VOL. 11, NO. 17, SEPTEMBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

THE METHOD OF CONSTRUCTING AN INTEGRATED CORPORATE INFORMATION SYSTEM

Vladimir Vyacheslavovich Kopytov[1], Andrey Olegovich Shulgin[1], Sergey Aleksandrovich Fedorov[1], Nikita Georgiyevich Demurchev[1], Andrey Vladimirovich Gritsenko[2] and Vladimir Viktorovich Naumenko[1]
[1]Infocom-S, Stavropol, Russia
[2]North-Caucasus Federal University, Stavropol, Russia
E-Mail: a.shulgin@infocom-s.ru

ABSTRACT

This article describes the integration method of mobile applications with corporate information system (CIS) through the previously proposed interaction architecture based on integration environment. The structure of the integration environment, which includes new approaches to the organization of data storage, conversion and presentation, is described in detail. Also this article represents the algorithm for implementation of CIS mobile applications integration including the proposed structure of integration environment.

Keywords: cloud, cross-platform, information system, mobile application.

INTRODUCTION

The use of mobile applications as a part of CIS technology can significantly extend the technological structure and functions of the system; therefore currently there are many different software solutions for mobile platforms on the market. In addition, there are work in progress on the development of new integration technologies of mobile devices and CIS.

For example, in [1] a universal technique that provides the work of any Enterprise-system users employing any mobile device as a workstation is proposed. This technique can be applied to any CIS, because it uses a universal technology that is independent of the “motherboard” design features, also it is characterized by cross-platform for mobile applications and an important place is given in it to the issues of information security and reducing the computational resources.

In [2] the development of cross-platform software for distributed collection systems, forwarding and data processing of biomedical sensors using mobile devices to cloud servers is described.

In [3] the way to create cross-platform library allowing interacting with different services designed for monitoring the patient’s health indicators through the Internet is considered.

In [4] the problem of geo-distance of medical institutions from each other is described and a technology is offered which allows quick combining diverse applications on multi-platforms based on the SOA concept and Web services with minimal financial and computing resources.

In [5] the software developed by authors allowing generating cross-platform applications for Google Android and Windows Phone for ultimate users without deep penetration into the inner structure of the platform is described.

In [6] a software framework is proposed for the EAS (Enterprise App Store) development and integration, i.e. the web portal through which users can access to business applications, with the concept of mobile knowledge management (mKM). Using this software framework improves flexibility, efficiency, effectiveness and availability of technology.

In [7] the universal recommendations for the development of cross-platform mobile applications for different mobile platforms are offered based on the PhoneGap framework.

In addition, the general lack of existing integration technologies of mobile applications with corporate information systems is that they have mainly “native” character and do not allow providing cross-platform. In such a case, the development of mobile applications is conducted without connection with already described architectural and interface models of information systems, which leads to an increase in development time.

Based on this, the authors of this article proposed a decision [8], the feature of which is unifying similar for different CIS technologies, principles and processes into a single software and technology platform (software framework). However, the CIS architecture integrated with the aid of such platform includes the following building blocks:

- CIS;
- subsystem of CIS extensions including a software package of transforming data and processes which are part of CIS, into data and processes corresponding to the ideology and requirements of cross-platform technology;
- private cloud of computing resources, which is the company’s computing capacity infrastructure designed only for use within the framework of integrated CIS.
- mobile applications that are executable software modules developed by using cross-platform technology for various mobile platforms.

VOL. 11, NO. 17, SEPTEMBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

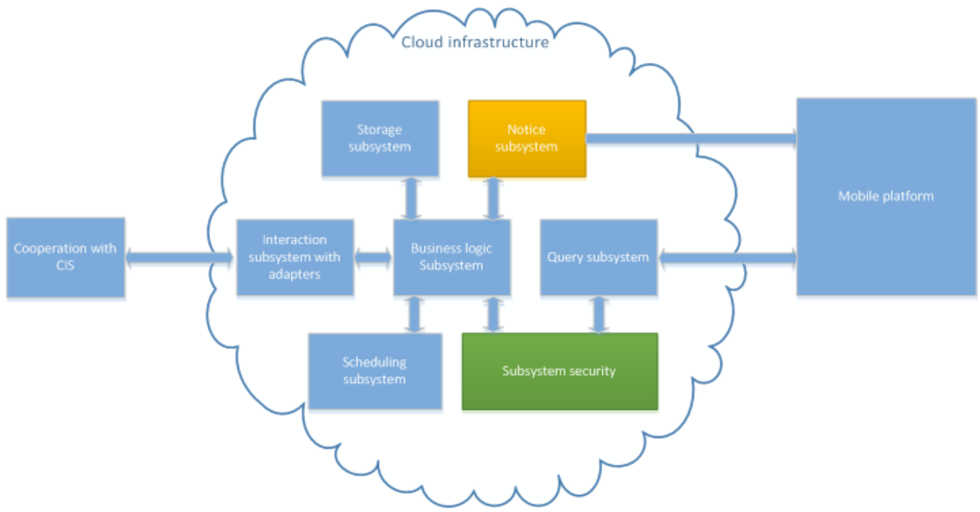

Figure-1. The private cloud structure of CIS mobile application.

