

УДК 004.65

ГЕТЕРОГЕННЫЕ ИСТОЧНИКИ ДОКУМЕНТОВ В СИТУАЦИОННО-ОРИЕНТИРОВАННЫХ БАЗАХ ДАННЫХ

А. С. Гусаренко¹, В. В. Миронов²

¹artyomgusarenko@gmail.com, ²mironov@list.ru

ФГБОУ ВПО «Уфимский государственный авиационный технический университет» (УГАТУ)

Поступила в редакцию 7 апреля 2015 г.

Аннотация. Рассматривается текущий уровень развития ситуационно-ориентированных баз данных на концептуальном уровне. Дается обзор СОБД и приложений на базе СОБД с позиции динамических систем. Задаются направления развития исследований по проекту ситуационно-ориентированных баз данных. Приводится архитектура СОБД по компонентам, а структура приложений по уровням сервера приложений и сервера баз данных. Сравнивается подход СОБД с уже существующим реляционным подходом. На имеющееся обеспечение в реляционных базах данных даются разработанные подсистемы СОБД. Рассматривается архитектура практической части СОБД, такие компоненты как буферные объекты, память текущего состояния, хранилище документов, иерархическая ситуационная модель. Приводятся используемые виды получаемого результата и структура запросов к системе управления базой данных. Излагается концепция СОБД, описывается структура, интерпретация модели, обработка документов на практике. Описывается содержимое буферных объектов, подмножества источников и приемников данных в состояниях СОБД.

Ключевые слова: веб-приложение; база данных; динамическая модель; NoSQL; RESTful-сервисы; XML; JSON; DOM; Smarty; PHP.

ВВЕДЕНИЕ

В настоящее время активно развиваются документо-ориентированные системы и веб-приложения, предоставляющие сервисы для своих клиентов в качестве удаленных ресурсов для решения задач обработки и генерирования документов. В качестве информационного обеспечения используются реляционный и не реляционный – NoSQL – подходы совместно, для создания платформы веб-сервисов и веб-приложений. Бурный рост нереляционных баз данных связан с тем, что не все задачи «закрываются» реляционным направлением, для этого используются различного вида NoSQL-базы данных [1, 2].

На текущий момент развивается проект «Ситуационно-ориентированных баз данных» в русле NoSQL, ориентированный на работу с различными видами источников. В этом направлении уже проведен ряд работ по созданию обеспечения [3–6], в данной работе рассматри-

вается теоретическая линия проработки подхода, опирающегося на динамические модели и их интерпретацию, в основе базы данных приложения. Этот подход ориентируется на использование при построении документо-ориентированных веб-систем, работающих с источниками, на основе которых генерируются персонализированные документы [7–21].

Рассматривая теоретическое описание концепции обработки документов в ситуационно-ориентированных базах данных (СОБД) [13–16] на основе буферных объектов, упомянутых в работах [17–19], для решения задачи построения документо-ориентированных веб-приложений, следует отметить, что уже сейчас существуют виды обеспечения СОБД за исключением обобщенной теоретической платформы для такого вида баз данных NoSQL. Исходя из возможностей использования СОБД для построения документо-ориентированных приложений, которые в настоящее время распространены, требуется документировать особенности архитектуры и методы функционирования составных частей в СОБД. Такое представление помогает специалистам увидеть, как построена база

данных, наметить дальнейшие направления для развития в плане реализации и расширения функциональности. Для наглядности и облегчения понимания требуется сравнить подход СОБД с уже существующими подходами построения реляционных баз данных. Для сравнения с подходом СОБД приводится не только архитектура реляционных баз данных, но и структуры запросов к СУБД (СУБД – система управления базами данных).

В предшествующих работах [17] обсуждались вопросы обработки внешних RESTful-источников данных для создания персонализированных документов на веб-сервере из источников хранилища приложения на базе СОБД. В этой статье рассматривается концептуальная составляющая проекта СОБД, архитектура, уровни приложения на базе СОБД, внутренние компоненты базы данных, интерпретатор и динамическая модель базы данных. Проводится аналогия с реляционным подходом.

1. СОБД КАК ДИНАМИЧЕСКАЯ СИСТЕМА

Статья продолжает развивать концепцию проекта СОБД, предложенную в работах [7–9]. Ранее обсуждались вопросы обеспечения для СОБД, в то время как теоретические аспекты построения и функционирования СОБД не были рассмотрены подробно. Проработка в теоретическом плане помогает использовать алгоритмы, методы, разрабатывать обеспечение и средства СОБД, а на этой базе строить документо-ориентированные системы.

