

The Use of Object-Oriented Technologies for Medical Data Storing and Retrieving

Vita Speckauskiene, Arunas Lukosevicius

Biomedical Engineering Institute at Kaunas University of Technology, Lithuania

Summary: The article focuses on object oriented technologies and the usage of these technologies, how to create an information system (IS) for applications in medicine, specifically in ophthalmology. The rapid evolution of medical data standards, concepts and technologies has created a problem – how to develop a system that would be flexible enough to work with evolving data. After the analysis of object-oriented technologies: Java 2 Platform, Enterprise Edition (J2EE) and Microsoft .NET and open source object-oriented data base management systems (OODBMS) we have proposed and created a system based on .NET and OODBMS db4o. Results include the developed system with proposed architecture and used techniques, also the advantages of object-oriented data bases (OODB), the systems' possible interoperability with the National Electronic Health Record system. Developed IS should help medical specialists to reach important information from any computer in any hospital.

Keywords: object-oriented technology, object-oriented data base, information system, data storing standards

Introduction

Information systems (IS) currently used in the medical field are one of the most evolving components for e-health. The main features of IS in medicine are 1) storage of multimedia; 2) information connections in complicated and evolving semantic relations; 3) big amounts of information involving all lifelong patients' records; 4) wide list of users, spanning between all three health care levels (they create a need of secure and convenient communication between these levels and IS); 5) compatibility of IS with the European Union (EU) [1] and international standards [2, 6]; and 6) usage of data for clinical decision support.

These factors raise specific requirements for IS and their architectures. Therefore, a search for new technologies in order to develop such systems is in demand [3, 4, 5, 7, 16, 17]. A number of standards has been created to store and transmit all data of Electronic Health Records (EHR) including medical images. The problem is how to make the best benefit of this data for users. The data has to be stored in such a way that it would be easy to generalize and retrieve it. Therefore, standards are under development for interchanging of medical data. Each country has its own way of solving this problem, so one will stumble over a number of different standards describing various IS and data storing aspects: ASTM, ASC X12, IEEE/MEDIX, NCPDP, OpenEHR, HL7 (Health Level 7) [6], DICOM [2] and others. An organization developing a standard usually specializes in one particular area. Unfortunately, those standards optimize and facilitate the work with medical data, but data processing and storage is still a technological problem. The technologies that are used for a long time are not flexible enough to work with such data. Medical data is stored in relational data bases (RDB); though studies [18, 19] show that object-oriented data bases (OODB) bring a lot of new advantages. Also the need of secure and flexible Internet forces to look for effective programming tools and technologies.

The aim of this article is to show the advantages of OODBMS, also to present the architecture and develop a system for medicine which in certain extent satisfies the above mentioned features. To obtain this goal, object-oriented data bases, compatible programming techniques, data storing methods, and main features of IS for medicine are analyzed and applied. Also, this system is being prepared for integration into the National EHR System (Lithuanian abbreviation NESS), which is now being developed (at the national

level). Unfortunately, we are not able to show the interoperability between NESS and the developed system, because NESS is not yet implemented. But still we have made a few observations.

Methods

The analysis of object-oriented data bases OODBMS is a new quickly developing technology which is increasingly applied by many different companies. The benefits of OODBMS bring a rise in users as well as demands. These DB have distinctive data storing methods. The data is stored in tables, which can have tables in themselves – they form a hierarchy. The tables are filed not only with concrete data, but also with objects. This technology can be implemented effectively for medical data storing. Because of the evolution of medical requirements OODBMS makes the systems more flexible and stable [18, 19]. It also enables interaction between objects in the data base.

RDBMS are convenient in storing text and numerical data; but they cannot work with undefined records which are formed in object-oriented programming languages. Moreover, these data base management systems (DBMS) are not fitted for storing methods. IS requires to work not only with a text, but also with spreadsheets, images, diagrams, sounds and other multimedia. Besides, all this information has to be organized as a united structure, handy for users. The better decision might be to implement OODBMS. We will see that this technology is able to vouch a new approach of data storing and mining.

