

ИСПОЛЬЗОВАНИЕ RESTFUL-СЕРВИСОВ В СИТУАЦИОННО-ОРИЕНТИРОВАННЫХ БАЗАХ ДАННЫХ

А. С. Гусаренко¹, В. В. Миронов²

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

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

Поступила в редакцию 18 января 2015 г.

Аннотация. Рассматриваются RESTful-сервисы в качестве источников XML- и JSON-данных при обработке документов в ситуационно-ориентированных базах данных (СОБД). В динамической модели СОБД предусматривается возможность задания REST-источников данных, предполагающих запросы к внешним XML-сервисам. На основе интерпретации REST-источников в ходе интерпретации динамической модели формируются обращения к сервисам, результаты принимаются и загружаются в DOM-объекты. Интерпретатор динамической модели СОБД, таким образом, берет на себя рутинные операции по формированию и обработке запросов к внешним REST-сервисам. Рассматриваются графические, лингвистические средства для спецификации REST-источников в динамической модели. Обсуждается реализация модели на платформе PHP с использованием сервисов XML для Elibrary и JSON REST. Как пример использования подхода рассматривается задача генерирования на веб-сервере персонализированных документов.

Ключевые слова: веб-приложение; ситуационно-ориентированная база данных; NoSQL; RESTful; XML; JSON; DOM; Smarty; PHP.

ВВЕДЕНИЕ

Концепция обработки XML-документов в ситуационно-ориентированных базах данных (СОБД) [1] на основе динамических DOM- или Smarty-объектов, предложенная в работах [2–6], в частности, для решения задачи генерирования персонализированных документов в веб-приложениях, требует расширения в плане использования сторонних сервисов. Это означает, что контент DOM- или Smarty-объектов должен загружаться не только из XML- или JSON-документов, размещенных в хранилище документов СОБД, но и поставляться по запросу из сторонних приложений [7–10]. Внешние сервисы должны принимать запросы от СОБД-приложения и возвращают результат в виде XML- или JSON документов, которые загружаются в DOM- или Smarty-объекты, предусмотренные в динамической модели СОБД.

Такие сервисы в общем случае называются сервисами RESTful (Representational State Transfer – передача состояния) по названию запросов

передачи состояния REST. Для того чтобы приложение [11, 12] на базе СОБД [13, 14] запрашивало состояние стороннего приложения или сервиса [15–17], требуется выполнить запрос REST. Поэтому источники данных DOM- или Smarty-объектов требуется оснастить возможностью задания спецификаций, которые интерпретатор может использовать как отправную точку входа в сторонний сервис. Основу спецификации составляет URL-адрес запроса, который включает адреса сервиса и входные параметры запроса. Выбор инструментария обработки принимаемых данных зависит от типа возвращаемого сервисом результата. Результат сервисов может возвращаться в формате XML или JSON. В СОБД [18, 19] для обработки этих видов данных соответственно используется технология динамических DOM-объектов [20] и динамических Smarty-объектов.

Углубляясь в особенности технологий REST, следует отметить, что целесообразно использовать методы `file_get_contents` и возможности популярного расширения `cURL`. Первый из методов использует HTTP-запросы GET и POST, с помощью которых передаются параметры, регламентирующие действия сервера.

Для обработки возникающих ошибок также требуется соответствующее обеспечение динамической модели и интерпретатора.

Отметим, что это не единственный метод выполнения запросов через HTTP, известны также расширение cURL и пакеты PEAR. Поэтому, если интерпретатор встречает в источниках динамических DOM-объектов [21] ссылки на сторонние RESTful-сервисы, он должен проверить, подключены ли соответствующие расширения, и если нет, подключить их самостоятельно. Необходимо запрограммировать эту функциональность, а модель оснастить соответствующим лингвистическим обеспечением, предоставляющим разработчику возможность работы с REST.

В статье развивается концепция СОБД, предложенная в работах [2–6]. Ранее в работах рассматривались аспекты обработки документов из хранилища на сервере, в этой статье рассматривается использование документов, генерируемых внешними сервисами.