Динамическая система СОБД. Практическая часть СОБД базируется на обработке документов, эти документы типа XML и JSON обрабатывались с помощью технологии динамических DOM-объектов и Smarty-объектов. Подобных объектов в базе данных приложения может создаваться множество, объекты с таким поведением будем называть буферными. Ситуационная база данных представляет собой динамическую систему (рис. 1), где есть входной параметр X_t и выходной Y_t , а система реализует функцию F по обработке данных.

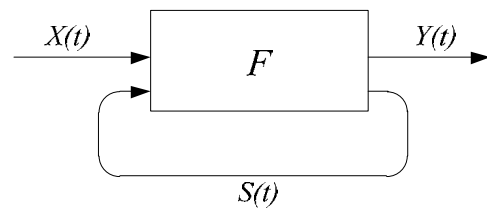


Рис. 1. Схема динамической системы СОБД

В качестве входного параметра может быть запрос на получение или изменение данных, выходной параметр представляет собой результирующие данные. Внутреннее исполнение СОБД представляется как динамическая модель с состояниями приложения (рис. 2), в состояниях обрабатываются ассоциированные данные D_N , при смене текущего состояния данные загружаются, очищаются, фильтруются. Переходы между состояниями модели S_N осуществляется в ходе интерпретации динамической модели с помощью интерпретатора СОБД.

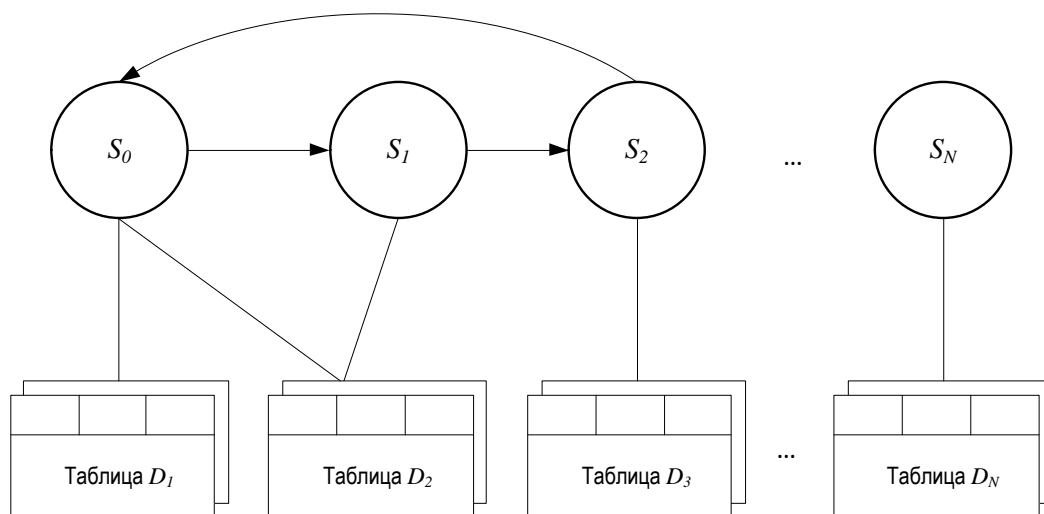


Рис. 2. Диаграмма смены состояний веб-приложения и обработки данных в контексте состояний

СОБД и реляционные СУБД. Сравнивая два подхода, выделяют два уровня информационных систем – уровень приложения и уровень базы данных (рис. 3). В приложение поступают запросы пользователей, результаты функции или параметры, которые в дальнейшем формируют суть запроса системы управления базой данных. Сформированные запросы операций с данными – CRUD, с помощью метаязыка и параметров содержимое базы данных поддерживается в актуальном состоянии. Выходные данные возвращаются в приложение, и пользователь их получает в ходе работы. В СУБД реализован буферный кэш (рис. 4), с которым работает каждый пользователь, подключенный к БД. Операции в базе данных проводятся над таблицами, кроме того предусмотрены специальные объекты в базе данных – хранимые процедуры. Хранимые процедуры на уровне базы данных используются как инструкции на языке высокого уровня, которые хранятся на сервере и компилируются для выполнения стандартных операций над данными.

В СОБД есть похожее по функциональности обеспечение. Отличия от реляционного подхода заключаются, прежде всего, в том, что в качестве входного параметра используются специальные объекты динамической модели и память текущего состояния. Запрос на управления данными формируется, исходя из спецификаций модели. Модель с состояниями и субмоделями автоматически интерпретируется специальной оболочкой – интерпретатором динамических моделей.

Результат обработки данных передается в приложение на базе СОБД. В составе динамической модели задается модель буферных объектов, для обработки источников данных, которые получаются из хранилища данных и внешних сервисов (рис. 5.) и др. программных источников.

