IJCTA, 9(30), 2016, pp. 289-295
© International Science Press

Special Aspects of Modeling and Software Implementation of the Data Caching Control Tools

Svetlana Jur'evna Beletskaya* Vera Nikolaevna Kostrova* Oleg Jakovlevich Kravets * Sergey Mikhajlovich Pasmurnov* Oleg Valerjevich Rodionov* and Evgeny Vasil'evich Shvedov*

Abstract : We review modeling and software implementation of the data caching tools. The aim of this work is to build the information model and develop principles of interaction of the modular structure of caching protocol, which refers to a certain combination and the set of interface transactions, in the frame of which data exchange, transmission and processing between different programs or program components are determined.

To select a software development environment (Software) and design of the DB structure, we should do the following: design the main data objects that will form the DB basis by the example of Internet-based information systems (IS); design Software modular structure by the example of Internet-based IS; design special software by the example of a special transaction processor.

As a result, the interfaces and data exchange mechanisms are defined. Special software for a special transaction processor for integration in an Internet-based IS is developed.

Keywords : Database, caching control, transaction processor, software interfaces.

1. INTRODUCTION

In the context of formation and analysis of the choice of software alternatives, database (DB) is one of the most important solutions in the process of program implementation.

Eventually, in order to choose a software (Software) development environment and to design the DB structure, we have to perform the following tasks:

- To develop the design of the main data objects that will form the basis of the DB by the example of Internet-based information system (IS),
- To develop the design of Software modular structure by the example of Internet-based IS,
- To develop the design of special software by the example of a special transaction processor.

In order to implement the integration of special Software in the Internet-based IS, we should define interfaces and data exchange mechanisms from both directions. In particular, to ensure coordinated operation of the main interfaces of data exchange between Software levels, we may use the following interaction interfaces:

- API (Application programming interface) - a set of classes, procedures, functions provided by the application, library for use in external software products,
- EJB (Enterprise JavaBeans) is a set of specifications that allows creating distributed applications to which the low-level API (Application programming interface) for transaction handling are applicable [1, 2, 3],
- QL (Structured Query Language) - universal structured query language used for exchanging, creating and control of data in a DB [4].

* Voronezh state technical university Moscow ave., 14, Voronezh, 394026, Russian Federation

- JDBC (Java DataBase Connectivity) - independent industry standard run on any platform for the interaction of Java applications with various DBs [5].

The above listed technologies are included in such software development environments like NetBeans IDE 7.

2. METHODS

Java programming language and free integrated development environment IDE NetBeans IDE 7 were used for the implementation of special software for data caching control. This specific software is designed to run under Windows operational system. Also, for the successful operation of the developed special software, JBoss Application Server using a servlet container should be installed.

On the basis of previously constructed algorithms [6, 7, 8] we developed special software of special transaction processor for integration in the Internet-based IS.

In the process of development using a Net Beans IDE 7 software development environment, the EJB module is created. This module implements the functionality of the special transaction processor and consists of a set of EJB components and JDBC driver. The EJB module and the EJB applications are interchangeable notions.

While designing the module in a Net Beans IDE 7 software environment, the module structure consisting of 3 functional layers was determined:

- **Web level.** This level contains a representation of the concept of the EJB module. This application runs on a JBoss Application Server and is a part of the Internet-based information system supporting teaching and research activities. Web level in the EJB module is implemented using the servlet that provides the client with access to the application (in our case, the client application is a web browser).
- **The level of processing.** This level contains the program code of processing the queries coming from the client to the EJB module. In the process of queries processing, data caching and transaction control are implemented. Level of processing in the module is essential to a process and directly represents a set of interfaces that control the data caching.
- **The third level is the level of interaction with the data source.** This level is represented by the JDBC driver in which the algorithm of caching protocol is implemented.

The final version of the run-time module is packaged with the help of NetBeans IDE 7 and has.jar extension. Next comes its deployment on the application server.

In the process of development with J2EE (Java 2 Enterprise Edition) technology application, which an EJB module was implemented.