OODBMS stores composite objects which compound out of other objects. These objects are stored with references to lower level objects. The number of levels is unlimited. Each object can be brought into many other objects as components. This forms a net structure data model. OODBMS allows construction of new data

types; this feature enables to store multimedia: images, audio and video files. There are two types of OODB: passive and active. Passive OODB save the object structure but do not control behaviour of objects. While active OODB allow communication between objects within the DB.

A lot of companies offer the products of OODBMS, but most of them are payable. The most known are listed in Table 1 [8].

Table 1. Commercial OODBMS.

OODBMS	Vendor
Versant Developer Suite and FastObjects	Versant Corporation
TITANIUM	Micro Data Base Systems, Inc.
GemStone/S and GemStone Facets	GemStone Systems, Inc.
Cache'	InterSystems Corporation
JADE	Jade Software Corporation
JYD Object Database	JYD Software Engineering Pty Ltd.
VOSS	Logic Arts, Ltd.
Objectivity/DB	Objectivity, Inc.
ObjectStore Enterprise and PSE Pro	Progress Software Corporation

platforms used are J2EE and .NET. The companies that offer these products are one of the biggest competitors. So, the developer has a dilemma – which one to choose. Sun Microsystems offers to use Java 2 Platform, Enterprise Edition (J2EE) and Microsoft – .NET technology. Both technologies allow implementation of web services. The main advantage of J2EE is the fact that it is a multiplatform – it is possible to use J2EE in different operating systems (OS). Microsoft .NET works only in Windows OS, although the Microsoft company offers an equivalent tool for Linux OS. Both technologies use object-oriented programming languages (.NET – Visual Basic and others, J2EE – JAVA). The advantage of .NET is its capability to render pages in various HTML formats, which abstracts developers from version-specific HTML.

Servlets in Java can achieve the same task, though with more manual coding [7]. Another difference and advantage of .NET is its multilanguage; it works with Visual Basic, C#, FORTRAN and COBOL; J2EE works in Java programming language. The main features of both technologies are presented in Table 3 [3, 4, 16].

As we can see both technologies are very similar. Most important is that both of them implement web services and are able to map data on the Internet. A .NET Web service could serve as a consumer to a J2EE Web service and vice versa because of the same Web services standards used [7]. Moreover, web services can be reused no matter what programming language they were coded in. Most authors do not point out particular reasons why one should choose either technology [5, 16, 17]. Still we can note some advantages of .NET in its development and handling capabilities. That's why this technology was selected for implementation.

Table 2. Open source OODBMS.

OODBMS	Programming environment	Limitations
db4o	J2EE .NET	Handling of default and null values (noticed during implementation).
SOD2	J2EE C++	Does not have a query mechanism.
ozone	J2EE	Implemented in JAVA and works only in the J2EE environment. Does not have a query mechanism.
ObjectDB	J2EE	Implemented in JAVA and works only in the J2EE environment. Free DB edition does not support a client-server architecture (unless the DBMS is in the same server).
GigaBASE	J2EE C++	Does not support null values.
XL2	J2EE	Does not have data recovery functions. Does not have a query mechanism.
Orient	J2EE, C++	Available for C++ (ODMG 3.0 interface) and Java with the support of Sun JDO 1.0 technology.

System architecture and functionality IS for medicine data mining could be implemented using both above analyzed technologies. No matter which object-oriented technology is chosen, the system architecture is analogical, it is presented in Figure 1.

Still there is a part (and much bigger) of software developers who propose free products. In this article we have selected to review seven open source OODBMS; they are listed in Table 2 [10, 11, 12, 13, 14, 15]. We can see that open source data bases are implemented for J2EE, and not all of them correspond to the standards, as well as overall OODB flexibility. The standards of OODB are handled by the Object Database Management Group (ODMG) [9]. Since (at the analysis stage) we did not penetrate any limitations of OODBMS db4o, we chose it for the development of the final result.

