

SMARTY-ОБЪЕКТЫ: ВАРИАНТ ИСПОЛЬЗОВАНИЯ ГЕТЕРОГЕННЫХ ИСТОЧНИКОВ 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 16 апреля 2014 г.

Аннотация. В рамках ранее разработанной концепции модели DOM-элементов в составе динамической модели ситуационно-ориентированной базы данных предлагается динамическая модель более высокого уровня абстракции, предусматривающая графические и текстовые средства спецификации источников данных на основе JSON. Описывается способ спецификации JSON источников с помощью JSON-элементов, ассоциированных с состояниями динамической модели, которые регламентируют заполнение Smarty-объекта и/или Smarty-массива JSON-контентом, когда родительское состояние становится текущим. Предлагается реализация модели на платформе PHP с использованием tpl-шаблонов Smarty.

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

Развивая концепцию ситуационно-ориентированных баз данных (СОБД), предложенную в работах [1–6], необходимо отметить, что в основу NoSQL баз данных [7–14] входят различные виды источников [2–6]. Текущее положение дел таково, что базы данных имеют в своей структуре источники JSON-типа, представляющие собой массив или объект данных и не только используемые для информационного обмена между сервером и клиентом веб-приложения, но и служащие основой баз данных. Такой формат является дополнительным способом хранения данных на сервере, с его помощью можно описывать объекты и сложные иерархические структуры данных, но СОБД в своем наборе инструментов пока не имеет осознанного подхода к подключению и обработке источников данного типа, а также алгоритмов, лингвистического обеспечения и встроенных средств вывода контента пользователю [3].

В работах [3–5] рассматривались различные аспекты использования источников XML и отображения динамического контента с помощью XSLT, но не рассматривались вопросы, связанные с JSON и Smarty как платформы СОБД. Это привело к тому, что сейчас нет специальных

видов обеспечения для СОБД по работе с указанной функциональностью [3].

Если в базах данных XML для отображения контента используется технология XSLT-шаблонизации, то для JSON-источников эту роль может взять на себя набравший популярность шаблонизатор Smarty. Компилирующий обработчик шаблонов Smarty имеет базовые и расширенные возможности по отображению таких структур данных как массивы и объекты, создание шаблонов имеет структуру, подобную PHP-программе, также как в JSON описание данных похоже на описание на JavaScript-программы.

В данной статье рассматриваются вопросы автоматического формирования контента из ассоциированных JSON-объектов и JSON-массивов в состояниях динамической модели СОБД, а также спецификации отображения с помощью Smarty в модели веб-приложения [7–10].

1. ВВОДНЫЕ ЗАМЕЧАНИЯ

Данная статья является развитием работ по созданию инструментальных, лингвистических средств СОБД [3], в которых были разработаны средства спецификации динамических DOM-объектов. В данном случае возникает задача автоматического создания Smarty-объектов и заполнения их контентом из JSON-источников.