Статья опирается на предшествующие работы авторов [7–18], где были исследованы различные аспекты обработки XML-документов [22] из хранилища документов на основе технологии динамических DOM/Smarty-объектов, задаваемых средствами динамической модели СОБД.

1. ВНЕШНИЕ RESTful-СЕРВИСЫ

В качестве примеров RESTful-сервисов используются специализированные шлюзы, кото-

рые реализуют XML- и JSON сервисы. Эти сервисы разработаны авторами в рамках Системы открытых журналов OJS (Open Journal Systems – программное обеспечение, разработанное в рамках Public Knowledge Project) с использованием API этой системы. Первый сервис реализован в виде XML-шлюза "OJSXMLJournalReport", второй – в виде JSON-шлюза "OJSJSONJournalReport". Оба сервиса носят прикладной характер и генерируют отчеты по журналам системы OJS. Отчеты сервиса "OJSXMLJournalReport" генерируются в формате XML, а сервиса "OJSJSONJournalReport" – в формате JSON. Для дальнейшей обработки данных, содержащихся в отчетах, они загружаются в DOM- или Smarty- объекты. Общая картина использования сервисов представлена на рис. 1.

Здесь параметры запроса [23] задаются в модели СОБД, а затем в процессе интерпретации модели собирается запрос, направляемый внешнему сервису. Обработка запроса [24] осуществляется на стороне сервиса и заключается в создании объектов доступа к данным, через которые происходит обращение к базе данных сервиса и получаются порции персонализированных данных, объединяемых в результирующий отчет, возвращаемый в приложение [25]. В приложении полученные данные загружаются в динамические DOM- или Smarty-объекты [26] для последующей обработки.

Параметры запросов REST API. При обращении к сервисам используются HTTP запросы GET, POST, PUT, DELETE, MOVE, COPY.