Finally, EJB module, as part of a special transaction processor, is a set of controlling and interacting components with a set of classes. Each component class contains its own interfaces, methods, and processing functions.

In our case, for the interaction of the EJB module with the client and in particular with the web browser, the control component Manage Componet_Special Handling Transactions was created. This component is intended to handle queries coming from the client (web browser) via the servlet to the EJB module.

Then we developed class library that contains the remote interface - Remote (Remote Interface_ Special Handling Transactions). We need the class library because the client implements the access to the EJB-module components through a remote interface. Since we have chosen three-tier client-server architecture, on which special software is deployed, a remote interface is an essential element for the remote communication.

The next important point is the development of a session component Session Componet _Special Handling Transactions with a remote interface Remote Interface_Special Handling Transactions in the class library. Thus, it is possible to notice that Remote Interface_Special Handling Transactions interface is implemented in a session component Session Componet _Special Handling Transactions. At the same time, the remote interface Remote Interface_Special Handling Transactions is a class library of EJB module, in which the parameters and rules of interaction are defined.

The next step is to develop session facade of the session component (Session Componet_Special Handling Transactions). It is necessary because the clients using special software locally should also have access to the session component but without the remote interface (Session Componet _Special Handling Transactions). For this, we developed a special class for session facade Session Front_Special Handling Transactions. This class contains the program code of transaction handling of the EJB module, using the User Transaction interface.

After the development of a session component in the process of program design, the interface that implements the access to DB via HTTP is declared. As noted earlier, the use of a JDBC driver will allow us to develop an EJB module that will not depend on the platform on which a DB is built (*e.g.*, MySQL, SQLServer, Oracle, DBase). In our case, the JDBC driver of special transaction processor (EJB module) allows us to connect to IS DB, send the queries and receive the processing results. The special transaction processor JDBC driver consists of a certain set of classes and methods in which the processing parameters of database queries are defined.

The mechanism of a special transaction processor driver operation consists of the following (fig. 1):

- Through a web browser (web interface), client accesses an EJB module via servlet, using the remote interface Remote Interface_Special Handling Transactions.
- In the EJB module, the control component Manage Componet_Special Handling Transactions accesses the driver JDBC_Special Handling Transactions.
- Driver translates and processes BD queries, using the interfaces Statement, Prepared Statement, Result Set.