Cloud infrastructure is the core of technology platform that provides the integrated CIS functioning and connects the CIS interworking unit with the mobile platform unit into a single entity, using the universal structures described in the data model. This unit contains the majority of the business logic of CIS and mobile application interaction. Figure-1 shows the structure of a private cloud in terms of subsystems.

Within the framework of this article, the method implementing the process of developing mobile application integrated with the CIS in accordance with the proposed architecture was discussed. The proposed method involves the use of different types of models: transforming data model, data storage model and data representation model. These models combine all necessary technologies of procurement, processing and representation of data circulating in the CIS.

In turn, the method of integration with the CIS includes a sequence of actions covering the entire cycle of the mobile application software development and is implemented as an algorithm that uses the capabilities of the previously proposed platform for the deployment of the CIS private cloud, by which the models of storage, transformation and representation of the data of a particular CIS are created, on the basis of which, in turn, the initial code of mobile application for the server and client side is generated automatically.

Before proceeding to describe actions of the proposed algorithm of mobile application integration implementation, we consider the basic principles of the work of proposed integrated platform models.

CONCEPT HEADINGS DATA STORAGE MODEL

Data storage model is a structure consisting of a set of objects and relationships reflecting the vertical hierarchy (see Figure-2).

Each object (O_k) in this model has its own set of parameters (P_1, ..., P_n) as well as their values, some of which are intended for storing data received from the CIS. Schematically, the data storage structure can be represented as a directed graph, whose vertices correspond to objects and object parameters. In this case, each vertex *O* creates a respective subgraph of parameters consisting of peaks P.

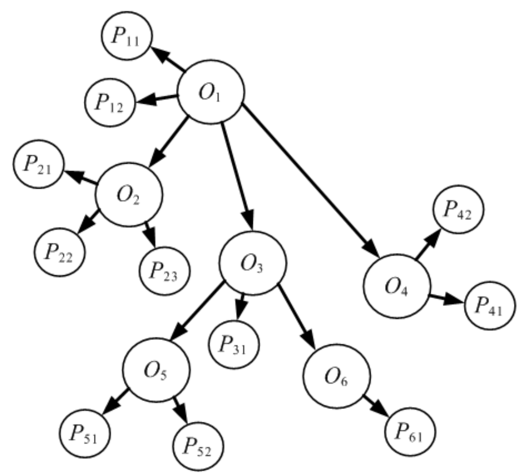

Figure-2. An example of the data storage structure in the private cloud.

The data storage model can be represented in the system as a form of xml file, where an object corresponds to each tag, and parameters tags correspond to the object parameters. Furthermore, the objects parameters can be of three types: a number, a text, or an array.

The proposed data storage model allows describing complex elements, such as tables, which can be grouped into certain structures as well as storing the method parameters used in the subsystem of business-logic of the private cloud.

CONCEPT HEADINGS TRANSFORMING DATA MODEL

The data transformation model is a plurality of A_1, ..., A_m adapters, through which the business-logic

ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

subsystem converts the CIS values of entities S_a in the objects of private cloud data storage model O_k.

Transformations produced by the business-logic subsystem can both receive data from the CIS and make changes in them, and are described by a variety of:

$T(S, A_m, O, P)$,

where S is the entity set requested from the CIS,

A_m is an adapter used to access the CIS entities,

O is the object set in the parameters of which the record received from the CIS database is made,

P is the parameter set passed to the adapter by the business-logic subsystem.

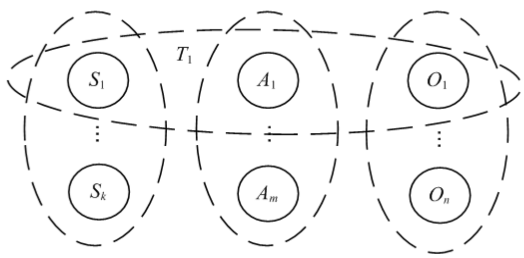

Figure-3. The model of the private cloud data transformation.

Each transformation type is implemented as an adapter customization, which is caused by the corresponding business-logic subsystem method. The proposed architecture supports four types of adapters: interaction with database management system (DBMS) adapters, adapters of the interaction with files, networking adapters, and also security adapters. Consider the implementation details of each of them in this system.