2. МОДЕЛЬ СОБД И БУФЕРНЫЕ ОБЪЕКТЫ

Ориентируясь на модель СОБД, требуется расширить описание источника данных, добавив функциональность задействования внешних сервисов. Ранее для задания документов в модели СОБД использовались элементы `doc` и `xml`, примеры их использования разобраны в работах [7–9]. Эти документы загружались в ходе интерпретации, затем использовались в источниках для формирования результирующих DOM-объектов. Спецификации документов из RESTful-сервисов в модели указываются в строке описания `doc`. При этом оба типа документов XML и JSON возможно задавать с помощью деклараций, указывая тип сервиса, например, `type="XMLService"` или `type="JSONService"`. Для каждого сервиса в зависимости от вендора (поставщика, производителя) требуется своя функциональность подключения и задействования, поэтому в модели для передачи параметров предусматриваются параметры `prmt`, указываемые через пробельные символы `prmt = "параметр1 параметр2 ..."`.

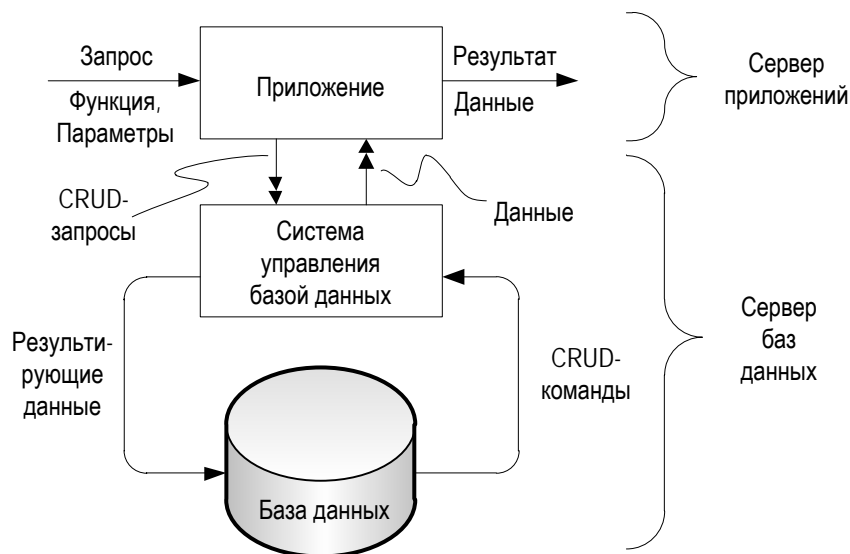


Рис. 3. Диаграмма использования реляционной БД в веб-приложении по уровню сервера приложений и сервера баз данных

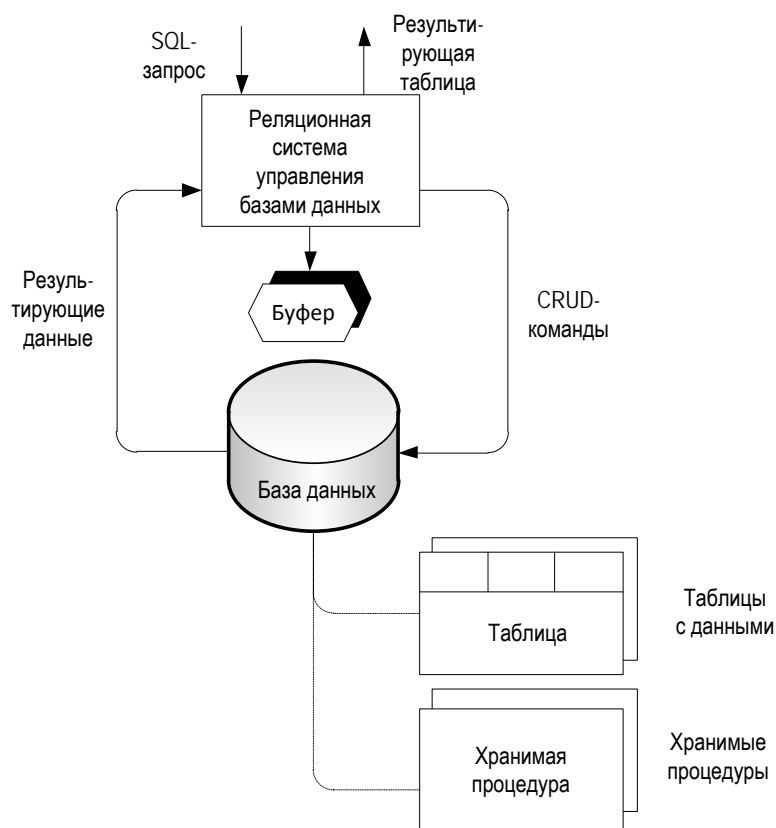


Рис. 4. Диаграмма реляционной базы данных с внутренними инструментами на уровне сервера баз данных