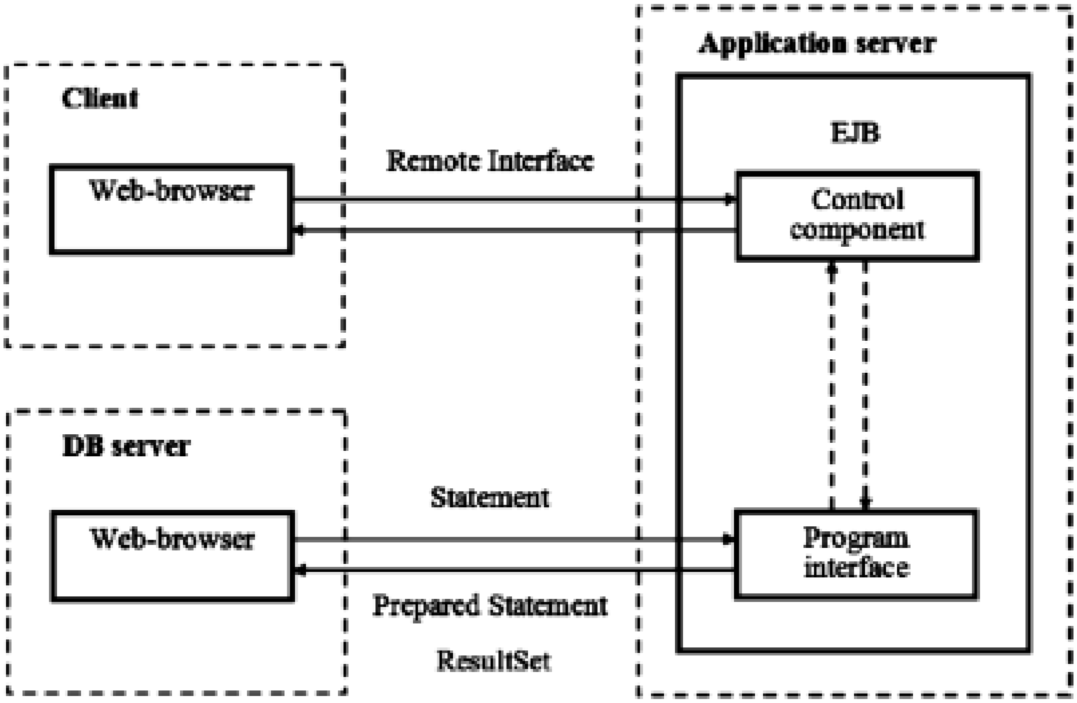

Fig. 1. Structure model of the interface interaction of a special transaction processor.

In the context of the Statement interface in the EJB module, on the basis of the described mathematical tool [7] the algorithm of protocol of caching based on time stamps [8, 9] is implemented.

We understand a caching protocol as a certain combination and a set of interface sequences, in the frame of which the data exchange, transmission and processing between different programs or program components are defined. Within the framework of the study, we understand interface as a set of rules, methods and algorithms of interaction between the programs and program components.

Call EJB_ Special Handling Transactions method in its own class EJB Access_ Special Handling Transactions and as a part of the servlet is responsible for calling the EJB module at the client level.

Finally, a special transaction processor includes the following components (Fig. 2):

- The control component that implements the EJB module control interface,
- Driver of interaction with the BD,
- A session component that implements the remote interface of interaction between the client and special transaction processor

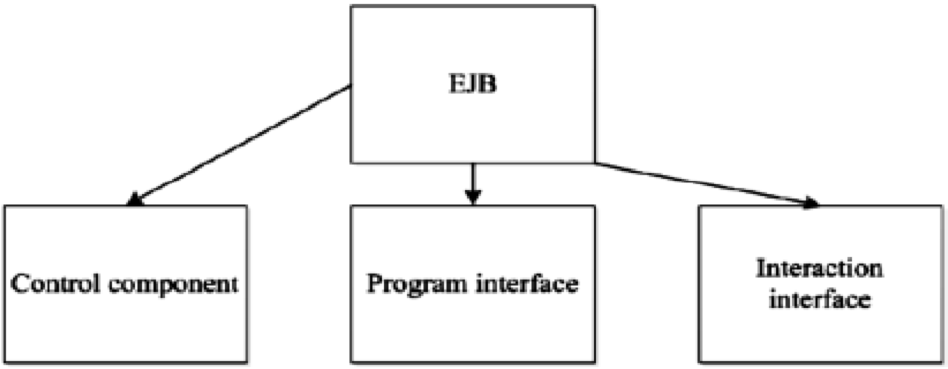

Fig. 2. Combination of the main interface components of a special transaction processor.