Interaction with database management system adapters (DBMS). This type of adapters is designed to interface with various DBMS types (MSSQLServer, Oracle, MySQL, PostgreSQL, etc.) using the SQL language, understandable for all platforms.

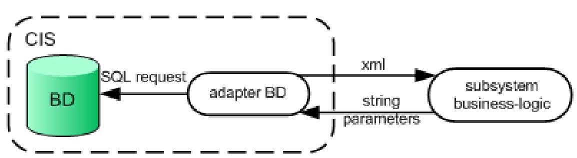

Figure-4. The scheme of interaction with the DBMS adapter.

The content of each interaction with the DBMS adapter includes a set of various configurations, containing the basic types of simple SQL requests such as retrieving, deleting and editing lines, as well as the creation of the table and removing/adding columns. If it is necessary to perform a more complex query to the database, a separate configuration is created for it.

In the case of receiving data from the CIS the interaction with DBMS adapter configurations return values in terms of a data object in XML format, describing the table, with parameters corresponding to columns and containing an array of lines entity.

Adapters of the interaction with files. This adapter type is intended to be used for direct interaction with files stored on servers and CIS workstations or through FTP. Interaction with the CIS via these adapters is performed by reading structured data from a file and/or making changes to it, or to another file.

The scheme of functioning of adapters of the interaction with files would be as follows:

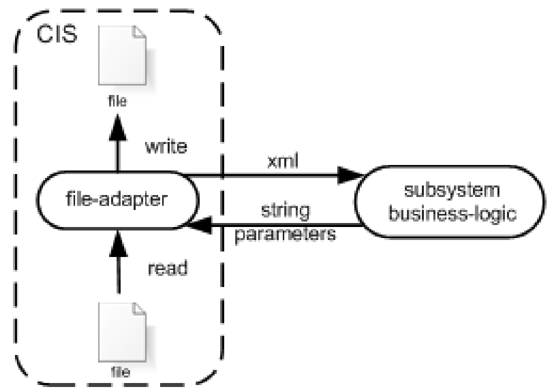

Figure-5. The scheme of interaction with the file adapter.

Networking adapters. This type of adapter is designed to interact with the CIS by various application-layer protocols of OSI model (HTTP, XMPP, OSCAR, TELNET, etc.), and also supports high-interaction techniques: SOAP, XML-RPC, Rest.

The scheme of networking adapter functioning is as follows:

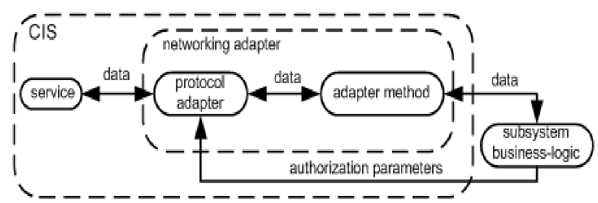

Figure-6. The scheme of the networking adapter functioning.

This adapter includes two subsystems: protocol adapter and interaction technique adapter. Protocol adapter provides connection to the CIS service and user's execution authorization (if required by the protocol). Interaction technique adapter implements exchange of structured data between the business-logic subsystem and the CIS with using either known protocols, or on the basis of protocols used only within a particular CIS. In the second case the development of a special interaction technique adapter is required.

Security adapters. Security adapters are components of the private cloud security subsystem, and their objective is to authorize the mobile application users in the CIS. Within the framework of the proposed solution there are two types of adapters that are used, one of which (the LDAP-adapter) controls connection with the CIS using the LDAP protocol. It can be applied in the case if a

ARPN Journal of Engineering and Applied Sciences

single authorization system based on the LDAP protocol is applied in the CIS, i.e. one account is enough to connect to all CIS subsystems (database, file server, services). It also allows using the same login details both in authorization in the private cloud, and authorization in the CIS. After the successful registration of the user in a private cloud of mobile application, the LDAP-adapter, in turn, uses the user's login and password for the further work with the CIS subsystems.

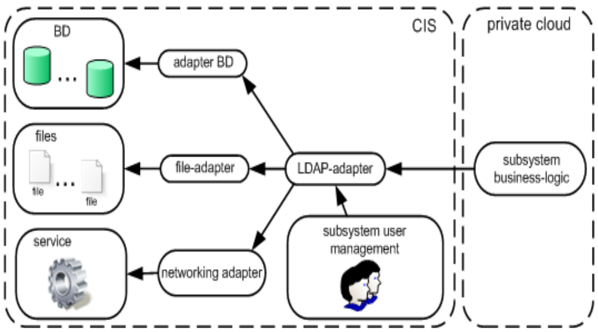

Figure-7. The scheme of the LDAP security adapter functioning.

The second type of security adapters are role adapters, that are designed for the association of the mobile application user with CIS subsystems users, called roles. It is used in the case if the creation of individual accounts is necessary for authorization in the various subsystems in the CIS that are in no way connected with the accounts of other subsystems.