Таким образом, после того как интерпретатор из источника *src = "OJSXMLJournalReport"* получает на интерпретацию описание документа *OJSXMLJournalReport* типа *type = "XMLService"* с параметрами, он собирает URL-запрос к серверу с сервисом от вендора OJS, который находится по адресу, указанному в базовом URL *baseurl*. Базовым он назван потому, что имеет постоянный адрес и к нему добавляются конструкции в соответствии со спецификацией REST API от вендора сервиса OJS, установленного на домене *http://journal.ugatu.ac.ru/*. Эти конструкции собираются интерпретатором с использованием параметров *prmt* модели, как правило, добавляются в конец базового URL. Аналогичным образом завершается сборка запроса для сервиса *OJSJSONJournalReport* для получения отчета в JSON-формате. Описание документов ассоциируется с источниками конкретного типа, например, в Smarty-объектах задаются источники JSON от сервисов *JSONService*, тогда как в DOM-объектах зада-

ются источники от *XMLService* в соответствии с зависимостью от реализованной технологии обработки получаемых данных. При оснащении модели такой функциональностью обеспечивается выход на декларативный модельно-ориентированный [22–33] подход работы с внешними RESTful-сервисами, при котором снижается объем программирования и расширяется коллекция источников данных СОБД.

Запросы HTTP в модели. Данный подход отличается от модели прежнего уровня абстракции главным образом тем, что запросы осуществляются не только к интерпретатору пользователем, но и интерпретатор самостоятельно обрабатывая модель, собирает и выполняет запросы к внешним RESTful-сервисам, получая отчеты в двух форматах, а затем загружает эти отчеты в динамические объекты с последующей обработкой в интересах пользователя. Другим немаловажным отличием модели является использование REST-запросов, с помощью которых пополняются динамические объекты:

- у DOM-элементов и Smarty-элементов в состояниях динамической модели заданы параметры для компилирования запроса GET или POST. Эти запросы часто используются в веб-приложениях, запрос POST отправляет данные по указанному URI, при получении параметров сервер распознает, каким образом следует обработать данные в контексте адреса. Запрос может быть адресован процессу, который принимает данные, к шлюзу или отдельному объекту;

- у DOM-элементов и Smarty-элементов появилась возможность компилировать запросы PUT. Отличием запроса PUT от POST является то, что он загружает файл или ресурс по специально заданному в модели URI, при этом, если файл или ресурс уже существуют, то по заданному URI производится их замена. В том случае, если не существует URI или файла, происходит их создание. Запрос идентифицирует сущность, включенную в запрос, и пользователю известно, для чего предназначен URI. У сервера нет возможности применить запрос к другому ресурсу, если это необходимо, сервер уве-

домляет пользователя о временном перенаправлении (код ответа HTTP 301), при этом пользователь имеет право самостоятельно принять решение о перенаправлении запроса.

REST-метод cURL. Для создания запросов к сторонним сервисам на платформе PHP используется специальное расширение cURL. Расширение позволяет взаимодействовать со многими внешними сервисами по различным протоколам, в том числе и HTTP.

Чтобы создать запрос к сервису, требуется запрограммировать на языке PHP запрос с указанием URL и других пользовательских опций специфики обработки ответа. Указать тип запроса HTTP, например, это может быть GET-или POST-запрос. Запросы выполняются в сеансовом режиме, также требуется выбрать метод обработки получаемого XML-ответа; для DOM-объектов такими методами являются XMLReader или DOM [10–15], для JSON-ответа – динамические Smarty-объекты.

