Middleware components in the distributed integrated University's information environment

Carina Shakhgeldyan, Vladimir Kryukov

Vladivostok State University of Economics,
carinash@vvsu.ru
kryukov@vvsu.ru

Abstract

Component-based model of the distributed integrated University's environment is the main subject of the paper. We discuss architecture of the environment, control components, their interactions with the projects and other components of the environment.

Keywords: Component-based architecture, web-service, integrated information environment

1 Introduction

Informatization of University is a complex process due to University has several specific features:

1. University is a big organization with updating staff and contingent of students.
2. University is organization with wide range of activities including education, research study, management, advertising, construction and so on.
3. There are many different information technologies in one University; sometimes it is impossible to use the only technology in the information environment.
4. The business processes of University are changeable.
5. University is open to exchange information between Universities, research centers, and other organizations.

These features have to be taken into consideration while the model of University's information environment is been developing. The development of the model is a part of common process of informatization. The activity contains the following items.

1. Construction of University's infrastructure.
2. Development of model University's information environment.
3. Development of middleware layer software.
4. Development of user's layer software.

In practice all of the items are fulfilled at the same time. Often the problems of user's layer result in modification of second and third layers, and even the first one. Of course in the view of end-users the forth layer is the most important. But in fact the second layer is the most important. If the model of the information environment has a lot of faults then the environment would get impossible to maintenance and use it for a long time.

There are several approaches to develop the information environment of University. One of them implies using client-server software and database management system (DBMS). Data are integrated into the unified DBMS and the applications get client-servers [1]. The second approach is application integration into unified software (ERP-system) [2].

We are considering the third approach that is integration of technologies, applications, and data. It seems the approach is the most attractive for University. It allows using various operating software (including ERP-systems) and databases. It allows developing new applications using different technologies and databases into heterogeneous environment. It allows implementing the integrated information environment of University (IIEU).

We had started considering the model of IIEU based on technology of integration and component approach in [3]. We are going to continue the description of the model and we are discussing the architecture of middleware components of IIEU. In this article we are discussing model of integrated information environment of University based on component approach. The demands of IIEU are offered in the second part of the article. Architecture is discussed in third part. In the forth part we discuss logic model of IIEU: control and applied components. We consider mathematical model of IIEU in fifth part, integration with branches in sixth, and quality of data in seventh.

2 The requirements to integrated information environment and integration technologies

We have to define the requirements to the information environments of University.

1. The users of IIEU are all students, teachers, and employees of University. They have authorized access to information resources of the University according to their rights and responsibilities at their University. The management of rights is both automatic and manual.

2. The applications of IIEU covered the directions of University activities: education, management, scientific

research, business, information technologies, etc. The applications have the regimes: data collection and saving, processing and analyzing, decision-making and execution of decision.

3. The component model is the base of architecture of IIEU. It allows integrating different technologies, applications, and data.

4. Unified repository of different DBMS provides data integration.

5. There are the procedures providing actuality, validity, consistency of data.

6. Extraction of general functions from all applications into the unified systems. Registration and access management, creation of reference books, reporting are such general functions that can be moved from the applications to unified systems.

7. Using specific parameters which estimate performance, efficiency, stability, application's usage, etc.

8. IIEU has to be opened for integration with other Universities and research organizations.

There are different technologies at University. ERP-systems (SAP\R3, OEBS, or Axapta) allow automating financial area, planning, and personal administration. Geo Information Systems (AcrVew/ArcInfo, MapObject/MapExtreme) allow managing accommodations of the University and controlling construction and repair. Lotus/Notes/Domino provides for document circulation. Oracle Application Server, Sun One, .Net or client-server applications support corporative portal, management of education, and education. Directory service (Active Directory, e-Directory, Sun ONE Directory Server) manages user accounts and access to network resources.

Every of the technologies can be used at University to solve some problem in the best way. The using all of the technologies (or part of them) is attractive for University. There is a way to use the technologies simultaneously.

Integration of technologies may be fulfilled on base of a component technology. CORBA, DCOM, J2EE, web-service may be used as the base technology of the integration. CORBA, J2EE, and DCOM have constraints: DCOM is limited by Windows operating system, CORBA and J2EE have the troubles with firewall which do not allow being open for outside integration. CORBA and J2EE are not used in .Net. But .Net is very popular technology in many Universities.