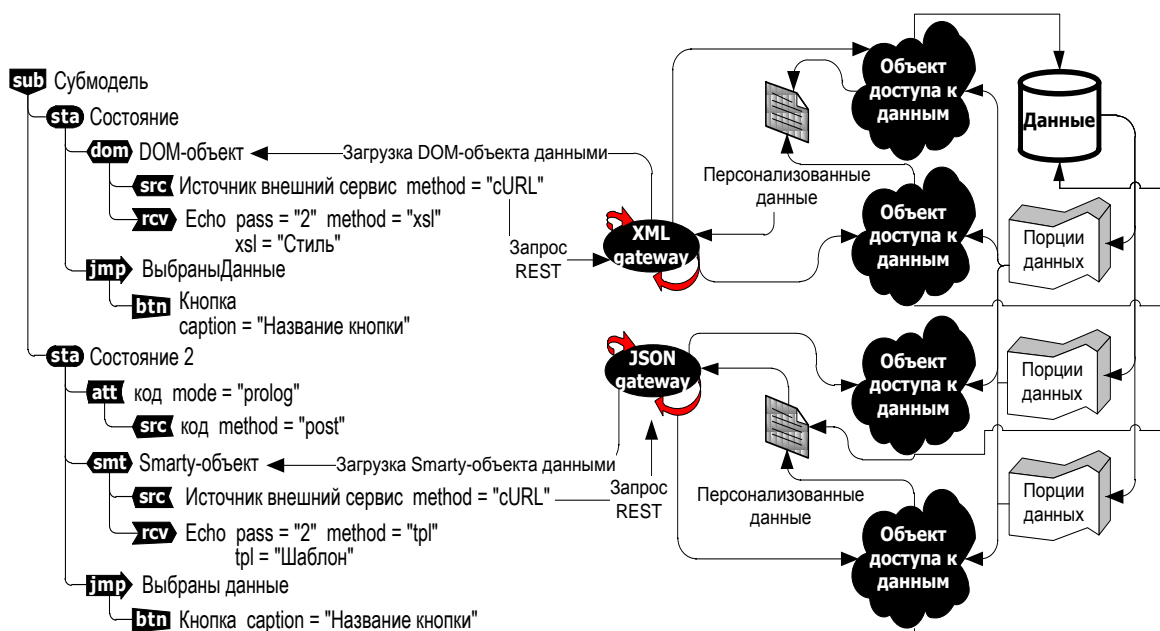


Рис. 1. Задействование внешних RESTful-сервисов из приложения СОБД

В URL задаются параметры аутентификации, если она требуется, токен доступа, локализация для указания языка, в котором возвращаются ответы сервиса. С помощью REST API возможно не только получение файлов и их загрузка, но и аутентификация пользователя в сервисе, получение персональных данных пользователя, перемещение, удаление и создание файлов. Параметры также устанавливают требуемые свойства для файлов и папок. В примере на рис. 1 приведен сервис, не использующий аутентификацию (он генерирует отчет с использованием данных [27], находящихся в открытом доступе).

В сервисах от известных поставщиков (вендоров) часто требуется регистрировать приложение. Например, для того чтобы наладить работу с сервисами Microsoft, требуется создать учетную запись и зарегистрировать приложение, получив секретные идентификационные сведения для доступа к сервису. В рассматриваемом примере требуется лишь подключить оба сервиса в настройках OJS, после этого открытые данные станут доступными для приложения СОБД по URL сервисов.

2. RESTful-ИСТОЧНИКИ ДАННЫХ В МОДЕЛИ СОБД

В плане лингвистического обеспечения HSM – динамической модели [28–30] СОБД – требуется расширить описание источников дан-

ных, добавив возможность задания внешних сервисов.

В динамической модели СОБД конкретные документы, загружаемые в DOM- и Smarty-объекты, специфицируются с помощью элементов **doc** [2–18], которые задают отображение источников данных на хранилище документов СОБД. Аналогичные же doc-элементы используются для отображения источников данных на RESTful-сервисы. Документы, получаемые от сервисов, задаются в атрибутах элемента **doc** (рис. 2). При этом тип документа XML и JSON задается атрибутом типа сервиса: *type = "XMLService"* или *type = "JSONService"*, а поставщик сервиса – атрибутом *vendor*. Адрес доступа к сервису задается в атрибуте *baseurl*. Для каждого сервиса в зависимости от вендора требуется своя функциональность подключения и задействования, поэтому в doc-элементе предусмотрен атрибут *prmt*, задающий список параметров для обращения к сервису: *prmt = "параметр1 параметр2 ..."*.

В процессе обработки doc-элемента интерпретатор динамической модели подготавливает запрос к сервису. При обработке src-элемента (источника данных), ссылающегося на doc-элемент, интерпретатор исполняет подготовленный ранее запрос к сервису, получает от сервиса результат-документ и загружает его в соответствующий DOM- или Smarty-объект.

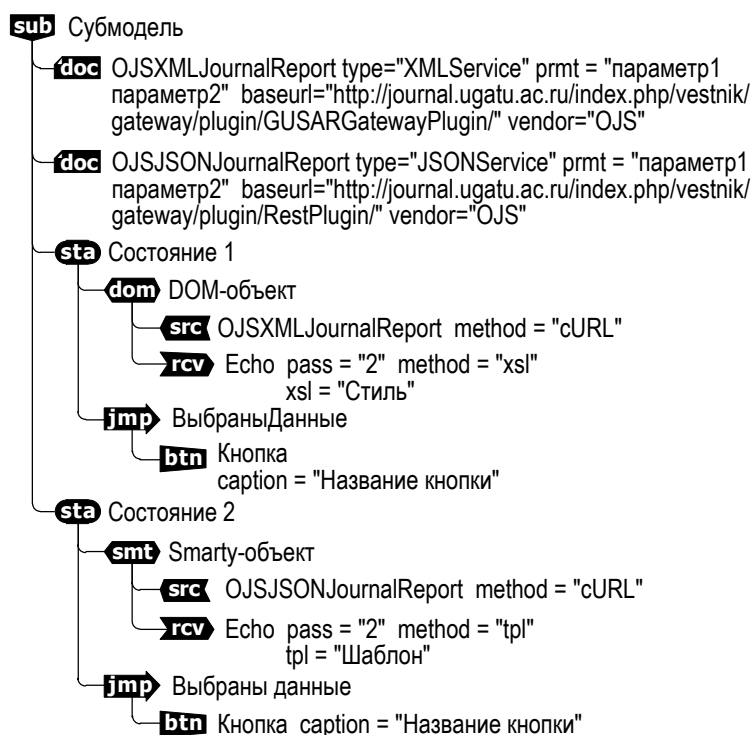


Рис. 2. Диаграмма динамической модели HSM, с состояниями которой ассоциированы DOM- и Smarty-объекты, загружаемые документами из внешних RESTful-сервисов