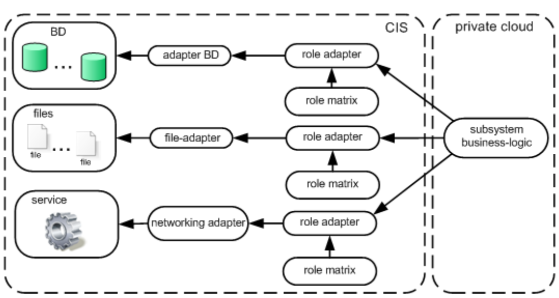

Figure-8. The scheme of the role adapter functioning.

Role adapters store login details of CIS users and make connection to the CIS subsystems in case if the request from the business-logic subsystem is generated by an eligible user in accordance with the compliance matrix. At the same time a separate roles adapter with its own compliance matrix is used to connect to the CIS subsystems.

Considering the fact that the data received from the CIS are presented in the form of objects or object parameters outside of the private cloud data storage model, then, the business-logic subsystem must perform the function of broadcasting the CIS objects into private cloud data objects for the final solution of the data transformation problem. For this each of the methods of this business-logic subsystem must contain the scheme of objects transformation, which is most conveniently represented as a set of the following commands.

Schematically, the transformation of objects can be represented as follows:

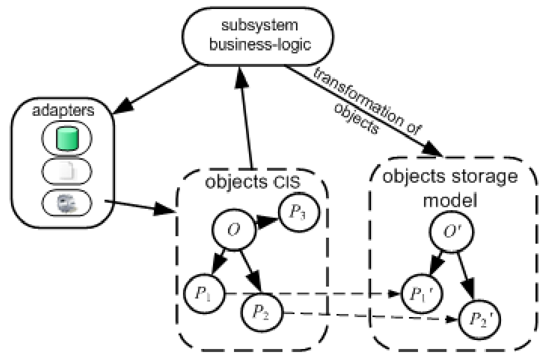

Figure-9. The scheme of converting CIS objects in data objects.

CONCEPT HEADINGS THE DATA REPRESENTATION MODEL

The data representation model includes C_{S1}, ..., C_{Sm} - a variety of private cloud components, implementing methods aimed to convert CIS objects into data storage model objects, as well as C_{C1}, ..., C_{Ck} - user interface components of client-end portion of mobile application that solve the problem of data representation received from the CIS.

Data representation produced by the business-logic subsystem can be described by the following multitude:

$V\,(C_{Sm}, C_{Ck}, O)$,

where C_{Sm} is a component of the private cloud, which implements the appropriate data transformation and representation,

C_{Ck} is the user interface component displaying the data generated by the method (using C_{Sm} component)

O is a variety of objects of data storage model, the parameters of which are used to represent data.

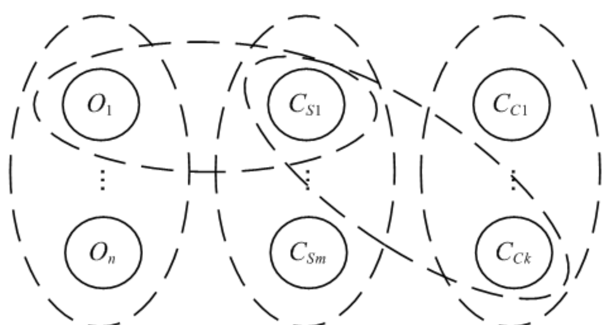

Figure-10. The model of the private cloud data representation.

Methods performing the necessary data transformations and representations are described using the corresponding properties of private cloud components, which are the elements of business process description

ARPN Journal of Engineering and Applied Sciences

language. Therefore, we consider in detail the language constructs and technical features of the implementation of the mobile client application component interaction with the business-logic subsystem methods.

This study employs a language using a series of BPMN specification construction, which includes the following elements:

- component;
- event;
- data flow;
- component with sub-processes;
- condition;
- mathematical function.

Each of the elements of the proposed description language of the business-logic subsystem functioning has a number of attributes that describe behavior of its operation. We examine these elements in detail.

The component describes an information process in which the processing of the user query is implemented, which contains delegated parameters, and making requests to the appropriate adapters using parameter values received from the user. Whereafter the necessary data are distinguished from the responses received from the adapters, out of which a response that returns to the user is formed.