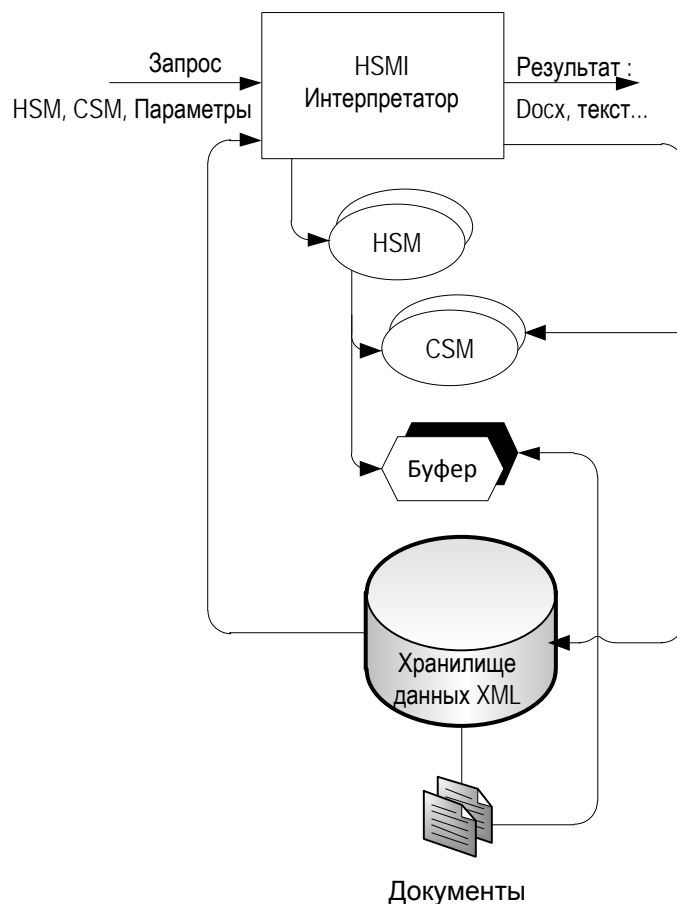


Рис. 5. Диаграмма внутренней архитектуры СОБД на уровне сервера приложений и сервера баз данных

ЗАКЛЮЧЕНИЕ

В статье рассмотрена задача использования сторонних RESTful-сервисов в качестве внешних источников данных для приложений на основе СОБД. Таким образом, приложение, используя технологии динамических DOM- и Smarty-объектов, получает возможность не хранить данные локально, а получать их из сервисов сети Интернет, при этом генерировать на основе полученных данных документы Office Open XML. Эта технология пояснена на примере сервисов XML и JSON, из которых с помощью запросов модели принимаются и обрабатываются данные. В результате модель оснащена инструментами задания операций по подключению к внешним сервисам и запрашиванию персональных данных. А именно, источники данных динамических объектов автоматически подключают REST-инструментарий запросов. Интерпретатор динамической модели СОБД обрабатывает спецификации в ассоциированных с состояниями динамической модели метаязыковых конструкциях работы с XML- и JSON-сервисами. Конечный результат пользователь получает в виде Office Open XML документа, выгружаемого через браузер на клиентский персональный компьютер.

Данная модель отличается от уже реализованной модели более низкого уровня абстракции тем, что:

1) элементы-источники расширены описанием внешних RESTful-сервисов, и для каждого источника DOM-объекта в элементе документа появилась возможность задать описание задействования XML-сервиса, а для каждого Smarty-объекта есть возможность задать внешний JSON-сервис;

2) в ходе процесса интерпретации модели запросы к сторонним сервисам собираются и выполняются автоматически без ручного программирования создания и выполнения REST-запросов, далее полученный ответ загружается в готовый динамический объект;

3) на основе полученных данных от внешних сервисов появилась возможность генерирования Office Open XML документов с использованием технологии динамических DOM-объектов и Smarty-объектов согласно декларативным спецификациям в модели СОБД;

4) модель реализует два метода выполнения REST-запросов к внешним сервисам `cURL` и `file_get_contents`;

5) Модель реализует REST API для взаимодействия с вендором Open Journal Systems;

Согласно предложениям установлено, что создавать приложения на базе СОБД удобнее,

используя спецификации модели, которые затем автоматически обрабатываются интерпретатором моделей СОБД, и создавать Office Open XML заготовки с учетом сведений из внешних RESTful-сервисов.

Последующая работа будет посвящена реализации предложенной модели на платформе PHP для генерирования Office Open XML документа автоматического содержания с задействованием сторонних сервисов `OJS GUSAR-GatewayPlugin` и `RestPlugin`.

СПИСОК ЛИТЕРАТУРЫ

1. **Strauch C., Kriha W.** NoSQL databases. [Электронный ресурс]. URL: <http://www.christof-strauch.de/nosql dbs.pdf> (дата обращения 07.11.2012). [[C. Strauch and W. Kriha (2012, Nov. 11). *NoSQL databases* [Online]. Available: <http://www.christof-strauch.de/nosql dbs.pdf>]]

2. **Фаулер М., Садаладж П. Дж.** NoSQL: новая методология разработки нереляционных баз данных. М.: Вильямс, 2013. 192 с. [[P. J. Sadalage, M. Fowler, *NoSQL Distilled: A Brief Guid to the Emerging World of Polyglot Persistence*. Addison-Wesley, 2013.]]