Например, на рис. 2 элемент *doc:OJSXML-JournalReport type="XMLService"* подготавливает запрос к XML-сервису, находящемуся по адресу *"http://journal.ugatu.ac.ru/index.php/vestnik/gateway/plugin/GUSARGatewayPlugin/"*. Этот URL дополняется параметрами *"параметр1 параметр2"* со спецификацией REST API от вендора сервиса OJS. Аналогичным образом производится 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 задаются источники от *XMLService*.

Данная функциональность соответствует декларативному модельно-ориентированному подходу работы с внешними RESTful-сервисами [24–30], обеспечивающему снижение объема программирования и расширение ассортимента источников данных СОБД.

Запросы HTTP в модели. Таким образом, предлагаемый подход позволяет обрабатывать документы, не только размещенные в локальном хранилище СОБД [31], но генерируемые внешними RESTful-сервисами. Использование REST-запросов имеет следующие особенности:

- в DOM- или Smarty-элементах, заданных в состояниях динамической модели, указываются параметры для компилирования запроса GET или POST. Эти запросы часто используются в веб-приложениях; так запрос POST отправляет данные по заданному 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 отдельной сущности объекта;

- для DOM- 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 уведомляет пользователя о временном перенаправлении запроса; при этом пользователь имеет право самостоятельно принять решение о перенаправлении.

REST-метод cURL. Для организации запросов к сторонним сервисам на платформе PHP может использоваться специальное расширение cURL. Расширение позволяет взаимодействовать со многими внешними сервисами по различным протоколам, в том числе HTTP. Чтобы создать запрос к сервису, требуется запрограммировать на языке PHP запрос с указанием URL и других пользовательских опций спецификации 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 методами являются *XMLReader* или *DOM* [19–21], для JSON-ответов – это динамические Smarty-объекты.

REST-метод file_get_contents. Другой альтернативный встроенный метод – *file_get_contents* – не требует настройки специального расширения, но обладает более скромными возможностями получения ответов от RESTful-сервиса. Это функция PHP, которая получает данные из файла в виде строки. Дальнейшая обработка ответа сервиса заключается в обработке полученной строки, которая может быть загружена в DOM-объект. В случае если в модели задано использование расширения cURL, но оно не включено или его невозможно задействовать, интерпретатор автоматически использует инструментарий создания запросов с помощью *file_get_contents*. Параметры в запрос передаются из модели тем же образом, что и для запросов cURL, и типы REST-запросов поддерживаются аналогично.

3. ПРИМЕР СТОРОННИХ СЕРВИСОВ В СИСТЕМЕ OJS

Для того чтобы предметно проиллюстрировать предлагаемый подход, рассмотрим практический пример СОБД-приложения с использованием двух сервисов, разработанных в рамках системы OJS, используемой в качестве платформы для журнала «Вестник УГАТУ». Система OJS позволяет разработчикам создавать плагины, обслуживающие специфические для данного журнала функции. В частности, она позволяет создавать XML- и JSON-сервисы. Используя эти возможности, авторами были разработаны и установлены в системе два сервиса, формирующих отчеты о содержании указанного выпуска журнала. Оба сервиса предоставляют одну и ту же информацию в различных форматах:

- **GUSARGatewayPlugin** – отчет о содержании выпуска журнала в формате XML, URL: <http://journal.ugatu.ac.ru/index.php/vestnik/gateway/plugin/GUSARGatewayPlugin/>;
- **RestPlugin** – отчет о содержании выпуска журнала в формате JSON, URL: <http://journal.ugatu.ac.ru/index.php/vestnik/gateway/>.