3. DISCUSSION AND RESULTS

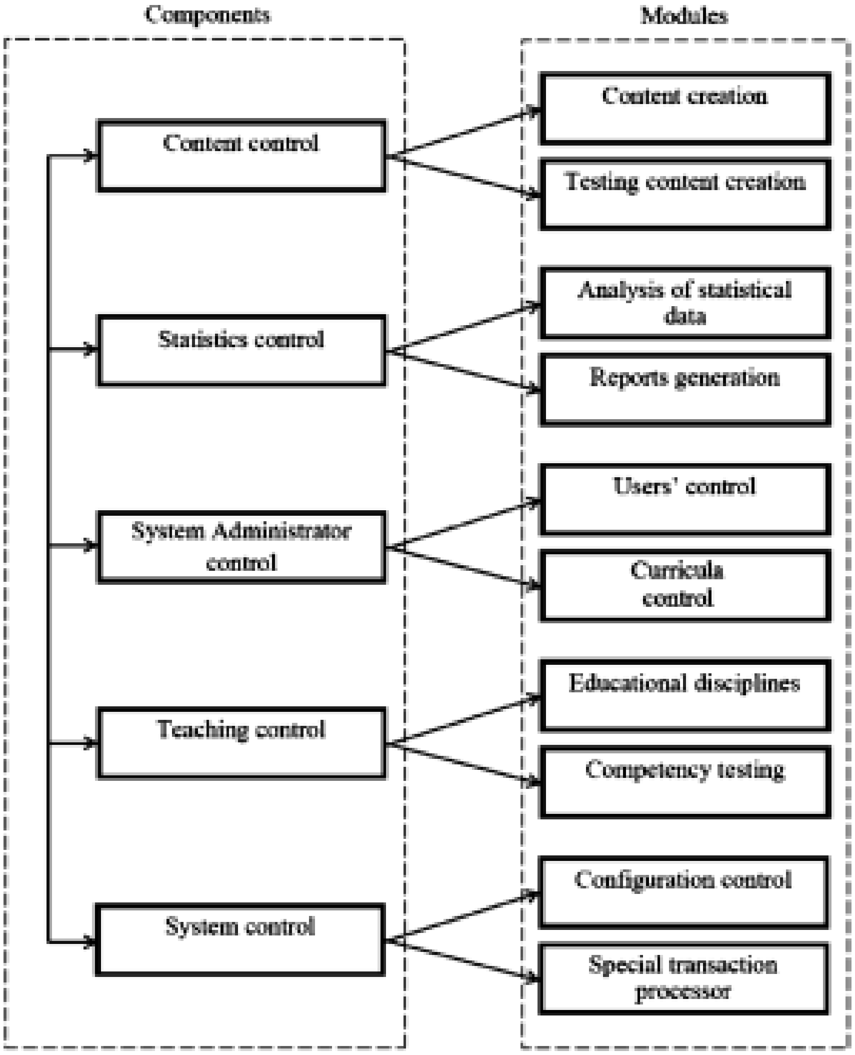

Fig. 3. The modular structure of the Internet-based information system.

In the process of designing the Internet-based information system (IS), we used the approach to building with application of a modular architecture. The main advantage of modular architecture is that each of the main structural units (consisting of modules) can be modified without requiring changes to the others. In this case the IS has the flexibility to meet the changing needs of the organization.

The Internet-based information system is designed in accordance with a modular concept (Fig. 3).

The main option for the use of IS is distributed processing of data and interaction with user. The basic principle fixed in the basis of the design is the possibility of distribution in the network as for DB and for the main modules of the system as well. The main features of distributed architecture are remote application, scalability, platform interoperability and performance. The main components (participants) of this architecture interaction are the client (user), the application server (in our case this is a web server) and the DB. The application server here is considered as a server program that allows the installation of application-specific components, so that they can be remotely called (usually using one of the methods of the remote object call) [10, 11, 12, 13, 14]. Fig. 4 illustrates three-level architecture of the interaction of the special software main components. In the framework of the three-tier architecture, the remote client can access a remote data source via the application server [10]. Connection of client, application server (web server) and DB, as well as the data processing and transfer are implemented via the Internet. The client performs the operation with the system using the web interface. The result of inter-module interaction is data processing and storage, as well as implementation of the main functionality of the developed IS [15].