3. **Миронов В. В., Шакирова Г. Р.** Концепция динамических XML-документов // Вестник УГАТУ. 2006. Т. 8, № 5. С. 58–63. [[V. V. Mironov, G. R. Shakirova, "Concept of dynamic XML documents," (in Russian), *Vestnik UGATU*, vol. 8, no. 5, pp. 58–63, 2006.]]

4. **Миронов В. В., Шакирова Г. Р.** Интерпретация XML-документов со встроенной динамической моделью // Вестник УГАТУ. 2007. Т. 9. № 2. С. 88–97. [[V. V. Mironov, G. R. Shakirova, "Interpretation of XML documents with embedded dynamic model," (in Russian), *Vestnik UGATU*, vol. 9, no. 2, pp. 88–97, 2007.]]

5. **Миронов В. В., Шакирова Г. Р.** Программно-инструментальное средство для создания и ведения динамических XML-документов // Вестник УГАТУ. 2007. Т. 9, № 5. С. 54–63. [[V. V. Mironov, G. R. Shakirova, "Programming instrumental tool for dynamic XML documents development and performing," (in Russian), *Vestnik UGATU*, vol. 8, no. 5, pp. 54–63, 2007.]]

6. **Миронов В. В., Юсупова Н. И., Шакирова Г. Р.** Ситуационно-ориентированные базы данных: концепция, архитектура, XML-реализация // Вестник УГАТУ. 2010. Т. 14, № 2 (37). С. 233–244. [[V. V. Mironov, N. I. Yusupova, and G. R. Shakirova, "Situation-oriented databases: concept, architecture, XML realization," (in Russian), *Vestnik UGATU*, vol. 14, no. 4 (39), pp. 200–209, 2010.]]

7. **Миронов В. В., Гусаренко А. С.** Ситуационно-ориентированные базы данных: концепция управления XML-данными на основе динамических DOM-объектов // Вестник УГАТУ. 2012. Т. 16, № 3 (48). С. 159–172. [[V. V. Mironov, A. S. Gusarenko, "Situation-oriented databases: the concept of managing XML-data based on dynamic DOM-objects" (in Russian), in *Vestnik UGATU*, vol. 16, no. 3 (48), pp. 159–172, 2012.]]

8. **Миронов В. В., Гусаренко А. С.** Динамические DOM-объекты в ситуационно-ориентированных базах данных: лингвистическое и алгоритмическое обеспечение источников данных // Вестник УГАТУ. 2012. Т. 16, № 6 (51). С. 167–176. [[V. V. Mironov, A. S. Gusarenko, "Dynamic DOM-objects

in situation-oriented databases: lingware and knoware of data sources" (in Russian), in *Vestnik UGATU*, vol. 16, no. 6 (51), pp. 167–176, 2012.]]

9. **Гусаренко А. С.**, Обработка XML-документов в ситуационно-ориентированных базах данных на основе динамических DOM-объектов: автореф. дис. ... канд. техн. наук. Уфа, 2013. 16 с. [[A. S. Gusarenko, "Handling XML-documents in situation-oriented databases based on dynamic DOM-objects" (in Russian), thesis abstract for the degree of candidate of technical sciences, UGATU, p. 16, 2013.]]

10. **Канашин В. В., Миронов В. В.** Иерархические виджеты: организация интерфейса пользователя в веб-приложениях на основе ситуационно-ориентированных баз данных // *Вестник УГАТУ*. 2013. Т. 17, № 2 (55). С. 138–149. [[Mironov V. V., Kanashin V. V. "Hierarchical widgets: user interface organization in web applications on the basis of situation-oriented databases" *Vestnik UGATU*. 2013; vol. 17, no. 2 (55), pp. 138–149.]]

11. **Канашин В. В., Миронов В. В.** Иерархические виджеты: ввод и контроль данных пользователя в веб-приложениях на основе ситуационно-ориентированных баз данных // *Вестник УГАТУ*. 2013. Т. 17, № 5 (58). С. 166–176. [[Mironov V. V., Kanashin V. V. "Hierarchical widgets: input and control of the user data in web applications on the basis of situation-oriented databases," *Vestnik UGATU*, 2013; vol. 17, no. 5 (58), pp. 166–176.]]

12. **Канашин В. В., Миронов В. В.** Иерархические виджеты: алгоритмы контроля данных пользователя в веб-приложениях на основе ситуационно-ориентированных баз данных // *Вестник УГАТУ*. 2014. Т. 18, № 1 (62). С. 204–213. [[Mironov V. V., Kanashin V. V. "Hierarchical widgets: user data control algorithms in web applications on the basis of situation-oriented data-bases," *Vestnik UGATU*, 2014; vol. 18, no. 1 (62), pp. 204–213.]]