Иерархическая модель данных, содержащихся в отчетах, приведена на рис. 3 (использована графическая нотация представления иерархической модели, предложенная в [18]; для наглядности имена элементов модели записаны на русском языке).

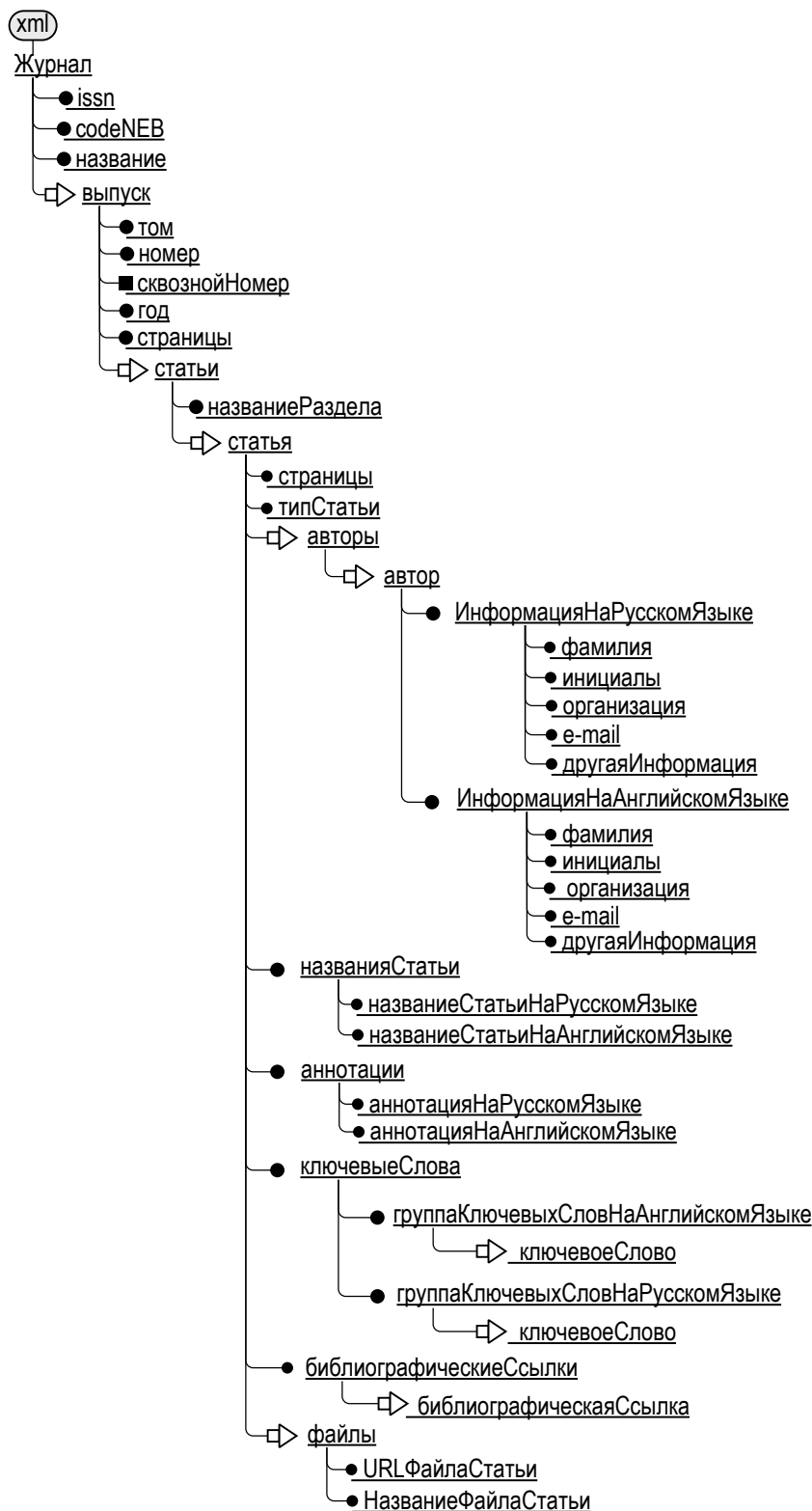


Рис. 3. Диаграмма RESTful-источника данных XML-сервиса OJS в состоянии "Отчеты журнала"