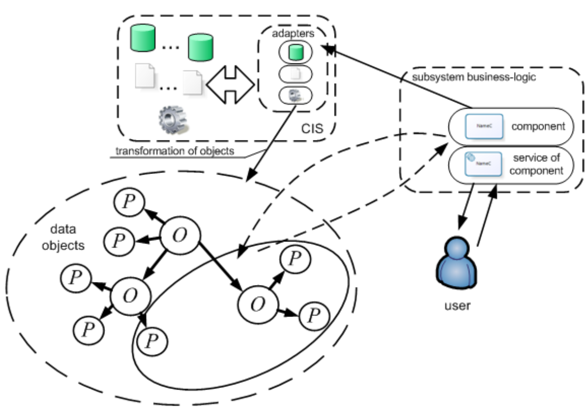

Figure-11. The scheme of the business-logic subsystem functioning.

Consider the algorithm of the business-logic subsystem components functioning. Each component has its own service accessible to the client-end portion of the mobile application and used as part of the formal language of the user interface description. With this service the component receives a structured request, and in accordance with these attributes makes requests to adapters associated with it by configuration mentioned in the parameters. Then, the values of the necessary parameters are selected from the data objects returned by adapters, which are sent to the user in a structured report.

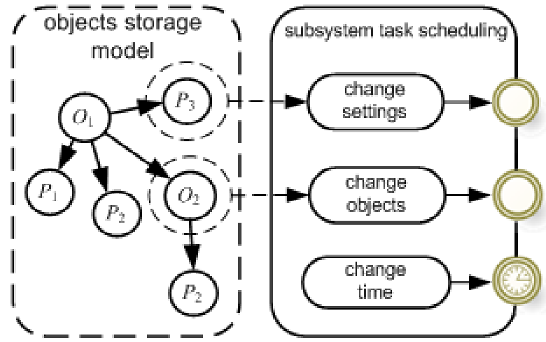

Figure-12. The scheme of the task scheduling subsystem functioning.

Events are intended to make a request to the component without the participation of the client-end portion of the mobile application. Events describe functions of task scheduling subsystem of private cloud and form the request when there are changes in the CIS, which are either reflected in the data storage model, or when time parameters change. The scheme of the task scheduling subsystem functioning is shown in Figure 12.

The data flows are intended for making requests from the events to the components or to broadcast variables from one component to another.

The component with sub-processes (complex component) is intended to describe the complex business-logic subsystem methods performing retrieving data from the CIS, or making changes in accordance with a certain algorithm. The component with sub-processes consists of a scheme (diagram) including simple components and elements describing the behavior of this algorithm.

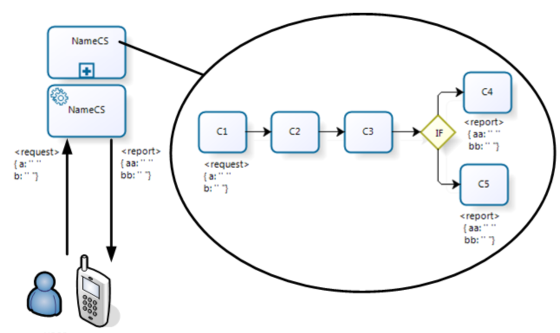

Figure-13. An example of the diagram of a component with sub-processes.

Conditions and mathematical functions are related to the elements describing the behavior of the algorithm functioning of a complex component.

The conditions are intended to describe the algorithm branches. In case the diagram presents the division into parallel streams, then the scheme is written as follows: *C1toC2toC3 ((toC4) and (toC5))*, where *and* is a branching statement, then such a fragment of the diagram corresponds to this scheme:

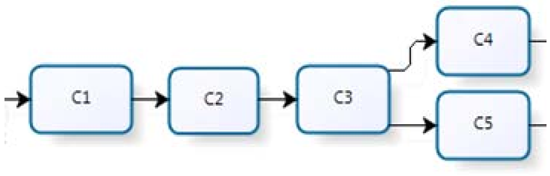

Figure-14. A fragment of the diagram of data flow components (option 1).

If branching occurs after the "Condition" element, then this construct is described by a branching statement or: *C1toC2toC3 ((toC4) or (toC5)):*

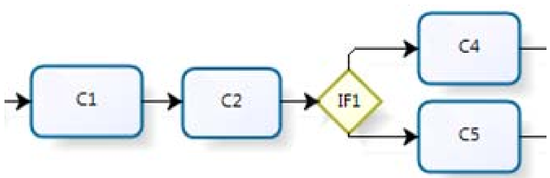

Figure-15. A fragment of the diagram of data flow components (option 2).

The mathematical function is intended to convert the input data in accordance with a certain formula.

Requests sent by the user to a complex component get to the first scheme component.

The output of the algorithm is displayed by the last scheme component or, in the case of the branching by the condition – by one of the last components, the formats of the responses of which must be the same and be congruent with the format of the complex component response (see Figure-14).