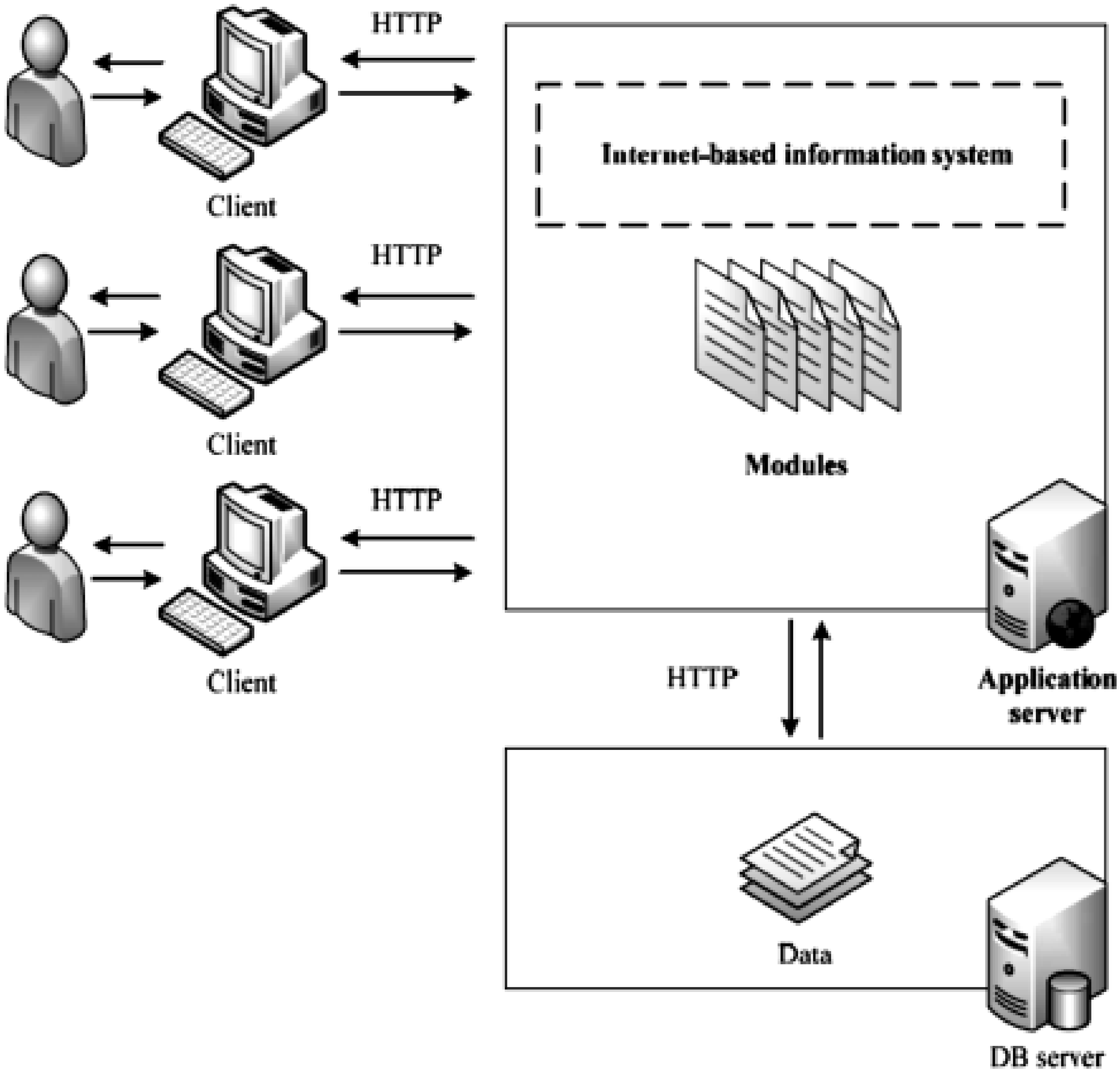

Fig. 4. Architecture of main components interaction of Internet-based information system.

Main data flows of special software of the Internet-based IS

Via web interface, the client sends a query to the “Users’ control” module to check the name and password in the DB. When authentication is performed, the user individual curriculum is initialized. Via the module “Ñurricula control”, the user controls his/her curriculum and choice of disciplines for study. Then a content creation query to the “Disciplines” module is performed. The user accesses the module “Competence testing” for the validation of competences. To analyze and generate statistical reports, the user uses the “Analysis of statistical data” module and the “Reports generation” module. In the “Educational content creation” module, the user performs the creation of educational disciplines. Development of tasks for the competence validation is implemented in the “Testing content creation” module.

