Situation-oriented databases: virtual data management

V.V. Mironov
Department of Computer Science and Robotics
Ufa State Aviation Technical University
Ufa, Russia
e-mail: mironov@ugatu.su

A.S. Gusarenko
Department of Computer Science and Robotics
Ufa State Aviation Technical University
Ufa, Russia
e-mail: gusarenko@ugatu.su

N.I. Yusupova
Department of Computer Science and Robotics
Ufa State Aviation Technical University
Ufa, Russia
e-mail: yussupova@ugatu.ac.ru

Abstract[1]

We discuss the development situationally-oriented databases (SODB) methodology of implementing document management based on the built in dynamic model. The purpose - Search science-based ways to effectively use of SODB as an integrator of heterogeneous databases in web applications. It is envisaged to develop and justification of SODB architecture and web-based applications SODB where SODB acts as an intermediate layer between the level of functions integration and the storage level; SODB interfaces with higher and lower levels of web applications in the system, the linguistic, algorithmic plans; SODB software interfaces; confirmation of operability and effectiveness the proposed approach to the research prototype of real web application.

1. Introduction

This project is a development methodology for a new class of document-centric storage - situationally-oriented databases (SODB) governing the documents on the basis of the built in dynamic model, and is a continuation of previous projects supported by the RFBR grants "Electronic documents with embedded dynamic models" and "Methodology situationally-oriented databases: concepts, models, methods and tools".

The purpose of the project - search for science-based ways to effectively use of SODB as an integrator of heterogeneous databases in web applications.

Project tasks:

- development and justification of SODB architecture and architecture of web-based applications SODB where SODB acts as an intermediate layer of honey level integration of functions and the storage level;
- development and justification of SODB interfaces with higher and lower levels of web applications in the system, the linguistic, algorithmic plans;
- development of software interfaces and confirmation SODB operability and effectiveness of the proposed approach to the research prototype of real web application.

2. State of this field of science

The proposed in the project approach is consistent with the basic directions of development of information technologies in the field of building Web applications (Web 2.0 → Web 3.0).

Mash-up

Web 2.0 is an important principle that states "data - this is the next Intel Inside", that is, user and corporate data is a costly commodity. With regard to Web 2.0, this means that people are not attracted to "front-end" Web resource just what is inside, that is qualitative data. An example of this - a common approach today, "mixing» (mash-up term borrowed from the pop music industry, where the "mixing" - a new song, collected from the vocal and instrumental tracks of two or more works of various genres). Mixing technology in the web - a new (and often unexpected) a combination of content from different sources. "Crystals" that form the World Wide Web - blogs, photos, videos, maps, RSS-streams and simple HTML-page - can all be found and "mixed".

This project aims to integrate data from different sources, it is in line with these modern trends.

Proceedings of the 18th international workshop on computer science and information technologies 18thCSIT'2016, Czech Republic, Prague, Kunovice, 2016

Model-driven development

Modern approaches to software development suggest reliance on the model when the model become the main development targets of which is generated code and other objects. The model is considered as an abstract description of software that hides information about certain aspects in order to simplify description. The model can be a source object in the design, if it records information into a form suitable for humans interpretation and processing tools. The model defines the notation and meta-model. It describes meta-model definitions used in the model, and records the information in the form of metadata that can be handled tools. This project assuming the use of built-in dynamic models (for which the class of hierarchical situational models acts as a meta-model) for data management, fully meet the modern trends.

NoSQL

Classic relational databases have difficulties working with large data volume and high load. In this regard, actively working on the creation of fundamentally new solutions, united by a common term NoSQL. NoSQL concept does not deny the SQL, it only tends to solve the problems and challenges that may not work well with SQL - Not Only SQL (rather than no SQL at all). Today NoSQL strongly takes its place in the submission and processing of data. Well coping with the problems of scaling and distribution of data, NoSQL-databases are well suited for working with large data sets. NoSQL-databases are divided into several categories: a simple index like "key-value" rarefied tables Google Table type, document-oriented databases, graph storage and others.