Comparison between J2EE and .NET technologies

Web service technologies are very widely used because they provide to programmers a lot of capabilities and their implementation is not difficult. The main

Table 3. Comparison of main features of J2EE and .NET.

Feature	J2EE	.NET
Relational DB Access	Java Database Connectivity (JDBC) or L/J	Active Data Objects (ADO.NET)
Web Client	JAVA server pages (JSP) and servlets	ASP.NET
Graphical user interface (GUI)	AWT/Swing	Windows forms
Messaging	Java Messaging Service (JMS 1.0)	Microsoft Messaging Queuing (MSMQ)
Web Services Support	Java Web Services Developer Pack (JWS DP)	Built directly into .NET and Visual Studio
Infrastructure	Enterprise Java Beans (EJB)	COM+
Naming and Directory Service	Java Naming and Directory Interface (JNDI)	Active Directory Services Interface (ADSI)
Database storage	-	SQLServer (the official .NET platform database technology)

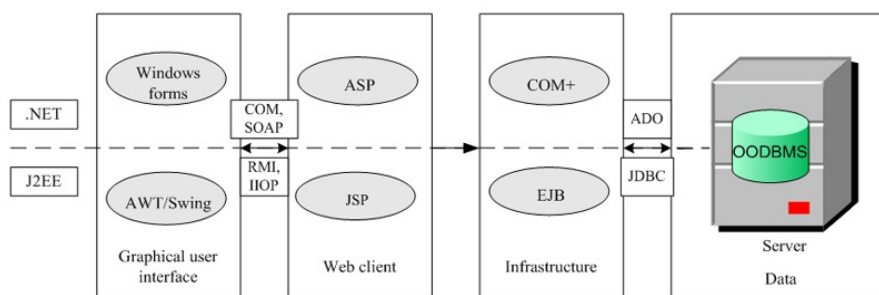


Figure 1. Architecture of medical data IS.

The advantage of similarity of those technologies is that both systems can communicate with the same DBMS, if there is a necessity to communicate. As mentioned in the above sections, for implementation we chose OODBMS db4o [10] and .NET technology to create a medical data storing, retrieving and mining system. In Figure 2 there is presented the architecture of the system we have developed.

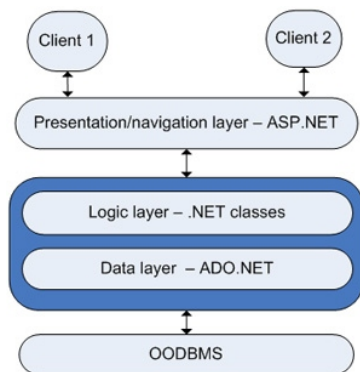


Figure 2. Architecture of implemented IS.

As we can see the system consists of three layers: presentation, logic and data. The communication between presentation and data layers is achieved through the logic layer. In Figure 3 there is presented a use case diagram of implemented IS.

The use case diagram shows the functionality of the system (the functions that a user and administrator can perform).

Results

The system has been developed using the above mentioned technologies. The integration between .NET and db4o did not cause any bigger problems. The communication among the application and the data base itself is achieved through the standard data base query mechanism, included in the standard database libraries. The examples of the source code, showing OODB queries and programming capabilities, used in the developed system, are presented in Table 4.

The data base is a single .yap type file; a blob type file can be any file with no limitation in size or file extension (if desired). So data and file storing and updating mechanisms are very flexible and convenient for a software engineer.