Web-services do not have the lacks [3]. They may integrate any technologies, operating systems, and do it through firewall [4]. But they have other disadvantages. The main shortcoming of web-services is lower performance than CORBA or DCOM. Performance may be increased using several servers and load balancing algorithms. Flexibility and ability to integrate any technologies make web-services the most attractive component technology for IIEU. Web-services, SOAP, WSDL are supported by Microsoft, Oracle, Sun, IBM, Bea and many others vendors what makes sure integration of wide range of technologies.

For example if a Lotus-application needs to authorize an user account from SQL Server database then the application should send request to the authentication web-service addressing to SQL Server database. The usage of web-services allows integrating various portal technologies (.Net, Sun One, OAS). It is useful for integration of several Universities.

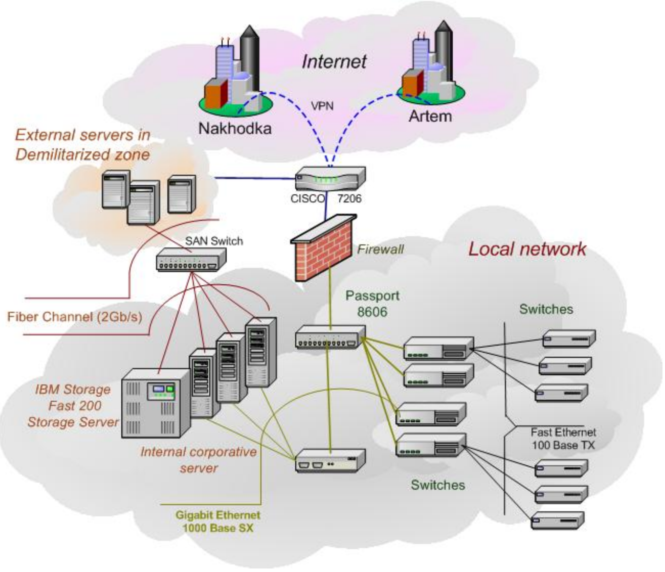
Web-services allow using any technologies inside the integrated information environment and not limiting one framework.

There are several different technologies at the Vladivostok State University of Economics (VSUE): .Net (C#, ASP), Oracle Application Server (J2EE), Lotus/Notes/Domino, client-server C++ and Delphi, MapObject (MapExtreme). Conception of web-services allows joining their into the integrated information environment.

3. Architecture of information environment

The information environment is integration of information infrastructure, information systems, and the data which the systems work with. In Russia the information infrastructure of University can consolidate both internal local network of the University and local nets of the University's branches.

The information infrastructure of VSUE is shown on picture 1. VSUE has two branches in Artem and Nakhodka. Internal local network of VUSE is separated from external demilitarized zone by firewall.


Internet
VPN
Nakhodka
Artem
External servers in Demilitarized zone
CISCO 7206
SAN Switch
Firewall
Local network
Passport 8606
Switches
Fiber Channel (2Gb/s)
IBM Storage Fast 200 Storage Server
Internal corporative server
Fast Ethernet 100 Base TX
Switches
Gigabit Ethernet 1000 Base SX

Pic.1. Architecture of VSUE's infrastructure

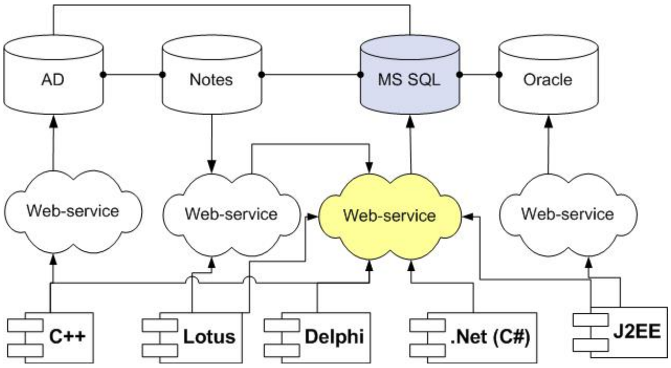
The information environment proposes the management of different systems, network, and communication recourses and the support of regulated, controlled access to the recourses. The combination of login and password are used for access to the recourses.