The project in this regard can be seen in the framework of the NoSQL movement, namely, as a kind of document-oriented databases.

XML

Focus on XML meets the world the direction the development of information technologies, which is clearly tendency dominance of XML and related technologies. Currently, XML is the basis of all the leading technologies (RSS, Atom, FOAF, etc.). The volumes of XML-data circulating on the web, every day is rapidly increasing. In this context, it becomes more and more important the search efficiency and storage of large volumes of XML-data. Known two ways to organize the XML-database. The first - databases with XML capabilities (XML-enabled) - involves the use of non-XML data storage and processing models. Typically, in such databases using the relational model, allowing the display to the appropriate XML-model. All major relational databases may be viewed as XML-enabled, such as support RDB-XML mapping. The second option - databases with the "native» XML (Native XML Databases, NXD). The unit of storage in a NXD is always XML-document. There may be several documents, they can be connected with each other, but the structure - always XML. Logic model describes the structure of the XML-document.

JSON

Currently more and more popular in web programmers of utilizing the JSON-markup as simple as an alternative to the XML-markup. Unfolded fight «JSON vs. XML» is unlikely to end someone's complete victory, however, JSON conquered their niche, mainly as a format for the external representation (serialization), the internal software objects. Thus, at present there is no ideal way to store document-oriented content. This justifies the integration of different approaches (as XML, and JSON) within a single integration layer undertaken by this project.

SODB

The concept of situation-oriented databases (SODB) (as the term "SODB" himself) was proposed in 2010 (V.V. Mironov, N.I. Yusupova, G.R. Shakirova) [1]. Since that time, with the support of RFBR studied various aspects of building SODB:

- The possibility of using the database in the construction of Web applications (V.V. Mironov, K.E. Malikova) [2];
- Organization of processing XML-documents SODB based on dynamic DOM-objects (V.V. Mironov, A.S. Gusarenko) [3-8];
- The hierarchical organization of the user interface database (V.V. Kanashin, V.V. Mironov) [9-12];
- Organization of OLAP-based database interface (E.S. Makarova, V.V. Mironov) [13, 14];
- Processing organization SODB documents in JSON format, as well as documents provided by Web services (A.S. Gusarenko, V.V. Mironov) [15, 16, 17].

3. Used scientific reserve

This project is based on the results obtained in the two directions of the study's authors: 1) embedded situational (dynamic) model; 2) XML-technologies in the construction of web applications. Background to the project comprises the following three stages.

Built-in dynamic models

This area was investigated by the project manager from the early 80-ies. as part of research commissioned by companies "Minaviaprom" in relation to the control systems of complex technical objects aerospace. In such systems, monitoring the current state of the dynamic model built on the basis of sensor signals is used to identify the current situation and the formation of appropriate control actions on the technical object. In this approach, the corresponding dynamic models and methods of interpreting them in real time have been designed. Built-in model were presented in a special class of discrete-event model - hierarchical situational models. During the relevant studies have been formulated the basic syntax description language of this class of models

for practical assignment rules and situational control algorithms for technical objects. As a result defended his doctoral (1995) and 5 PhD theses.

Dynamic digital documents

Further investigation showed the possibility of using the idea of embedding finite-dimensional dynamical models in different levels of objects - electronic documents and web-based applications. The electronic documents incorporation of dynamic model allowed to manage user access to certain parts of the document depending on the current situation of its use. In Internet applications built in dynamic models provide dynamic generation of content in accordance with the situation. These studies were used as a base platform, the XML technology: hierarchical situational model modernized under the XML-markup version developed in PHP interpreter to process dynamic models on the side of the web server.