13. **Канашин В. В., Миронов В. В.** Иерархические виджеты: опыт применения в веб-приложении на основе ситуационно-ориентированной базы данных // *Вестник УГАТУ*. 2014. Т. 18, № 2 (63). С. 185–196. [[V. V. Mironov, Kanashin V. V. "Hierarchical widgets: experience of use in the web application on the basis of situation-oriented database" *Vestnik*. 2014; vol. 18, no. 2 (63), pp. 185–196.]]

14. **Макарова Е. С., Миронов В. В.** Проектирование концептуальной модели данных для задач Web-OLAP на основе ситуационно-ориентированной базы данных // *Вестник УГАТУ*. 2012. Т. 16, № 6 (51). С. 177–188. [[E. S. Makarova and V. V. Mironov, "Web OLAP conceptual data model design on the basis of situation-oriented database," (in Russian), *Vestnik UGATU*, vol. 16, no. 6 (51), pp. 177–188, 2012.]]

15. **Макарова Е. С., Миронов В. В.** Функции аналитики в веб-приложениях на основе ситуационно-ориентированных баз данных // *Вестник УГАТУ*. 2013. Т. 17, № 5 (58). С. 150–165. [[E. S. Makarova and V. V. Mironov, "Analytical functions in web applications based on situation-oriented databases," (in Russian), *Vestnik UGATU*, vol. 17, no. 5 (58), pp. 150–165, 2013.]]

16. **Гусаренко А. С., Миронов В. В.** Smarty-объекты: вариант использования гетерогенных источников в ситуационно-ориентированных базах данных // *Вестник УГАТУ*. 2014. Т. 18, № 3 (63). С. 242–252. [[A. S. Gusarenko, V. V. Mironov, "Smarty-objects: use case of heterogeneous sources in situation-oriented databases," (in Russian), *Vestnik UGATU*, vol. 18, no. 3 (63), pp. 242–252, 2014.]]

17. **Гусаренко А. С., Миронов В. В.** Использование RESTful-сервисов в ситуационно-ориентированных базах данных // *Вестник УГАТУ*. 2015. Т. 19, № 1 (67). С. 204–211. [[A. S. Gusarenko, V. V. Mironov, "Using of RESTful-services in situation-oriented databases," (in Russian), *Vestnik UGATU*, vol. 19, no. 1 (67), pp. 204–211, 2015.]]

18. **Миронов В. В., Юсупова Н. И., Шакирова Г. Р.** Иерархические модели данных: концепции и реализация на основе XML / под ред. проф. Н. И. Юсуповой. М.: Машиностроение, 2011. 453 с. [[V. V. Mironov, N. I. Yusupova, and G. R. Shakirova, "Hierarchical data model: the concept and implementation of based on XML" (in Russian), 2011.]]

19. **Миронов В. В., Гусаренко А. С., Диметриев Р. Р., Сарваров М. Р.** Создание персонализированных документов на основе ситуационно-ориентированной базы данных // *Вестник УГАТУ*. 2014. Т. 18, № 4 (65). С. 191–197. [[V. V. Mironov, A. S. Gusarenko, R. R. Dimetrievev, M. R. Sarvarov, "The personalized documents generating using DOM-objects in situation-oriented databases," (in Russian), *Vestnik UGATU*, vol. 18, no. 4 (65), pp. 191–197, 2014.]]

20. **Миронов В. В., Юсупова Н. И., Гусаренко А. С.** Ситуационно-ориентированные базы данных: современное состояние и перспективы исследования // *Вестник УГАТУ*. 2015. Т. 19, № 2 (68). С. 188–199. [[V. V. Mironov, N. I. Yusupova, A. S. Gusarenko, "Situation-oriented databases: current status and prospects of research," (in Russian), *Vestnik UGATU*, vol. 19, no. 2 (68), pp. 188–199, 2015.]]

21. **Гусаренко А. С.** Модели создания документов в формате Office Open XML на основе ситуационно-ориентированной базы данных // *Прикладная информатика*. 2015. Т. 10, № 3. С. 62–75. [[A. S. Gusarenko, "Model for creating documents in Office Open XML format based on situation-oriented databases," (in Russian), *Applied Informatics*, vol. 10, no. 3, pp. 62–75, 2015.]]

22. **He W., Zhai J.** Application of the indent conversion based on XML and DOM // *Proc. 2013 Int. Conf. on Computational and Information Sciences (ICCIS' 2013)*, Vol. 1, P. 411–413, 2013.

23. **Benzaken V., et al.** Static and dynamic semantics of NoSQL languages. // *Proc. 40th Annual ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages (POPL' 2013)*, P. 101–113, Rome, Italy, 2013.