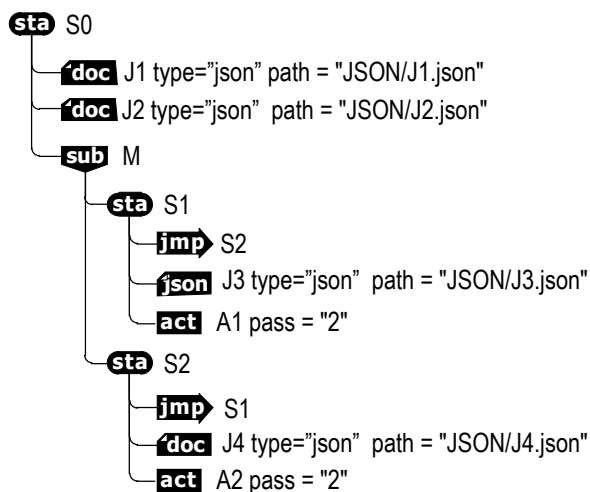


Рис. 1. Диаграмма динамической модели со спецификацией JSON-документов

Для создания и обработки новых источников требуется прилагать значительные усилия, особенно в условиях, когда JSON-источник имеет сложную иерархическую структуру, а СОБД не имеет реализованных методов и подходов к их обработке. Требуется предусмотреть в модели специальные элементы, специфицирующие новый тип источника – документы JSON (smt, рис. 1), в процессе интерпретации декодируемые и загружаемые в объекты / массивы Smarty. Для использования результирующего объекта / массива в состояниях модели также требуется предусмотреть элементы шаблонов tpl. Для перехвата результата, вывода компиляции шаблона также требуются атрибуты, указывающие тип ресурса, шаблона вместе с путем и названием шаблона. В соответствии с предлагаемым подходом в модели к состояниям привязываются Smarty-элементы. Элементы Smarty-объектов создаются, когда состояние, к которому они привязаны, становится текущим. Известный подход, в сравнении с предлагаемым, основан на спецификации шаблона и программировании функций по обработке JSON-источников в программе или подпрограмме. Ручной метод не позволяет использовать модель для повышения уровня абстракции при использовании типовых операций соединения, извлечения и фильтрации данных при обработке JSON-источников и вывода результата с помощью шаблонов tpl. В этих условиях логично оснастить этими функциями модель СОБД.

В предлагаемом подходе:

- у элементов-состояний динамической модели предусмотрены дочерние Smarty-элементы, дочерние элементы-источники которых указывают на загружаемые JSON-данные,

а дочерние элементы-приемники – на сохраняемый JSON-результат;

- в ходе интерпретации динамической модели интерпретатором происходит автоматическое порождение Smarty-объектов для текущих состояний модели и загрузка JSON-данных с указанным преобразованием.

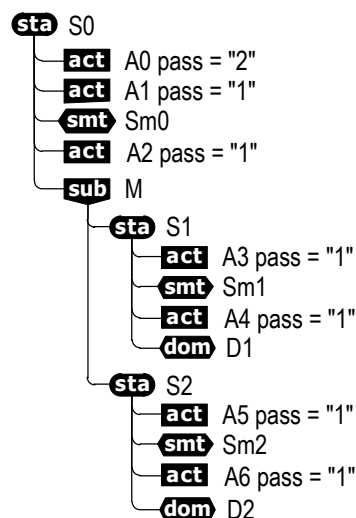


Рис. 2. Диаграмма динамической модели, с состояниями которой ассоциированы Smarty-объекты

Smarty-объекты. Объекты Smarty (рис. 2) порождаются в ходе интерпретации модели и имеют описание на языке программирования, заполняются JSON-данными, а затем используются в шаблонах tpl. Объект Smarty является основой для формирования результирующего объекта на основе информации из JSON-источника. Результирующий документ рис. 3 формируется на основе извлечения и фильтрации документа источника, по отношению к результирующему источнику документ источника является дочерним.

Smarty-массивы. Вторая особенность, которой обладает Smarty, – возможность спецификации ассоциативных массивов. И поэтому результирующий документ представляет собой содержимое слияния двух и более ассоциативных массивов JSON-данных. При этом определенные данные извлекаются, фильтруются в соответствии с условиями, заданными в модели.

Использование JSON-массивов и объектов обеспечивается языковыми средствами [3], и вполне оправданы, так как имеются реализованные средства кодирования/декодирования JSON в объекты и массивы.

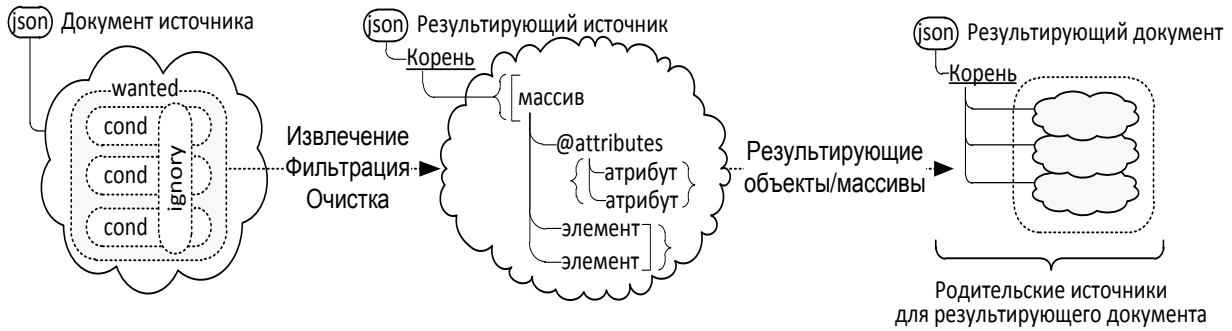


Рис. 3. Формирование Smarty-объекта на основе информации из JSON-источников данных

Диаграмма модели на рис. 2 предусматривает использование в состояниях динамические DOM-объекты [7–10] и Smarty-объекты. Динамические DOM-объекты предусматривают автоматическую загрузку данных XML-происхождения и облегчают задание спецификаций по обработке таких данных, но не решают задачу использования JSON-содержимого источников в ситуациях приложения. Таким образом, необходим инструментарий, встроенный в интерпретатор динамической модели [2], реализующий функциональность автоматической загрузки и обработки в состояниях динамической модели JSON-контента наряду с реализованным подходом в работах [3, 4] по обработке XML-источников данных.