Situation-oriented databases

This period of research related to the integration of dynamic models with databases. Studies of dynamic electronic document it is recognition of the fact that, in essence, we are dealing with a particular database class, which are based on situational model. If the basis of the hierarchical database is a hierarchical model based on relational - relational model, object-oriented - object model, etc. In our case, a database is a finite dynamic (situational) model, which is used for data management in the context of. current situation. For these databases we use the notation SODB data. The author develops one possible methodology for such databases (based on a particular class of finite-dimensional dynamical models - namely, hierarchical situational models of its own design). This project is a continuation of the project "Methodology of situationally-oriented databases: concepts, models, methods, tools", supported by a grant from RFBR. It has developed an approach to the creation of a new, previously unknown class data warehouses, which were called situationally-oriented databases (SODB). SODB understood as a database, which is based on built-in finite-dimensional dynamic model that tracks the current status (which characterize the current situation) and is used for data management in the context of the current situation. The basic technology for the realization of SODB were chosen XML-technologies, which corresponds to the direction of the global development of information technology, particularly in areas related to the Internet.

The idea of building SODB was based on three key ideas: 1) the idea of dynamic DOM-objects; 2) the idea of hierarchical dynamic widgets; 3) The idea of heterogeneous data sources.

Dynamic DOM-objects

The most flexible processing XML-data technology currently is to use DOM (Document Object Model - Document Object Model): the creation of DOM-programmed objects, load them in the XML-data, after which the data in a tree structure are available for transformation. It was given the opportunity to specify the DOM-objects associated with the current state, in the most dynamic model. In the interpretation of the model in accordance with these specifications, the interpreter automatically created DOM-objects are loaded into their XML-data from storage, making them available for the treatment from other elements of the model corresponding to the current state, it saves the changes to the XML-data in the document repository. Thus, the programmer was freed from routine tasks of programming of these processes.

Dynamic hierarchical widgets

Data that provides SODB as a result of external requests, should be formed from the XML-contents of the DOM-objects associated with the current state of the dynamic model. When using SODB as part of the server-side web applications resulting data in the format of HTML / JavaScript / CSS / XML returned to the Web browser client sent the request to form a corresponding image on the screen. It was suggested to specify the pieces of the user interface associated with the current state, in the most dynamic model. During interpretation model in accordance with these specifications interpreter automatically generates widget code by XML-transformation contents corresponding DOM-objects in the context of the current situation. This provides the ability to set the widget hierarchy, when a part of a parent widget may be present nested child widgets. Thereby significantly decreases time-consuming user interface programming.

Heterogeneous data sources

The XML-data loaded in the DOM-objects physically may be stored in various ways: in text files, in archive files in the XML-oriented databases (Native XML), in the tables of relational databases, databases such as "key-value" and so on. It was suggested to provide the opportunity to specify the diverse (heterogeneous) sources of XML-data associated with the current state, in the most dynamic model. During interpretation model in accordance with these specifications interpreter would automatically extracting data from the respective sources and prepared for loading into the appropriate DOM-objects or automatically save the contents of XML-DOM-objects corresponding to the physical storage sources. Thus, the programmer is freed from routine tasks of programming of these processes, providing the flexibility to transition from one way of physical data storage to another.

4. The methods and approaches

The fundamental idea of the project - is the idea of virtual SODB, embedded in a Web application that manipulates virtual documents that are displayed in real data in various storage media.

We refer to such a virtual SODB, since in this case it does not itself manages data storage in the physical layer, and uses for this capability ready traditional data storage media.

The idea of a virtual repository that integrates heterogeneous databases known. SODB However, for such an approach is new. Its implementation requires appropriate research studies and practical tests.

The proposed web application architecture with integrated SODB includes 4 levels (layers):

1. Level of calling application logic functions - provided by means of a Web application programming environment;
2. The level of performance logic functions - SODB provided at the request of a higher level by monitoring the current state the built in dynamic model and perform actions associated with the current state;
3. Level manipulation of virtual instruments - SODB provided upon request from the parent level (from the dynamic model SODB) by displaying a virtual documents on real data in a variety of storage media (file systems, databases of different types, web services, etc.);
4. Level manipulation of stored data - is provided by means of storage media (file systems, databases of different types, web services, etc.).

Thus SODB implements the 2nd and 3rd levels. Thus, the virtual SODB acts as an integration layer, designed for the collection putting together documents from a variety of physical storage, and managed in accordance with the logic implemented in the internal dynamic model (Model-driven approach).