24. **Sladic G., Milosavljevic B., Konjovic Z., Vidakovic M.** Access control framework for xml document collection. // *Computer Science and Information Systems*, Vol. 8, No. 3, P. 591–609, 2011.

25. **Sudarsan R., Gray J.** Metamodel search: Using XPath to search domain-specific models. // *J. of Research and Practice in Information Technology*, Vol. 38, No. 4, P. 337–351, 2006.

26. **Tang N., Yu J. X., Wong K. F., Li J. X.** Fast XML structural join algorithms by partitioning. // *J. of Research and Practice in Information Technology*, Vol. 40, No. 1, P. 33–53, 2008.

27. **Dekeyser S., Hidders J., Paredaens J.** transaction model for XML databases. // *World Wide Web-Internet and Web Information Systems*, Vol. 7, No. 1, P. 29–57, 2004.

28. **Jea K. F., Chang T. P., Chen S. Y.** A semantic-based protocol for concurrency control in DOM database systems. // *J. of Information Science and Engineering*, Vol. 25, No. 5, P. 1617–1639, 2009.

29. **Kudrass T., Conrad M.** Management of XML documents in object-relational databases. // *XML-Based Data*

Management and Multimedia Engineering-Edbt 2002 Workshops, Vol. 2490, P. 210–227, 2002.

30. **Nassis V., Dillon T. S., Rajagopalapillai R., Rahayu W.** An XML document warehouse model. // Database Systems for Advanced Applications, Proceedings, Vol. 3882, P. 513–529, 2006.

31. **Batory D.** Multilevel models in model-driven engineering, product lines, and metaprogramming. // IBM Systems Journal, Vol. 45, No. 3, P. 527–539, 2006.

32. **Dejanovic I., Milosavljevic G., Perisic B., Tumbas M. A.** Domain-specific language for defining static structure of database applications. // Computer Science and Information Systems, Vol. 7, No. 3, P. 409–440, 2010.

33. **Su-Cheng H., Lee C. S.** Efficient preprocesses for fast storage and query retrieval in native XML database. // IETE Technical Review, Vol. 26, No. 1, P. 28–40, 2009.

ОБ АВТОРАХ

ГУСАРЕНКО Артем Сергеевич, асс. каф. автоматиз. сист. упр. Канд. техн. наук (УГАТУ, 2013). Дипл. информатик-экономист (УГАТУ, 2010). Иссл. в обл. иерархических моделей, ситуационно-ориентированных баз данных, ситуационного упр. и NoSQL.

МИРОНОВ Валерий Викторович, проф. каф. автоматиз. сист. упр. Дипл. радиопизик (Воронежск. гос. ун-т, 1975). Д-р техн. наук по упр. в техн. сист. (УГАТУ, 1995). Иссл. в обл. иерархических моделей и ситуационного упр.

METADATA

Title: Heterogeneous document sources in situationally-oriented databases

Authors: A. S. Gusarenko, V. V. Mironov

Affiliation:

Ufa State Aviation Technical University (UGATU), Russia.

Email: mironov@list.ru, artyomgusarenko@gmail.com

Language: Russian.

Source: Vestnik UGATU (scientific journal of Ufa State Aviation Technical University), vol. 19, no. 4 (70), pp. 124-131, 2015. ISSN 2225-2789 (Online), ISSN 1992-6502 (Print).

Abstract: Article uses XML-processing documents in situationally-oriented databases based on dynamic DOM-objects for the task personalized documents generating. The paper proposes a dynamic SODB model providing specifications and new elements to create a finished electronic documents based on the Office Open XML blanks prepared by means of word processors. In the states of the dynamic model blanks are generated using information from XML documents SODB server web application. The model in this paper allows the specification of routine operations to generate documents at a higher level of abstraction, using the syntax model SODB. The article shows the graphical equivalents, and text tools introduced in the dynamic model and interpreted using the interpreter embedded dynamic models for solving the problem of generation of documents. Interpreter embedded dynamic models works with documents Office Open XML, using the technology of dynamic DOM-objects, while the resulting documents have ways of representing and docx vdx.

Keywords: Web-application; dynamic model; NoSQL; RESTful-services; XML; JSON; DOM; Smarty; PHP.

About authors:

GUSARENKO, Artem Sergeevich, Cand. of Tech. Sci. (USATU, 2013), Dept. of Automated Systems., Grad. informatic-economist (USATU, 2010).

MIRONOV, Valeriy Viktorovich, Prof., Dept. of Automated Systems. Dipl. Radiophysicist (Voronezh State Univ., 1975). Cand. of Tech. Sci. (USATU, 1978), Dr. of Tech. Sci. (USATU, 1995).