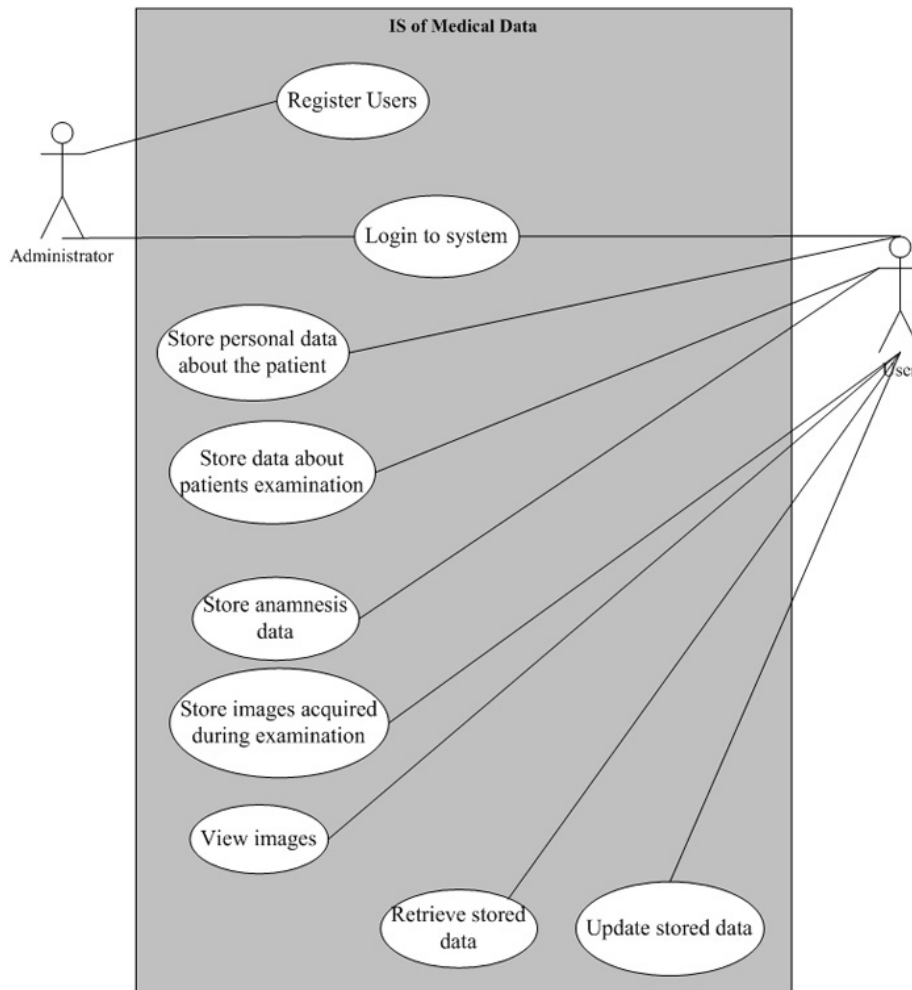


Figure 3. Use case diagram.

Table 4. Parts of the source code of OODB queries and programming capabilities used in the developed system.

using com.db4o; using com.db4o.query;	Standard libraries (mainly used)
Connecting to the data base and making a query	
ObjectServer db4oServer = Db4o.openServer("db.yap", 4488); db4oServer.grantAccess("..."); ObjectContainer client = Db4o.openClient("..."); Query query1 = client.query(); query1.constrain(typeof(Vartotinfo)); query1.descend("_vartotojas").descend("_pavarde").constrain(p.Text); ObjectSet result1 = query1.execute(); client.close(); db4oServer.close();	Connection to the server Create server client Make a query Set the class to look into Find a particular data Execute query Closing the client Closing the server
Setting and updating new data	
Vartotojas vartotojasn = new Vartotojas(P1.Value,p.Text,'N'); client.set(vartotojasn); Vartotinfo vartotinfo = new Vartotinfo(v.Text,el.Text,g.Text,k.Text,s.Text); vartotinfo.Vartotojas = vartotojasn; client.set(vartotinfo);	Generating new data for class "Vartotojas" Storing the data Generating new data for class "Vartotinfo" Creating a reference Storing the data
Working with "blob" type files	
Db4o.configure().setBlobPath("blobs"); Byla snap = new Byla("..."); client.set(snap); snap.Bylapav.readFrom(new File(File1.Value)); double status = snap.Bylapav.getStatus(); while(status > Status.COMPLETED) { status = snap.Bylapav.getStatus(); }	Setting the blob path Generating new data for class "Byla" Storing the data (also the text data about the file: the file name, extension and size) Preparing to store the blob type file itself Storing the file itself
Byla found = (Byla)result.next(); string pav = found.Bylapav.GetFileName();	Querying for the file Retrieving the file

Testing of the system: method.