Attractive solution for creation of user account at University is usage the unified registration system for internal local and external nets. The user will obtain the only login for access to all recourses independently from its location. The system should support the actuality of user accounts. As a result of the registration two accounts for one user will be created. One account is account of the directory service and another account is record in database for external recourses where AD controllers are not available [3].

4. Logical model of IIEU

The data of IIEU are stored in various DBMS - MS SQL Server, Oracle, MySql, Lotus/Notes, or in a directory service (Active Directory, e-Directory, or others LDAP-servers). The same data should be used by various information systems. For example the information about organization structure of VSUE stored in MS SQL Server is used in the almost all information systems of IIEU including .Net and OAS web-applications, Lotus, C++, and MapObject client-server applications. The assessment system developed through Oracle and J2EE uses the data on the students and courses stored in MS SQL Server.

The Information systems are two- or multi-tier applications. Middle tier of the systems either contains web-services or sends/receives messages to/from some web-services (pic.2).



Pic.2. Logical model of IIEU

All systems developed for the information environment have to use the web-services for requests to the databases. It allows minimizing volume of software modification under change of a database, simplifying development of new information systems since work with data has already been implemented in some web-services. Web-service also allows integrating such different databases as MS SQL or Oracle and Lotus/Notes.

The legacy applications can be integrated into environment by steps:

1. data of the legacy applications should be data of the environment or integrated with other data on logic level;
2. registration of the application users and management of their access rights should be carried out through the unified registration and access rights management system;
3. the requests to data of the environment should be implemented through the web-services if it is possible.

Sometimes third-party software has to be integrated into IIEU. Some web-services must be developed for the integration. They receive the inquiries from other systems or web-services and redirection the inquiries to an application server of the third-party software. The application server fulfills request and return data to the web-services. If the software does not have application server the web-services request to database of the software directly.

The information systems have to be registered into the environment. They have name, URL addresses, and user's roles. These systems are called projects. Every project interacts with the web-services which connect to databases, solves some tasks, or sends requests to other web-services. The projects do not know anything about data sources. They may only get or send some data to some web-services, they do not know about database model, data sources, and so on.

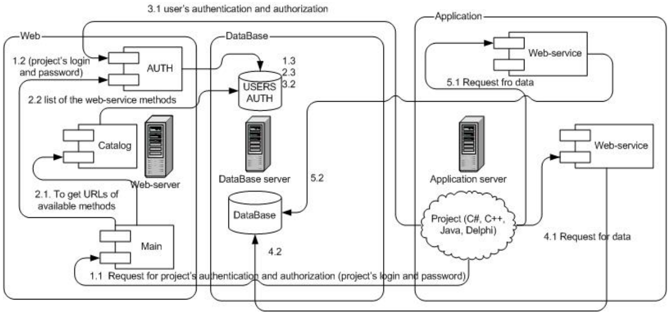
4.1. Control components

There are three basic control web-services in IIEU. There are Main, Catalog, and Auth web-services. (pic.3.). There is Load-Balancing component also.

Main web-service is a router. It receives requests from other web-services and the projects for routing in the information environment. In the first place the projects should request to this web-service for authentication and authorization of itself.

All web-services of IIEU are listed through the intended Catalog web-service. It means that information about web-services is saved into a database. The information contains URLs of all available web-services, parameters of web-service's methods, specific tickets for authentication of the projects in the methods of web-services.

Every application should be authenticated and authorized by every web-service's methods. But some web-service's methods do not need for authentication. They are accessible for anybody. Free web-services allow integrating anything. Authentication and authorization of user accounts are implemented with the intended web-service of authentication and authorization (Auth web-service).


3.1 user's authentication and authorization
Web
Application
DataBase
1.2 (project's login and password)
AUTH
1.3
2.3
3.2
USERS AUTH
Web-service
5.1 Request fro data
2.2 list of the web-service methods
Catalog
Web-server
DataBase server
Application server
Web-service
2.1. To get URLs of available methods
5.2
DataBase
Project (C#, C++, Java, Delphi)
4.1 Request for data
Main
4.2
1.1 Request for project's authentication and authorization (project's login and password)

Pic.3. Control web-services

4.1.1 Main web-service

Every project of IIEU can operate with several web-services. But sometimes the web-services are located on various servers. Besides load balancing algorithm allows using the same web-service located on different servers. The project does not know web-service's URL. In addition access rights to web-service's method for the project can be changed and it need to have actual tickets to authorize in web-service's methods.