2. КОНЦЕПЦИЯ SMARTY-ОБЪЕКТОВ

Smarty-элементы. Для того чтобы в ходе интерпретации автоматически создавались Smarty-объекты, в которые загружаются JSON-данные, ассоциированные с состояниями модели, необходимо предусмотреть в модели Smarty-и JSON-элементы. Элементы JSON [13] описывают источник данных, принимаемый к обработке. Для обработки заданных в состояниях данных предусмотрим в модели Smarty-элементы и вложенные в них источники данных. В соответствии с этими спецификациями добавим в интерпретатор функциональность по созданию Smarty-объектов, что позволит создавать эти объекты, когда родительские состояния становятся текущими, и удалять их, когда состояние перестает быть текущим. Это позволит снизить трудоемкость подготовки Smarty-объектов и массивов при обработке JSON, а также задавать типовые операции преобразования JSON-данных. Пример спецификации приведен на рис. 4.

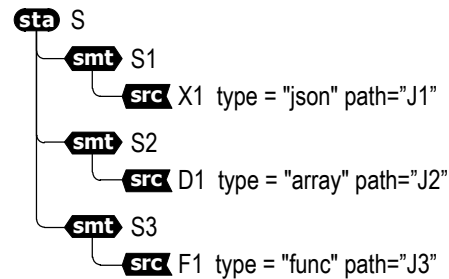


Рис. 4. Диаграмма динамической модели, в которой заданы Smarty-объекты с помощью Smarty-элементов

Для спецификаций Smarty-объектов предусмотрен специальный символ **smt**, обозначающий в графической нотации модели описание объекта; в нем задается имя объекта и требующийся шаблон для отображения результата.

В описание источника **src**, ранее введенное для источников динамических DOM-объектов [4–6], добавим конструкции задания типа источника `type="json"` и `type="array"`, а также ссылки `path`, по которому интерпретатор извлекает требующийся контент.

Дальнейшая реализация идеи по сокращению трудоемкости требует последующего исследования вопросов по обеспечению источников данных.

JSON-контент в Smarty-объектах. Подключение JSON-источников наряду с XML-контентом актуально, если рассматривать эту задачу исходя из того, что в сравнении с XML JSON имеет более компактную форму записи при обмене результатами выполнения HTTP-запроса с использованием объекта `XmlHttpRequest`, используемого для реализации технологии AJAX. Это может быть массив или объект в памяти, который загружается через объект `XmlHttpRequest`. В случае если мы используем JSON-контент с сервера, требуется спецификация пути, по которому находится файл. В таких

условиях СОБД требуется инструментарий, позволяющий обрабатывать и отображать JSON-документы, находящиеся как в оперативной, так и во внешней памяти.

3. ПРИМЕР ИСПОЛЬЗОВАНИЯ SMARTY-ОБЪЕКТОВ

Чтобы проиллюстрировать предлагаемую концепцию в качестве практического примера использования Smarty-объектов, рассмотрим пример решения задачи с использованием JSON, аналогичный тому, что приводился в работе [2] для задачи отображения сведений о студентах, предметах и сдачах с помощью динамических DOM-объектов [15–25].

Исходные данные. Пусть данные о студентах, предметах и сдачах хранятся в файлах JSON: stud.json, predm.json, sdacha.json, которые расположены в папке JSON сервера СОБД. На рис. 5 представлены концептуальные модели этих документов и тестовые экземпляры их содержимого. Документ stud.json (рис. 5, а) в корневом объекте «Студенты» может содержать несколько вложенных экземпляров объекта «студент» с атрибутами «код» (код студента)