Отчет в формате XML, генерируемый сервисом GUSARGatewayPlugin, используется в СОБД-приложении, формирующем страницы содержания для бумажной версии выпусков журнала в виде документа Word. Прежде страницы содержания приходилось формировать вручную, заноса в шаблон-заготовку Word сведения о статьях выпуска: название, список авторов, номера страниц. Теперь специальное СОБД-приложение запрашивает у данного сервиса XML-отчет о составе нужного выпуска журнала и загружает его в DOM-объект. В другой DOM-объект загружается шаблон-заготовка Word. Далее сведения из отчета заносятся в шаблон-заготовку, формируя результирующий Word-документ. Детали данной технологии генерации документов представлены в работе [19]. Автоматическая генерация документа Word на основе данных, имеющихся в базе системы OJS, заметно снижает трудоемкость подготовки страниц содержания.

Отчет в формате JSON, генерируемый сервисом RestPlugin, в настоящее время лишь демонстрирует возможности представления иерархически организованных данных, альтернативные XML. В дальнейшем планируется использование его для автоматизации составления других видов отчетов в системе OJS.

ЗАКЛЮЧЕНИЕ

В статье рассмотрено использование сторонних RESTful-сервисов в качестве внешних источников данных в приложениях на основе СОБД. На базе технологии динамических DOM/Smarty-объектов возникает возможность загружать для обработки XML/JSON-документы, не только извлекая их из хранилища СОБД, но и получать их из сервисов сети Интернет.

Для этого были расширены функциональные возможности динамической модели СОБД:

- элементы doc, отображающие источники данных на хранилище документов, расширены возможностью спецификации внешних RESTful-сервисов (для источника DOM-объекта появилась возможность задать внешний XML-сервис, а для Smarty-объекта – внешний JSON-сервис;

- интерпретатор динамической модели дополнен функциями формирования запроса к сторонним сервисам при обработке doc-элементов, отправки запроса, получения от сервиса ответного документа и загрузки его в соответствующий DOM/Smarty-объект.

Реализуются два метода выполнения REST-запросов к внешним сервисам: *cURL* и *file_get_contents*.

Разработанная технология проиллюстрирована на приме двух сервисов XML и JSON, созданных в системе OJS для генерации сведений по выпускам журнала открытого доступа. Использование сервисов позволяет СОБД-приложению формировать сложно оформленные отчеты, в частности, автоматически генерировать Word-документы содержания выпусков журнала.

СПИСОК ЛИТЕРАТУРЫ

1. **Strauch C., Kriha W.** (2012, Nov. 11). *NoSQL databases* [Online]. Available: <http://www.christof-strauch.de/nosql dbs.pdf>
2. **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.]]
3. **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.]]
4. **Гусаренко А. С.** Обработка XML-документов в ситуационно-ориентированных базах данных на основе динамических DOM-объектов: автореф. дис. ... канд. техн. наук / Уфимск. гос. авиац. техн. ун-т. Уфа, 2013. [[A. S. Gusarenko, *Handling XML-documents in situationally-oriented databases based on dynamic DOM-objects*, (in Russian), Thesis abstract for the degree of candidate of technical sciences, UGATU, 2013.]]
5. **Гусаренко А. С., Миронов В. В.** Алгоритмическое и лингвистическое обеспечение ситуационно-ориентированных баз данных на основе динамических DOM-объектов // Информационные и инфокоммуникационные технологии: сб. науч. тр. 8-й всерос. зимн. шк.-сем. аспирантов и молодых ученых. Уфа: УГАТУ, 2013. Т. 1. С. 119–122. [[V. V. Mironov and A. S. Gusarenko, "Algorithmic and and linguistic support of situation-oriented databases based on dynamic DOM-objects," (in Russian), in *Proc. 8th Workshop of winter school postgraduates and young researchers*, Ufa: UGATU, 2013, vol. 1, pp. 119-122.]]
6. **Гусаренко А. С., Миронов В. В.** Профилирование алгоритма ситуационно-ориентированной базы данных на основе динамических DOM-объектов // Информационные и инфокоммуникационные технологии: сб. науч. тр. 8-й всерос. зимн. шк.-сем. аспирантов и молодых ученых. Уфа: УГАТУ, 2013. Т. 1. С. 115–118. [[V. V. Mironov and A. S. Gusarenko, "Profiling algorithm of situation-oriented database based on dynamic DOM-objects," (in Russian), in *Proc. 8th Workshop of winter school postgraduates and young researchers*, vol. 1, pp. 115-118, Ufa: UGATU, 2013.]]