Let us proceed to the technical implementation of data representation. For this the formal language of mobile application interface description, including elements of outlining information processes performed by the mobile application and user interface components associated with it, is used. Components that are similar to the components of private cloud are used for the description of information processes:

- component services - provide access to the private cloud components, allowing sending requests to the component and get answers;
- events - are intended to make a request on the events of user interface components;
- data streams - are intended to form request chain and broadcasting variables from one component service to another;
- conditions - are intended to describe algorithm branching of the client device work;
- mathematical function is intended to transform data received from the private cloud components in accordance with the introduced formula;
- data objects - correspond to attributes of user interface components that are inputs for creating the request to the services of a private cloud components.

In addition, the mobile application behavior is described by the sequence of requests from one service component to another (see Figure-16).

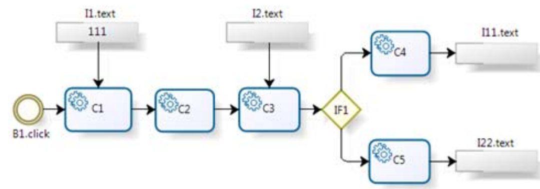

Figure-16. An example of the data flow diagram between the services of private cloud components.

A set of objects, basic for all platforms, is used to describe the user interface, but only the necessary components to work with the CIS components are separated from them. Figure-17 introduces a diagram showing the hierarchy of components of the proposed language model of user interface for CIS mobile application.

This approach will not arise difficulties in transferring the user interface model into the program code for the appropriate platform.

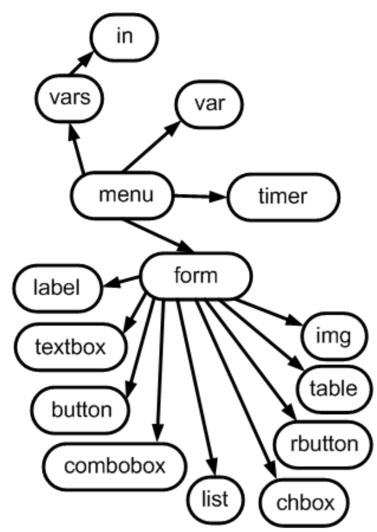

Figure-17. The hierarchy of user interface components for the CIS mobile application.

On the basis of the proposed language model a typical set of user interface components can be determined, on the basis of which the individual functions of the system are described. That is, a specific set of user interface components will correspond to each method of business-logic subsystem that allow performing automatic generation of client-end portion elements of the mobile application. An example of automatic generation of elements of client-end portion of mobile application is shown in Figure-18.

VOL. 11, NO. 17, SEPTEMBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

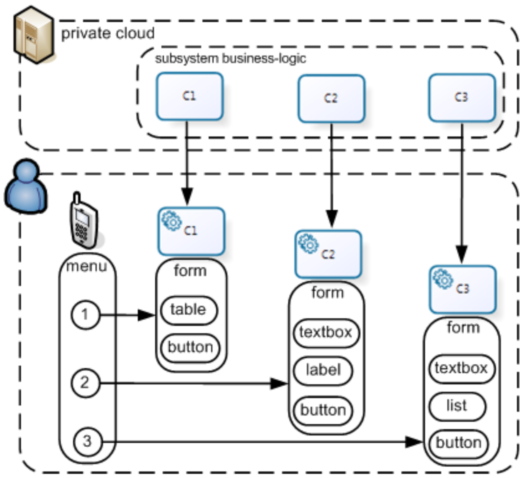

Figure-18. The scheme of automatic generation of elements of client-end portion of mobile application.

CONCEPT HEADINGS THE ALGORITHM OF CIS MOBILE APPLICATION INTEGRATION REALIZATION

Implementation of the CIS integration technique and setting up the mobile application platform is carried out by means of an algorithm that runs in 7 steps. We consider in detail each of them.

Step 1: *CIS functional selection required for the expansion in the mobile application.* At this stage, the selection from a variety of CIS functions is undertaken, which will be involved in a mobile application. In addition, it is revealed which data must be obtained from the CIS, as well as what kind of changes it is necessary to introduce to the CIS using mobile applications.

Step 2: *Selection of the entities that are involved in the implementation of the objective functions.* At this stage it is revealed what kinds of entities in the CIS are employed to store data used in the implementation of the target functions. That is, tables in the database, individual fields, records in files, elements of the structured packets, transferred using networking protocols, are determined. The relationships between the entities are determined, the completeness of the data obtained from the CIS is checked, and also it is checked whether the change of the target data causes the changes to other data entities.

Step 3: *Creating the platform data storage model.* The private cloud data storage model is created on the basis of the entities selected from the CIS, but in this case the following criteria must be guided for its construction.

a) Completeness - the data storage model should describe all CIS target entities, including their parameters and properties.

b) Consistency - the data storage model should not include objects describing simultaneously different entities, as well as objects describing the same entity.

c) Flexibility - the data storage model should be constructed in such a way that the modification implementation in it will not lead to the need to make significant changes in the scheme of information processes of the private cloud.

d) Hierarchy principle - the data storage model should be built according to the hierarchical principle that makes the navigation between objects easier and, thus, reducing the finding of necessary objects in the development of business-logic subsystem methods.