All projects know location of the only web-service (Main web-service). After start every project requests to Main control web-service. The project sends project's account to Main web-service for project authentication, authorization, getting URL of other available web-services and tickets.

If the project is a client-server application then it requests to the web-service one time and stores the received data in the process memory. If the project is a web-project then it requests to the main web-service on condition that the common memory of the project is empty. Main web-service sends to the projects all necessary information about the authorized web-services if authentication is fulfilled.

4.1.2. Catalogue web-service

There is web-service supporting catalog of the web-services (Catalog). It analyses WSDL file and makes records of the web-service's metadata in a database. Besides the metadata from WSDL file, the additional information is included into the records (keywords of web-service's domain, links with the objects of databases, all URLs of the web-service).

To support actuality of description Catalog web-service compares WSDL with the web-service description into database. If it was being found discrepancy then records into database was updated and e-mail was sent to the web-service's administration.

Main web-service sends request to Catalog web-service to obtain URLs of the available authorized web-services. Using load balancing algorithm catalog web-service selects appropriate URLs and returns them to the main web-service.

Search of the necessary methods may be carried out by catalog web-service using keywords. Some projects from outside want to get access to public methods. Main web-service may return information about all public methods. It requests the information from the catalog web-service.

4.1.3. Authorization and authentication web-service

Every project has to pass authentication and authorization in IIEU. The project's account has to be checked out. The project has rights to call only some web-service methods. The rights are defined thorough access management system.

Authentication and authorization of project's account are implemented with the intended web-service of authentication and authorization (Auth web-service).

Auth web-service checks login and password of project's accounts. If authentication is fulfilled successfully Auth web-service obtains available methods for the project and actual tickets for every available method.

Every web-service's method has to implement authentication of project and check right the project to use the method. It may be implemented in the web-service's methods by two ways.

One way is check of project's account and its rights through Auth web-service. The project should call web-service's methods with project's login and password. It will be checked the login and password through Auth web-service. The way is not desirable, it is too slow.

Another way is check up specific tickets that the web-service's method and project got if they have been authorized. In present we use the first way but the second one may be more efficient.

Web-services may have public methods. The access to them is defined as open for all project's accounts.

Auth web-service implements user's authentication and authorization also. After authentication and authorization project's account the project sends request to Auth web-service to authenticate and authorize user's account. Auth web-service check user's login and password and get access rights of the user for the project.

Management of access to web-service's methods provides safety of University information environment. On the one hand the approach limits access to data and projects of the environment but on another hand it allows integrating the environment and outside systems. The University's web-services may be used from other Universities, research centers, and organizations.

4.2 Load-balancing component

To raise performance of IIEU load balancing components should be used. They must be located on the every servers of IIEU where web-services operate. An applied web-service may be fulfilled on the several servers. A project of IIEU requests Catalog we-service to get web-service's URL.

There are several algorithms of load balancing.

1. Catalog web-service requests load-balancing components on the all servers where the web-service is located. Load balancing component returns processor usage. Catalog web-service selects the least loaded server and returns URL of suitable web-service. The project connects to the only one web-service. This approach has disadvantage. After the project connects to the web-service located on the least loaded server the other server may get the least loaded. But the project can not connect to other server.

2. Catalog web-service returns to the projects all URS of available web-services. The project may request Load-balancing components to find out the least loaded server and connects to the web-service located on it. The approach demands from the project to know about Load-balancing components. It is disadvantage.

3. In primary case we may not use Load-balancing components. Catalog web-service returns to the project the only URL for every available web-service. URL of web-service is selected one after another.

4.3. Applied web-services

There are many web-services for the work with databases and for calculation. Their quantity and functionality depend on applications and tasks of information environment. The web-services are intended to work with databases or to implement some business logic. The web-services can refer to another web-service to get necessary information. For example, Auth web-service can implement authentication using both Active Directory account and record about user in the MS SQL server.

Applied web-services of IIEU have to fit on IMS or IEEE standards. It allows supporting openness of IIEU.

All applied web-services must be catalogued by its author through Catalog web-service. The author writes name, description of the web-service and gives wsdl-file. In future he writes description of web-service's methods, parameters of the methods, and description of user's data types.