7. **Канашин В. В., Миронов В. В.** Иерархические виджеты: опыт применения в веб-приложении на основе ситуационно-ориентированной базы данных // Вестник УГАТУ. 2013. Т. 18, № 2 (63). С. 185–196. [[V. V. Mironov, V. V. Kanashin, "Hierarchical widgets: experience of use in the web application on the basis of situation-oriented database," *Vestnik UGATU*, vol. 18, no. 2 (63), pp. 185-196, 2014.]]
8. **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. [[V. V. Mironov, V. V. Kanashin, "Hierarchical widgets: user interface organization in web applications on the basis of situation-oriented databases," *Vestnik UGATU*, vol. 17, no. 2 (55), pp. 138-149, 2013.]]
9. **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. [[V. V. Mironov, V. V. Kanashin, "Hierarchical widgets: input and control of the user data in web applications on the basis of situation-oriented databases," *Vestnik UGATU*, vol. 17, no. 5 (58), pp. 166-176, 2013.]]
10. **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. [[V. V. Mironov, V. V. Kanashin, "Hierarchical widgets: user data control algorithms in web applications on the basis of situation-oriented databases," *Vestnik UGATU*, vol. 18, no. 1 (62), pp. 204-213, 2014.]]
11. **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.]]
12. **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.]]
13. **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.]]
14. **Миронов В. В., Юсупова Н. И., Шакирова Г. Р.** Ситуационно-ориентированные базы данных: внешние представления на основе XSL // Вестник УГАТУ. 2010. Т. 14, № 4 (39). С. 200–209. [[V. V. Mironov, N. I. Yusupova, and G. R. Shakirova, "Situation-oriented databases: external view in the basis of XSL," (in Russian), *Vestnik UGATU*, vol. 14, no. 2 (37), pp. 233-244, 2010.]]
15. **Миронов В. В., Маликова К. Э.** Интернет-приложения на основе встроенных динамических моделей: идея, концепция, безопасность // Вестник УГАТУ. 2009. Т. 13, № 2 (35). С. 167–179. [[V. V. Mironov and K. E. Malikova, "Internet applications based on embedded dynamic models: idea, concept," (in Russian), *Vestnik UGATU*, vol. 13, no. 2 (35), pp. 167-179, 2009.]]
16. **Миронов В. В., Маликова К. Э.** Интернет-приложения на основе встроенных динамических моделей: архитектура, структура данных, интерпретация // Вестник УГАТУ. 2010. Т. 14, № 1 (36). С. 154–163. [[V. V. Mironov and K. E. Malikova, "Internet applications based on embedded dynamic models: architecture, data structure, interpretation," (in Russian), *Vestnik UGATU*, vol. 14, no. 1 (36), pp. 154-163, 2010.]]
17. **Миронов В. В., Маликова К. Э.** Интернет-приложения на основе встроенных динамических моделей: элементы управления пользовательского интерфейса // Вестник УГАТУ. 2010. Т. 14, № 5 (40). С. 170–175. [[V. V. Mironov and K. E. Malikova, "Internet applications based on embedded dynamic models: user interface controls," (in Russian), *Vestnik UGATU*, vol. 14, no. 5 (40), pp. 170-175, 2010.]]
18. **Миронов В. В., Юсупова Н. И., Шакирова Г. Р.** Иерархические модели данных: концепция 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). Moscow: Masinostroenie, 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, and 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. **He W., Zhai J.** "Application of the indent conversion based on XML and DOM," in *Proc. 2013 Int. Conf. on Computational and Information Sciences (ICCIS' 2013)*, vol. 1, pp. 411–413, 2013.
21. **Benzaken V., et al.** "Static and dynamic semantics of NoSQL languages," in *Proc. 40th Annual ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages (POPL' 2013)*, pp. 101-113, Rome, Italy, 2013.
22. **Sladic G., Milosavljevic B., Konjovic Z., Vidakovic M.** "Access control framework for xml document collections," in *Computer Science and Information Systems*, vol. 8, no. 3, pp. 591-609, 2011.
23. **Sudarsan R., Gray J.** "Metamodel search: Using XPath to search domain-specific models," in *J. of Research and Practice in Information Technology*, vol. 38, no. 4, pp. 337-351, 2006.
24. **Tang N., Yu J. X., Wong K. F., Li J. X.** "Fast XML structural join algorithms by partitioning," in *J. of Research and Practice in Information Technology*, vol. 40, no. 1, pp. 33-53, 2008.
25. **Dekeyser S., Hidders J., Paredaens J.** "A transaction model for XML databases," in *World Wide Web-Internet and Web Information Systems*, vol. 7, no. 1, pp. 29-57, 2004.
26. **Jea K. F., Chang T. P., Chen S. Y.** "A semantic-based protocol for concurrency control in DOM database systems," in *J. of Information Science and Engineering*, vol. 25, no. 5, pp. 1617-1639, 2009.
27. **Kudrass T., Conrad M.** "Management of XML documents in object-relational databases," in *XML-Based Data Management and Multimedia Engineering-Edbt 2002 Workshops*, vol. 2490, pp. 210-227, 2002.
28. **Nassis V., Dillon T. S., Rajagopalapillai R., Rahayu W.** "An XML document warehouse model," in *Database Systems*