Step 4: *Setting of adapters directed at obtaining data from the CIS.* At this stage an adapter configuration is created through which business-logic subsystem methods obtain the CIS data from the target entities. There already exists a set of typical configurations in the system for each adapter type describing the basic commands for reading and writing of data. In that case if it is necessary to perform more complex request to the adapter, a new configuration is created. Moreover, if there are services using their own data exchange protocol in the CIS, then a separate networking adapter is created for it at this stage.

Step 5: *Creating the platform components.* At this stage, business-logic subsystem methods are created, each of which is realized by means of the respective component. Adapters, through which the reference to the CIS, the request format and response format are carried out, schedule task subsystem events are created that run the business process on a schedule or in case of the CIS change. Simple components that perform functions to read and write the CIS data are specified for complex ones. Then a scheme of successive requests, describing the internal behavior of the complex component, is formed from them.

Step 6: *Creating user interface components*. User interface components are automatically created on the basis of the request format and response format of the private cloud components. That is, for each component a separate entry in the application menu and a form corresponding to this entry are created. Furthermore, user authorization form in the system is automatically generated. If it is necessary to perform processing of data obtained from the CIS on the client side, then a scheme building requests between services of private cloud components is performed.

Step 7: *Setting up user accounts*. At this stage, there is a setting up of user accounts of the private cloud mobile application, at the same time if there is one account in the CIS used for the access to all areas, then registration data are not stored in a private cloud, but immediately forwarded to the CIS through the appropriate security adapters. If different user accounts are applied for the authorization in various CIS subsystems, then their own accounts for each user are created in the private cloud and the adjustment of access permission, based on the matrix of compliance of actors with roles, is performed.

CONCEPT HEADINGS THE DISTRIBUTION OF THE COMPUTATION LOAD IN THE CIS MOBILE APPLICATION INFRASTRUCTURE

The use of the cloud technology requires solving an important task of computational load distribution between the system components. As proposed at the previous stage of the research of the CIS mobile

VOL. 11, NO. 17, SEPTEMBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

application architecture, the private cloud mobile application is a central control subsystem that implements the processing of all requests from the client devices in the CIS. At the same time the cloud computing technology used for its implementation does not limit the scalability of this system. The client devices, in turn, have limited resources and differ greatly from device to device, that should be considered when balancing the load between the server and client-end portion of CIS mobile applications. It is also required that the technology distribution of computational load, proposed in this study, will also implement of the load balance between the private cloud servers.

To achieve the original problems within the framework of the proposed platform the technology of caching the results of user requests is used. The saving the user request results is accomplished on the mobile backend application and in case of no changes introduced in the CIS database, at the next reference the result is given from the cache. In addition, the user request results are stored in the local cache of the client-end portion of mobile applications. If the user re-requests the same data, the application sends only the unique identifier of the response to the server. In a case, if a request result with this ID is stored in the cache server, then a “NotModified” message returns and the application again displays the results from the local cache (see Figure-19) on the mobile device screen. The proposed ways to optimize the load should significantly reduce the power consumption of the CPU time, and reduce the traffic between the server and the client as well. And in this case the volume of consumption of RAM will not grow significantly.

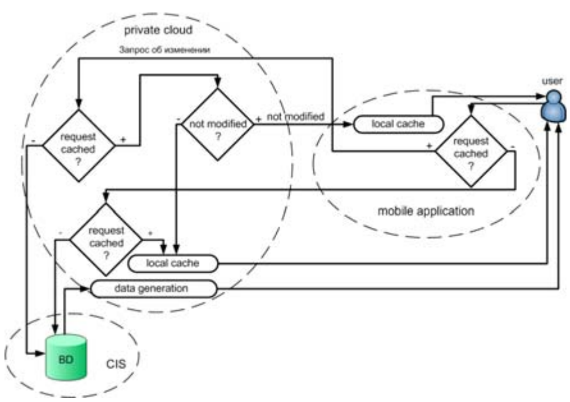

Figure-19. The scheme of request caching of CIS mobile application users.

In order to improve the fault tolerance of backend components of mobile applications an algorithm for the user requests distribution between the private cloud cluster nodes is developed. This algorithm is based on the results of works carried out previously [9] by the group of authors of this article and performs the prediction of the computer system workload for advance reservation of resources required to perform one or another task, initiated by the user request.