This approach requires the study and elaboration of interfaces SODB interaction with both the higher level to allow to contact SODB of application code, as well as a lower level in order to ensure interaction with the used storage media. Said study affects the system, linguistic (language) and algorithmic aspects.

Stages of solving the problems

Step 1: Design and rationale SODB architecture and architecture of web-based applications SODB where SODB acts as an intermediate layer of honey level integration of functions and the storage level. It is supposed to develop such SODB architecture, which allows you to embed SODB in the Web application and use ready-made media storage as a virtual document storage.

Step 2. Design and rationale SODB interfaces with higher and lower levels of web applications in the system, the linguistic, the algorithmic plan. It is supposed to develop interface solutions to send to the built-in SODB of web application code, as well as to interact with a virtual repository of documents with the specifications of the dynamic model SODB.

Step 3: Software development SODB interfaces and confirmation of efficiency and effectiveness of the proposed approach to the research prototype of the real web application. It is supposed to implement the previously-developed architecture and interface solutions in the form of program code in the language of PHP, debug and test on a real Web application.

Detailing of the first stage

1. Architectural study of logic level applications call functions within the web application software environment;
2. Architectural study of the level of implementation of logic functions within SODB by tracking the current state the built in dynamic model and perform actions associated with the current state;
3. Architectural study of the level of manipulating virtual documents at the request of the dynamic model SODB by displaying virtual documents on real data in a variety of storage media (file systems, databases of different types, web services, etc.);
4. The architectural study of manipulating the level of the stored data by displaying a virtual repository of documents for real storage environment.

5. Expected project results

1. Original SODB architecture and web-based applications SODB, characterized in that SODB acts as an intermediate layer of honey level integration of functions and the storage level;
2. Original SODB interfaces with higher and lower levels of web application system characterized by the construction, linguistic and algorithmic support;
3. Original Software interfaces SODB, based on the results obtained, and confirmation of efficiency and effectiveness of the proposed approach to the research prototype of real web application.

The components and connections that make up the architecture being developed should provide the following functionality query processing client applications:

1. Receive requests of the client applications that enable or identify a registered user, or work with the user as anonymous;
2. Enable dynamic model HSM in accordance with the request of the client application;
3. connect the stored memory CSM current status for the registered user or session memory for the current state of an anonymous user;

4. perform basic interpretation of a passage connected HSM connected to CSM in order to correct the current state of the dynamic model;

5. perform additional passes interpretation of the purpose of processing virtual documents stipulated in HSM for current conditions;

6. access the virtual storage of documents for access to virtual documents necessary for processing by the upload / download content DPO document processing facilities;

7. redirect a virtual repository of documents on the corresponding real data store to retrieve / write data;

8. fragments to form a response to the original request by transformation of documents loaded in the DPO;

9. form a response to the original request by gathering putting together fragments of a response and send the response to the user.

Relevance of the project due to the great interest shown in the present time to the non-relational data (movement NoSQL), in particular – to document centric data, and the methods and means to manage them effectively.

In scientific terms, the project will enhance the understanding of how to effectively integrate disparate data document centric and manage a high level of abstraction (based on the internal dynamic model) in the implementation of modern web applications.

In technical terms, the project results will bypass the obstacles to the creation of a full-scale SODB, consisting in the fact that the practical implementation from ground up to grade database management system (DBMS) - an extremely time-consuming and lengthy task. It proposed the idea of a virtual SODB - an elegant solution to this problem in the form of add-on known database and other storage systems.