After the development we tested the system and made an experiment with various system changes. The purpose of the experiment was to show the flexibility of used technologies in making changes in all three system layers – data base, user interface and logical layer. To accomplish this task, we made an experiment in which we dealt with an evolving data problem. We created a scenario in which the functional requirements are complemented. The new data containing parameters of images are to be added to the system. First of all we tested the systems, then made changes according to the scenario and then again tested the system including the changes we made. The following four sections describe the testing procedure, changes in data base, presentation and logic layers.

The system was tested using the data that was collected during patients' examinations in Clinics of Kaunas University of Medicine. We used a limited amount of data, because our main task was to see how the data behaves in OODBMS, and not to measure any rates yet. We tested user's functions (e.g. store/retrieve patient information, analyze patient's image), administrator's functions (e.g. user registration), and user interface (e.g. navigation links, data mining results). All the procedures were performed successfully, as well as the overall testing.

Changes in the data base.

To show the flexibility of OODBMS a similar RDB model (to OODB classes) was created. The changes were compared using visual diagrams. A simplified example of one part of the visual diagram is shown in Figure 4.

The figure shows that there is less information stored in OODB; there is no need to store identifiers because the connection between classes is made by an operation ("EyeImageData()") in Figure 4 a). The field "Filename" (seen in Figure 4 b)) is needless in OODB because the blob type (any multimedia or other type) file information is generated automatically in the DB engine. So there is no need to create extra fields between tables, and the data is not doubled, also the multimedia data management is more convenient.

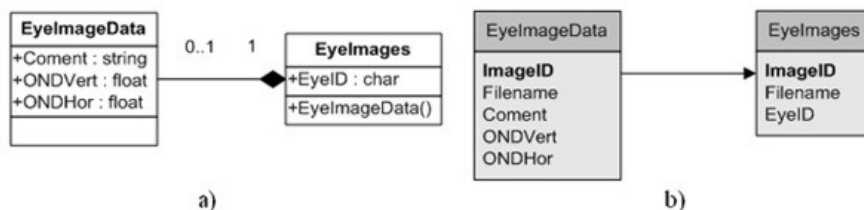


Figure 4. Data base models: a) OODB b) RDB.

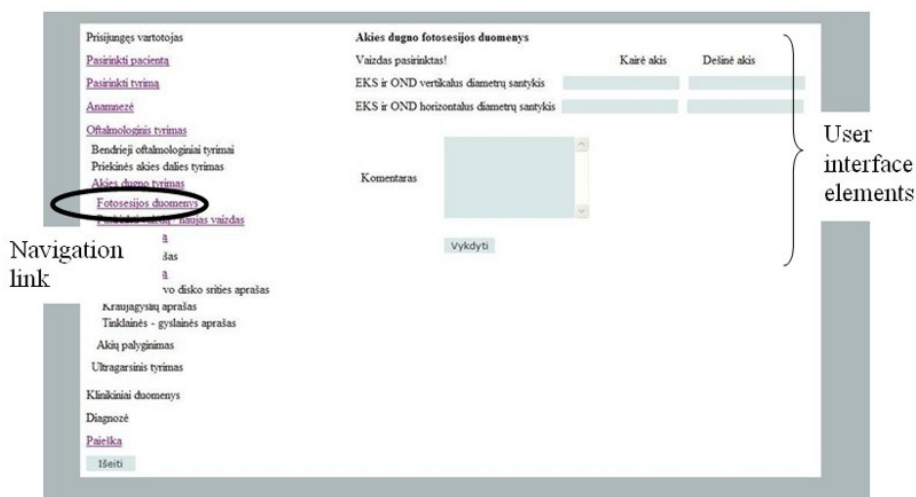


Figure 5. Changes in user interface.