for *Advanced Applications, Proceedings*, vol. 3882, pp. 513-529, 2006.

29. **Batory D.** "Multilevel models in model-driven engineering, product lines, and metaprogramming," in *IBM Systems Journal*, vol. 45, no. 3, pp. 527-539, 2006.

30. **Dejanovic I., Milosavljevic G., Perisic B., Tumbas M. A.** "Domain-specific language for defining static structure of database applications," in *Computer Science and Information Systems*, vol. 7, no. 3, pp. 409-440, 2010.

31. **Su-Cheng H., Lee C. S.** "Efficient preprocesses for fast storage and query retrieval in native XML database," in *IETE Technical Review*, vol. 26, no. 1, pp. 28-40, 2009.

ОБ АВТОРАХ

ГУСАРЕНКО Артем Сергеевич, асс. каф. автоматизированных систем управления. Канд. техн. наук (УГАТУ, 2013). Дипл. инф.-экон. (УГАТУ, 2010). Иссл. в обл. иерарх. моделей, сит.-ориент. баз данных, NoSQL.

МИРОНОВ Валерий Викторович, проф. каф. автоматизированных систем управления. Дипл. р/физик (Воронежск. гос. ун-т, 1975). Д-р техн. наук по упр. в техн. системах (УГАТУ, 1995). Иссл. в обл. иерарх. моделей и сит. упр-я.

METADATA

Title: Using of RESTful-services in situationally-oriented databases

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

Affiliation:

Ufa State Aviation Technical University (UGATU), Russia.

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

Language: Russian.

Source: Vestnik UGATU (scientific journal of Ufa State Aviation Technical University), vol. 19, no. 1 (67), pp. 204-211, 2015. ISSN 2225-2789 (Online), ISSN 1992-6502 (Print).

Abstract: Article uses XML-processing documents in situationally-oriented databases (SODB) based on dynamic DOM-objects for the task of generating personalized documents. The paper proposes a dynamic model SODB 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. The work looks at ways to specification of sources and processing XML-based data dynamic DOM-objects bound to the states of the dynamic model. Proposed to implement the model on the platform PHP using blanks docx and 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).