6. Conclusion

The main results of research are:

- SODB represents a new approach to building data-processing applications, based on the principles of management according to the model (Model Driven Approach).
- At the SODB core lies built a dynamic model with a finite number of states in the form of transition graphs of states (situations), which reflects the business process logic, data serviced by the application.
- Data management is performed by interpreting the built in dynamic model of tracking the current status of and access to data associated with the current conditions.
- To set the built-in dynamic models developed by the HSM declarative language, which allows to describe in graphic or text form hierarchy of sub-models that contain multiple conditions for which, in turn, you can specify the state transitions, promotions, and other sub-model.
- For jobs in the HSM document processing specifications stipulated document processing elements that generate DOM- or Smarty-objects. elements of sources are available for loading documents into objects in HSM.
- To specify the processing of uploaded documents and upload the results to HSM provides elements of receivers. XML-documents are processed by XSL-transformation, and JSON-documents - processing templates by the compiler.
- Access to documents in the form of heterogeneous virtual documents XML or JSON integration layer is provided by a doc-elements that specify the format conversion of documents "on the fly" with regard to the specifics of the actual storage.
- Further development SODB under the consideration of the concept involves expanding the range of document formats handled by the introduction, in the first place, new types of document processing elements, and secondly, - new types of virtual instruments.

Acknowledgments

This project is supported by RFBR, grant 16-07-00239.

References

1. Mironov V. V., Yusupova N. I., Shakirova G. R. “Situation-oriented databases: the concept, architecture, XML-implementation”. *Vestnik UGATU*. 2010; 37: 233–244.
2. Mironov V. V., Malikova K. “Internet-applications based on built dynamic models: the user interface controls”. *Vestnik UGATU*. 2010; 40:170–175.
3. Mironov V. V., Gusarenko A. S. “Situation-oriented databases: The concept of XML-data management based on the dynamic DOM-objects”. *Vestnik UGATU*. 2012; 48:159–172.
4. Mironov V. V., Gusarenko A. S. “Dynamic the DOM-objects in the situation-oriented databases: linguistic and algorithmic support of data sources”. *Vestnik UGATU*. 2012; 51:167–176.
5. Kanashin V. V., Mironov V. V. “Hierarchical widgets: the organization of the user interface in web applications based on situation-oriented databases”. *Vestnik UGATU*. 2013; 55:138–149.
6. Kanashin V. V., Mironov V. V. “Hierarchical widgets: data input and user control in web

applications based on situation-oriented databases". *Vestnik UGATU*. 2013; 58:166–176.

7. Kanashin V. V., Mironov V. V. "Hierarchical widgets: Algorithms control the user data in web applications based on situation-oriented databases" *Vestnik UGATU*. 2014; 62:204–213.
8. Kanashin V. V., Mironov V. V. "Hierarchical widgets: the experience of the application in a web application based on situation-oriented database". *Vestnik UGATU*. 2014; 63:185–196.
9. Makarova E. S., Mironov V. V. "Проектирование концептуальной модели данных для задач Web-OLAP на основе ситуационно-ориентированной базы данных". *Vestnik UGATU*. 2012; 51:177–188.
10. Makarova E. S., Mironov V. V. "Design of a conceptual data model for Web-OLAP tasks based on situation-oriented database". *Vestnik UGATU*. 2013; 58:150–165.
11. Gusarenko A. S., Mironov V. V. "Smarty-objects: using of heterogeneous sources in situation-oriented databases". *Vestnik UGATU*. 2014; 63:242–252.
12. Gusarenko A. S., Mironov V. V. "Using of RESTful-services in situation-oriented databases". *Vestnik UGATU*. 2015; 67:204–211.
13. Mironov V. V., Gusarenko A. S., "Dimetriev R. R., Sarvarov M. R. Creating personalized documents based on situation-oriented database". *Vestnik UGATU*. 2014; 65:191–197.
14. Mironov V. V., Yusupova N. I., Gusarenko A. S. "Situation-oriented databases: current status and prospects for research". *Vestnik UGATU*. 2015; 68: 188–199.
15. Mironov V. V., Yusupova N. I., Shakirova G. R. "Hierarchical data models: concepts and implementation based on XML". Mashinostroenie, Moscow, 2011.
16. Gusarenko A. S. "Models for creation documents in Office Open XML format based on situation-oriented database". *Applied Informatics*. 2015; V. 10, 3:62–75.
17. Gusarenko A. S. Improvement of situation-oriented database model for interaction with MySQL // Journal of Instrument Engineering. 2016; V. 59. 5:355–363.