its experimental evaluation will be held in autumn 2009 in three study courses.

4 Conclusions

This paper describes evolution of the concept map based knowledge assessment system developed by researchers from the Department of Systems Theory and Design of the Faculty of Computer Science and Information Technology of Riga Technical University. The paper emphasizes technical aspects of system's implementation.

Three versions of the system have been already implemented and evaluated. Drawbacks and technical issues revealed during experimental evaluation of the system resulted in incremental improvement of system's architecture and its functionality. The main problems of the first prototype were related to low performance and secureness of the system. With aim to eliminate the mentioned problems the transition from the two-tier architecture to the three-tier architecture was made in the second version of the system. However, a new issue concerning slow operation of GUI engine on the client side was identified and was chosen as one of the new improvement directions in the third version of the system. As a result of implemented changes system's performance increased significantly and the system response time to user initiated actions became quite reasonable.

References

1. Bloom, B.S.: Taxonomy of Education Objectives. Handbook I: the Cognitive Domain – New York: David McKay Co Inc., P.207 (1956)
2. Lukasenko, R., Vilkelis, M., Anohina, A.: Deciding on the Architecture of the Concept Map Based Knowledge Assessment System. Proceeding of the International Conference on Computer Systems and Technologies, pp.V.3-1 - V.3-6. Bulgaria (2008)
3. Grundspenkis, J., Anohina, A.: Evolution of the concept map based adaptive knowledge assessment system: implementation and evaluation results. Scientific Proceedings of Riga Technical University "Computer Science, Applied Computer Systems". RTU Publishing, Riga (2008)
4. Vilkelis, M., Anohina, A., Lukashenko, R. Architecture and Working Principles of the Concept Map Based Knowledge Assessment System. Proceedings of the 3rd International Conference on Virtual Learning, pp. 81-90. Romania (2008)
5. Threads and Swing, http://www.it.uu.se/edu/course/homepage/devgui/vt03/out/ThreadsAndSwing.pdf (last visited on 10.04.2009)
6. Java programming language. Multithreading and concurrent programming, http://www3.ntu.edu.sg/home/ehchua/programming/java/J5e_multithreading.html (last visited on 15.04.2009)
7. Java Swing Components, http://www.cs.qub.ac.uk/~P.Hanna/JavaProgramming/Lecture5/Java%20%20Lecture%205%20-%20Components.pdf (last visited on 19.04.2009)
8. Thread and AWT/Swing, http://www.ibm.com/ developerworks/library/j-thread.html (last visited on 20.04.2009)
9. The Last word in Swing threads, http://java.sun.com/products/jfc/tsc/articles/threads/threads3.html (last visited on 20.04.2009)
10. Swing thread, http://mindprod.com/jgloss/ swingthreads.html (last visited on 20.04.2009)