These descriptions are available for all developers of IIEU. Every developer may use the web-service for his applications. But he has to get access rights to the required web-service's methods for his applications. Administrator registers the application in IIEU as a project having its login and password. Administrator sets access rights for the projects to the required web-service's methods if it is possible.

5. Mathematic model of IIEU

We may present IIEU as following

$$G = \{D, W, P, E, U\}, \qquad (1)$$

where $U = \{u_n\}_{n=1}^{N}$ defines users of IIEU, $W = \{w_i\}_{i=1}^{I}$ - defines web-services or other components, $P = \{p_k\}_{k=1}^{K}$ - defines projects of IIEU, $D = \bigcup_{m=1}^{M_S} D_m$ - defines data of IIEU, where D_m is a database, $D = \bigcup_{m=1}^{M_s} d_m$, where d_m is a database object. Connections between the parts of IIEU are defined by

$$E = E^P \cup E^w \cup E^D \cup E^B \cup E^L, \qquad (2).$$

where $E^p = \{e_{ki}^p\}_{k=1,K}^{i=1,I}$ defines connection between p_k project and w_i - web-service.

$E^w = \{e_{ki}^w\}_{k=1,I}^{i=1,I}$ defines connection between w_k и w_i web-services.

$$e^w{}_{ki} = \begin{cases} 1, w_k \; web-service\,calls \;\; methods\,of \;\; w_i \;\; web-service \\ 0, w_k \;\; web-service\,does\,not\,call\,any\,methods\,of \;\; w_i \\ 1, k = i \end{cases}$$

$E^D = \{e_{ki}^D\}_{k=1,M_S}^{i=1,I}$ defines connection between d_k database and w_i web-service. Instead of E^D we may use $E^d = \{e_{kij}^d\}_{k=1,M_S}^{i=1,I,\, j=1,J}$ which shows connection between d_k database object and f_{ij} method of w_i web-service.

$E^B = \{e_{ki}^B\}_{k=1,M_S}^{i=1,M_S}$ indicates replication between D_k and D_i databases.

$E^L = \{e_{ki}^L\}_{k=1,M_S}^{i=1,M_S}$ shows logical connection between D_k and D_i databases. Instead of E^L we may use $E^l = \{e_{ki}^j\}_{k=1,M_S}^{i=1,M_S}$ which indicates logical connection between d_k and d_i database objects.

The connection between the projects and web-services may be detailed as $E^P = \{e_{ki}^P\}_{k=1,K}^{i=1,I} = \{e_{kij}^f\}_{k=1,K}^{i=1,I,\, j=1,I_i}$ what describes connection between p_k projects and f_j method of w_i web-service.

$$e_{kij}^f = \begin{cases} 1, \; f_{ij}\,method\,is\,available\;\,for\;\,p_k\;\,project \\ 0, \; f_{ij}\,method\,is\,unavailable\;\;for\;\,p_k\;\,project \end{cases}$$

In order to control IIEU we have to know the connections between IIEU's objects.

$Q^{PD} = E^{P} \bullet \left(E^{D}\right)^{T} = \left\{q_{ij}\right\}_{i=1,K}^{j=1,M}$ defines connection of first order of p_i project with d_j database.

$Q'^{Fd} = E^{F} \bullet \left(E^{d}\right)^{T} = \left\{q_{ijk}\right\}_{i=1,K,j=1,J}^{k=1,M}$ sets specified connection of the first order of f_{ij} method with d_k database object

$Q''^{Fd} = E^{F} \bullet E^{w} \bullet \left(E^{d}\right)^{T}$ shows connection of second order of f_{ij} method with d_k database object. Thus

$$Q^{Fd} = E^{F} \bullet \prod_{i} E^{w} \bullet \prod_{m} \left(E^{d}\right)^{T} \qquad (3)$$