Changes in user interface.

The changes in the presentation layer are equivalent in both (OODB and RDB) cases. Figure 5 shows the changes in the created system with OODB.

Changes in the logic layer.

The logic layer is more complicated. For both cases it consists of the following changes:

- programming code for new user interface elements,
- performing test for these elements using the remaining data,
- storing all the remaining data.

We will not go into detail how the OODBMS behaves in the case of these changes, but using the OODBMS db4o data management software „Object Manager“, we could see that DB structures are updated automatically, when the DB is activated. For the data store and update in the case showed in Figure 6 we need the same source code – only one statement is needed. However, in the RDB case we

need two SQL sentences (Insert Into and Update).

It obviously reduces the amount of the programming code. We can see this form in Figure 6, where the reference and new data are presented.

Also note that the reference (in the case showed above) spreads the new data, and avoids data duplication.

There should not be a problem to integrate the system into NESS, because the Integrating Healthcare Enterprise (IHE) is used for this purpose in NESS. Since db4o is a single file convertible to XML, which together with HL7 CDA Level 2 are applied in NESS, there should not be any trouble for systems to communicate.

Further observations and advantages of OODBMS are presented in the next section.

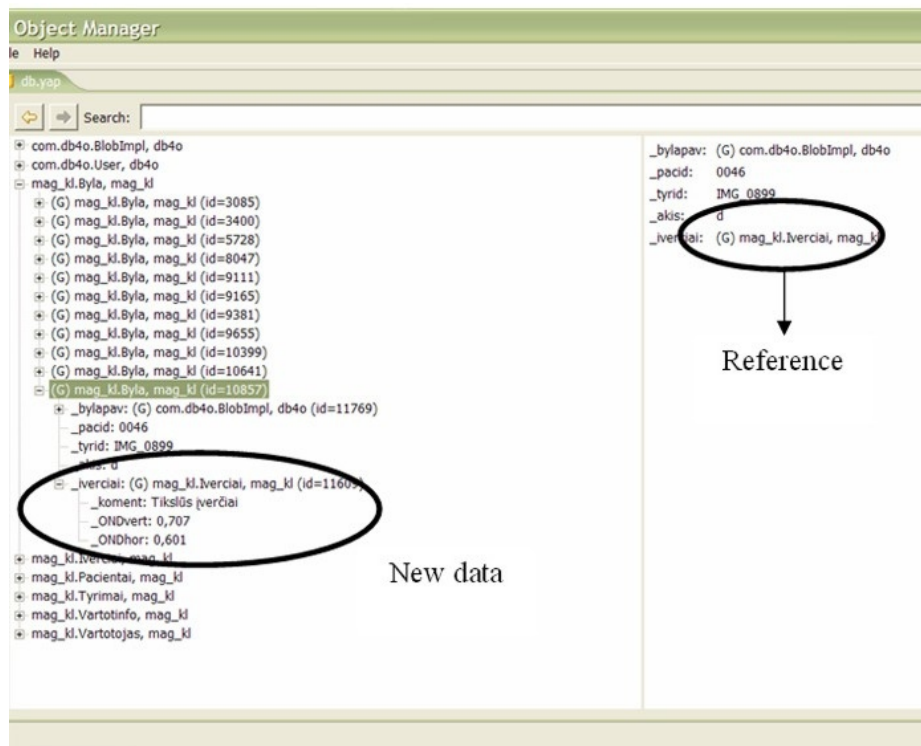


Figure 6. OODB reference and the new data.