The input data for “Content Control” block will be the data which are entered (imported) by the user. For example, the development and editing of content intended for education and validation of competences. The output data are the final presentation of the content which is its visualization.

The input data for “Teaching Control” block will be the entered data necessary for the competence validation that is selection or input from the response console. The output data will be the information processed and presented in the form of the final result.

The input data for “Statistics Control” block will be the entered information or selection criteria for the formation of report documentation. The output data in this case will be graphical and textual representation of the processed data, *i.e*. the report view.

The input data for “System Administrator Control” block will be the entered information about the user which is used to form the user profile configuration.

4. CONCLUSION

Thus, we have the macro flowchart of the main data flows and, in particular, the data extraction (reading) and adding (writing) in the DB system. Based on selected objects and described properties, we have built a logical model of the DB of the Internet-based IS for educational and scientific activities support.

5. REFERENCES

1. Monson-Haefel R. 2002. Enterprise JavaBeans, 3rd edition. Translation from English. SPb.: Simvol-PLYUS, p. 312.
2. Khabibulin I. S. 2003. Development of Web-services by means of Java code. SPb.: BHV ,Peterburg, p. 400.
3. Kravets O.Ja. 2013. Mathematical Modeling of Parametrized TCP Protocol. Automation and Remote Control, 74(7): 1218-1224.
4. McLaughlin B. 2013. PHP and MySQL. Comprehensive guide. SPb.: Peter, p. 512.
5. Kite T. 2007. Oracle for professionals: architecture, techniques of programming and the main features of version 9i and 10g.: Translation from English. Moscow: Publishing house “Williams”, p. 848.
6. Shakhvorostov G.I. and Kravets O.Ja. 2013. Special transaction processor application for the productivity of transaction processing increase and optimization in a specialized information system. Modern informatization problems in simulation and social technologies. Lorman, MS, USA: Science Book Publishing House, p. 283-289.
7. Shakhvorostov G.I. and Kravets O.Ja. 2013. Transaction theory features application as tool for the data caching special protocol implementation. Modern informatization problems in economics and safety. Lorman, MS, USA: Science Book Publishing House, p. 44-49.
8. Shakhvorostov G.I. and Kravets O.Ja. 2013. Features of the data caching protocol development on the basis of temporal tags for the special transaction processor. Modern informatization problems in the technological and telecommunication systems analysis and synthesis. - Lorman, MS, USA: Science Book Publishing House, p. 330-335.
9. Shakhvorostov G.I., Kravets O.Ja. and Solomakhin A.N. 2014. Control data caching based on transaction timestamp. - Yelm, WA, USA: Science Book Publishing House, p. 160.

10. GOST R ISO 13374-2-2011. The condition monitoring and diagnostics of machines. Evaluation of quality of software facilities. General provisions. Part 2. Data processing.

11. Florian Haack. 2011. Anforderungen an Ein It-Programm-Controlling: Eine Einzelfallstudie Anhand Der Einfuhrung Des Allianz Business Systems. Gabler Verlag, p. 248.

12. Darakhelevidze P. G. and Markov, E. P. 2003. The Development of Web services using Delphi. SPb.: BHV – Peterburg, p. 627.

13. Intelligent Engineering Systems Through Artificial Neural Networks, Volume 15: Smart Engineering System Design: Neural Networks, Evolutionary Programm. 2005. In Cihan H. Dagli (Editor). American Society of Mechanical Engineers, p. 894.

14. Ponamarev V. 2002. Delphi Programming with COM and ActiveX. Delmar Thomson Learning, p. 624.

15. Kravets O.Ya., Makarov O.Yu., Oleinikova S.A., Pitolin V.M. and Choporov O.N. 2013. Switching subsystems within the framework of distributed operational annunciator and monitoring systems: program design features. Automation and Remote Control, 74(11): 1919-1925.