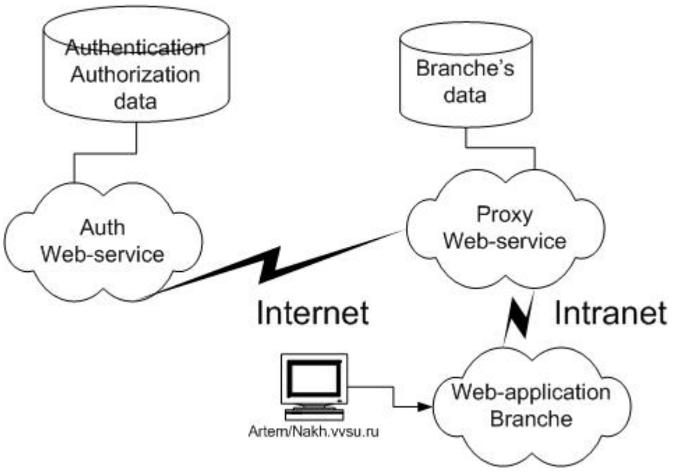
Expression (3) is final connection between f_{ij} method and d_k database object.

If d_k database object has been changed then we can find by means (1) all projects and all web-service that were connected with the object. We have to verify the projects and web-services after modification.

6 Integration with branches

Integration with branches is fulfilled using web-services. Replications between databases can be implemented both DBMS tools and web-services. Data of user's accounts are stored at the main database on head building. Auth web-service carries out authentication and authorization.

Web-applications operate in the branches and use proxy web-service to authenticate and get access rights. Proxy web-service has the same methods as auth web-service. But proxy web-service redirects requests to auth web-service (pic. 4).



Pic. 4. Integration with branches

7. Information quality for management

Quality of information may be determined through five characteristics: validity, actuality, consistency, completeness, and timeliness of delivery of data.

All the characteristics may be provided the integrated information environment.

Validity of data is supported through both organization arrangements and specific procedures of IIEU. Validity of data may be verified by several ways: verification during data input; automatic procedures of data correlation in various reports; getting the same data in various information systems and can compare their. The more users of information systems, the more probability to find out the mistakes.

General data repository allows rising data integration on higher level. It contains a description of object-oriented model of the integrated environment. The description converts data of domain to objects of the environment (database tables, views, web-services, other components, etc.). Software of the integrated environment can be adapted to modified business-processes on basis of communication between domain and objects of environments.

General database repository includes descriptions of peculiar features: links between data and projects, definitions of primary data, and relations between different databases. Actualization and replication procedures should be carried out using the repository.

Data integration is the most important features of IIEU [3]. It provides consistency of data. Completeness is supported the wide range of information systems automating activities of University.

Conclusion

The integrated information environment has been developing at the Vladivostok State University of Economics (VSUE) since 2003. The middleware layer has been developed. There are the three control components. They manage all the web-services of the environment. New projects are developed as multi-tier applications. They use the web-services to carry out business logic and connect to database.

All 45 projects are registered in the IIEU of VSUE by administrator of IIEU. All 15 web-services and their methods are registered in the IIEU using Catalog web-service.

Every project and web-service has account with login and password. The projects and web-services have or do not have access rights to the methods of the web-services. The management of the access rights is carried out using access rights systems.

There are different technologies in the IIEU of VSUE: Active Directory, MS SQL Server, Oracle, MySql, .Net (C++, C#, ASP), Oracle Application Server J2EE, Lotus/Notes Domino, MapObject/MapXtreme, Delphi.

Three control web-services Main, Catalog Auth are used for management of applied web-services and the projects.

The projects automate management of University, education, and information technology. There are projects:

- Enterprise management: financial management, stuff management, building and room management, docflow, planning and report.
- Education management: management of students, studies, classes, educational programs, planning, progress, schedule, etc
- Education: Integrated learning, test, and library systems.
- Information technology management: the unified registration and access rights system, enterprise and educational portal, access to Internet and e-mail.

References

[1] L. Stanton, J. Townsend. How was it for you?- The Oracle Student System and beyond. *Proceedings of the 10-th International Conference of European University Information Systems (ENUS 2004)*. Slovenia 2004. pp. 42-45.

[2] M. Woodfield. Envisioning the Future of Integrated Information Systems in Higher Education. *Proceedings of the 10-th International Conference of European University Information Systems (ENUS 2004)*. Slovenia 2004. pp. 37-41.

[3] C.J. Shakhgeldyan, V.V. Kryukov. Integration of University Information Resources into the Unified Information Environment. P*roceedings of the 10-th International Conference of European University Information Systems (ENUS 2004)*. Slovenia 2004. pp. 321-327

[3] Web Services Description Language (WSDL) 1.1. http://www.w3.org/TR/wsdl