Discussion and conclusions

Extensive international and collaborative work with standards and concepts gradually leads to the better quality of developed products, as well as software – the systems are able to work with different data types, to ensure compatibility and flexibility. In future we intend to integrate the system into NESS.

The analysis of OODBMS shows that they are very convenient for storing of semantically and quantitatively evolving medical data. It is possible to implement remote DB access using object-oriented technologies easily.

The developed IS had shown that such a system is flexible and evolutionistic because of the usage of appropriate technologies which are at the same time compatible with conceptual medical informatics standards. Although the system could be implemented by using different technologies, we prefer using .NET technology because of its development and handling capabilities, also the multilanguage ability and the user-friendly environment of Visual Studio .NET software.

Our experiment showed that OODMS has the following advantages:

- possibility to store and retrieve data with the same statement;
- absence of necessity to create additional fields between tables, because of the refeces OODBMS uses;
- absence of duplicated data;
- more flexible storage of multimedia data;
- absence of creating tables and links in the data level in the DB changing process.

For this purpose we used a small amount of data. However, we intend to store a large amount of data and rate systems behaviour.

It is rational, however, to mention that results presented do not imply refusal of RDBMS usage, rather we propose OODBMS as a flexible solution for solving specific problems related with inevitable and continuous evolution of eHealth systems.

References

- [1] CEN/TC 251 Standard Development Workgroup site, <http://www.centc251.org/>
- [2] DICOM standard site, <http://medical.nema.org/>

- [3] Estes B. T., Maxime O.: J2EE vs .Net: The Choice Depends On Your Needs. 2003; <http://www.computerworld.com/printthis/2003/0,4814,84155,00.html>
- [4] Farley J.: Microsoft .NET vs. J2EE: How Do They Stack Up?. 2000; http://java.oreilly.com/news/farley_0800.html
- [5] Hanson J. J.: .NET versus J2EE Web Services. A Comparison of Approaches. <http://www.webservicesarchitect.com/content/articles/hanson01.asp>
- [6] HL7 Standards Developing Organization site, <http://www.hl7.org/>
- [7] Lurie J., Belanger R. J.: Does One Web Services Platform Dominate the Other? 2002; <http://www.javaworld.com/javaworld/jw-03-2002/jw-0308-j2eenet.html>
- [8] Object-oriented database product vendors, http://www.service-architecture.com/products/object-oriented_databases.html
- [9] Object Database Management Group site, <http://www.odmg.org>
- [10] OODBMS db4o site, <http://www.db4o.com/>
- [11] OODBMS ObjectDB site, <http://www.objectdb.com/>
- [12] OODBMS ozone site, <http://ozone-db.org/frames/home/what.html>
- [13] OODBMS SOD2 site, <http://www.tneumann.de/sod/index.html>
- [14] OODBMS XL2 site, <http://www.xl2.net/>
- [15] O-RDB GigaBASE site, <http://www.garret.ru/~knizhnik/gigabase.html>
- [16] Sessions R.: Java 2 Enterprise Edition (J2EE) versus The .NET Platform Two Visions for eBusiness. 2001; <http://www.objectwatch.com/FinalJ2EEandDotNet.doc>
- [17] Sheil H., Monteiro M.: Rumble in the Jungle: J2EE versus .Net. How do J2EE and Microsoft's .Net compare in enterprise environments?. 2002; http://www.javaworld.com/javaworld/jw-06-2002/jw-0628-j2eevsnet_p.html
- [18] Sorli M., Romo J., Stach R., Bredehorst B.: REMOTE: GRD1-2000-25433. 2001, p. 40-42; www.remoteproject.org/downloads/REMOTE.E.D1.11.1--2001-07-09.pdf
- [19] Zamir S.: Handbook of Object Technology. CRC Press, 1999, 35ch.

Contact

Vita Speckauskiene

Biomedical Engineering Institute at
Kaunas University of Technology
Studentu str. 65 Kaunas, LT-51359
Lithuania
e-mail: vita.speckauskiene@med.kmu.lt