DISCUSSIONS

The proposed method of mobile application integration with corporate information system, implemented as an algorithm of building an integration environment for cross-platform mobile applications, allows significantly reducing the time for developing a functional part of the mobile application, through the use of interaction with the CIS adapters, as well as the use of typical configurations reducing the integration development process to adjusting parameters on the basis of which the methods of data obtaining and processing, as well as a finished source code of the client-end portion of mobile application are formed. In the proposed method the “seamless” integration is ensured through the use of special adapters installed on the side of the CIS and performing data receipt and transference of on behalf of an authorized user, which does not require making any changes to the structure of integratable subsystems. One of the main features of the method is the ability to connect integratable subsystems to the adapter using special configurations describing basic functions performed by the user in the CIS, which allows reducing the entire process of mobile application development to the indication of configuration that is necessary to solve one or another task, and to the description of the input and output data as well. In this case the client-end portion of the mobile application can be generated automatically based on the data representation model that contains explanation of the implemented business processes, thus reducing the labor effort required to create an integrated CIS.

The use of methods of computational load distribution between the components of an integrated corporate system will significantly improve the performance of mobile application that will expand the composition of the functions performed by the request of the remote client and ensure the integration of several information systems on the basis of a single private cloud.

CONCLUSIONS

Thus, the use of this mobile application integration method with corporate information system, as well as the distribution of the processing load method between the components of private cloud within the same software and technology platform allows creating the basis for the introduction of high-performance integrated solutions for various systems.

The main advantages of the proposed technology are:

- heterogeneity, contained in the integration with disparate corporate systems ability, i.e., both differing architectures and software platforms;
- cross-platform on the client-end portion of mobile application that provides automatic generation of source code on the basis of the data storage, transformation and presentation model;
- a substantial simplification of developing and integrating mobile applications with the CIS processes;

- reducing the load on the integrated CIS by the use of an intermediary link - a private cloud of mobile application.

In addition, the use of a new method of the computational load distribution within a private cloud of mobile application will greatly reduce the computational resources required for the functioning of the software and technology integration platform with the corporate information system.

The proposed method of mobile application integration with corporate information system can be used to build a variety of integrated solutions for the education sphere (mobile applications of information systems of higher education institutions), transportation (route network and logistics management), security (simultaneous integration with a variety integrated security systems).

However, it should be taken into account that the application of the proposed method is justified at the simultaneous development of mobile applications for multiple mobile platforms, or at the integration of several CIS.

ACKNOWLEDGMENTS

The research was performed in the course of the research scientific work titled “The development of cross-platform mobile application engineering technology with the specified contours of integration for the purpose of increasing functional and resource efficiency of corporate information systems” (unique identifier of applied scientific research is RFMEFI57614X0066) as part of the realization of the Federal Targeted Program for Research and Development in Priority Areas of Development of the Russian Scientific and Technological complex for 2014-2020.

REFERENCES

[1] Platonov, Yu.G. 2011. Development of mobile applications for work with corporate information systems. Informatics Problems. (3): 15-33.

[2] Petrov, A.V., Bolshakov, O.S., Lebedev, A.S., Golubeva, N.Ye. 2013. Method of application templates to improve the mobility of distributed systems for collecting and relaying information from biomedical sensors. Radionics. (5): 7.

[3] Noskov, I.I., Ponomarev, A.A. 2013. Creating a service for work with medical instruments and the development of mobile applications on the Windows Phone platform for monitoring vital exponents of the patient’s health. Siberia. Bulletin of Siberian Science. No 3 (9). pp. 101-106.

[4] Kester, Quist-Aphetsi. 2013. Using SOA with web services for effective integration of hospital information systems via an enterprise service bus. IJREAT International Journal of Research in Engineering and Advanced Technology. 1(2): 6.

[5] Sai Krishna D.R.S., Aparna G., Sravani B., Dr. Sreenivasa Ravi K. 2011. Cross platform compatibility on mobile operating systems. IJCST. 2(4): 462-464.

[6] Roshan, Kumar, Shubhamoy, Dey, Koteswara, Rao G. 2011. Investigation of mobile knowledge management: developing and integrating enterprise app store with existing knowledge management system. Proceedings published by International Journal of Computer Applications® (IJCA) Third Annual Global Business, IT and Management for Economic Development Conference (BIT-MED). pp. 6-11.

[7] Wolf F., Huffstadt K. 2013. Mobile enterprise application development - a cross-platform framework. FHWS SCIENCE JOURNAL. 1(1): 33-40.

[8] Kopytov V.V., Fedorov S.A. 2015. Development of architecture of cross-platform mobile applications integrated environment with corporate information systems. International Scientific and Research Journal. No 7-1 (38). pp. 51-54.

[9] Gritsenko A. 2012. Prediction of DRMS workload by identification of patterns in job submission processes. Journal of International Scientific Publications: Materials, Methods and Technologies. Info Invest Ltd. Bulgaria. Vol. 6. Part 1. pp. 93-100.