и «фио» (фамилия, имя, отчество студента), причем атрибут «код» является идентификатором. Экземпляр этого документа (рис. 5, а) содержит сведения о трех студентах: Иванове, Петрове и Сидорове. Аналогичным образом JSON-документ predm.json (рис. 5, б) в корневом объекте «Предметы» может содержать несколько вложенных экземпляров объекта «предмет» с атрибутами «код» (код предмета) и «назв» (название предмета), причем атрибут «код» является идентификатором. Модель экземпляра этого документа [26–28] (рис. 5, б) содержит сведения о предметах «теория систем», «матлогика» и «криптография». Документ sdacha.json (рис. 5, в) в корневом объекте «Сдачи» может содержать несколько вложенных экземпляров объекта «сдача», соответствующих сдачам определенным студентом определенного предмета, с атрибутами «кодСт» (код студента), «кодПр» (код предмета) и «оценка» (оценка студента по предмету), причем пара атрибутов «кодСт, кодПр» является составным идентификатором. Модель [29–39] экземпляра этого документа (рис. 5, в) содержит сведения о пяти сдачах: трех сдачах Иванова и двух Петрова (Сидоров не сдал ни одного предмета).

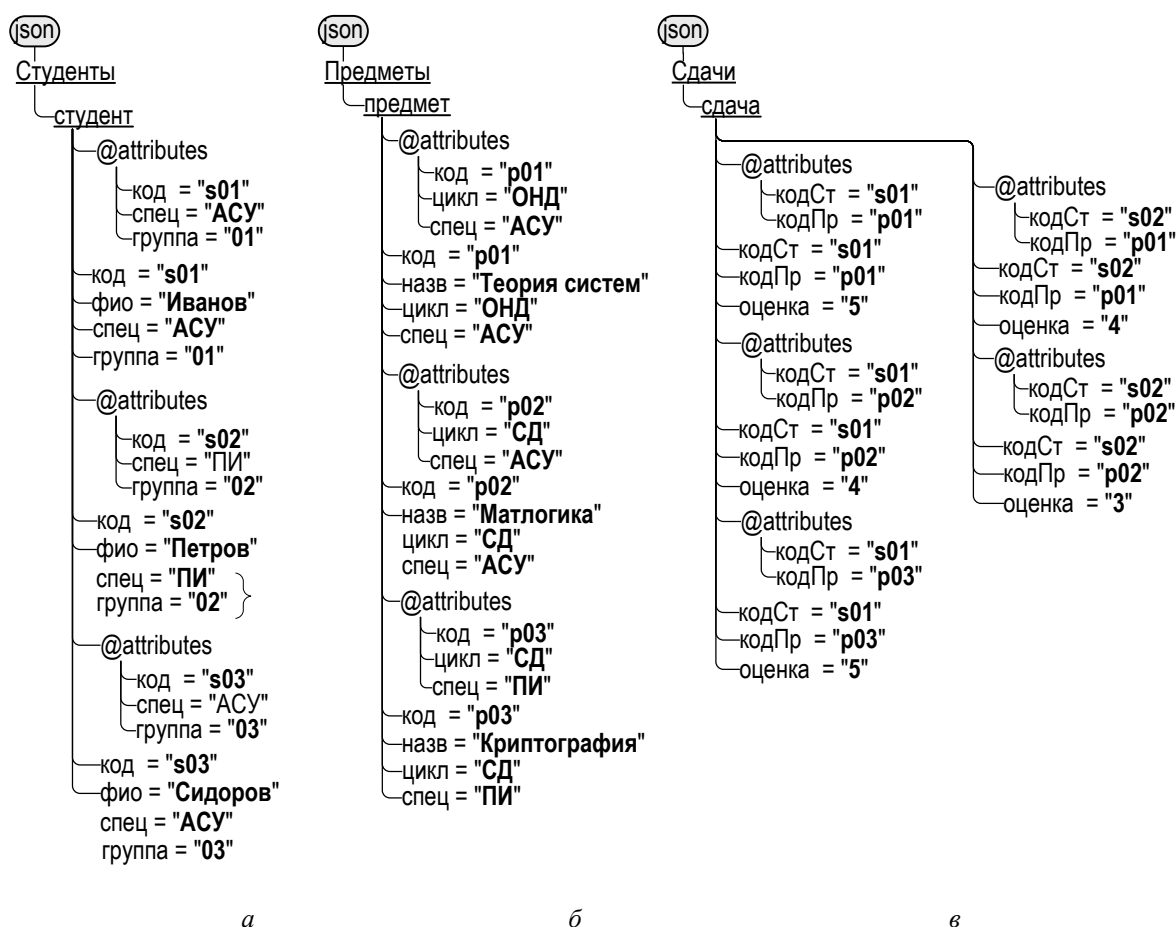


Рис. 5. Данные для формирования Smarty-объекта на основе тестовых экземпляров JSON-объектов

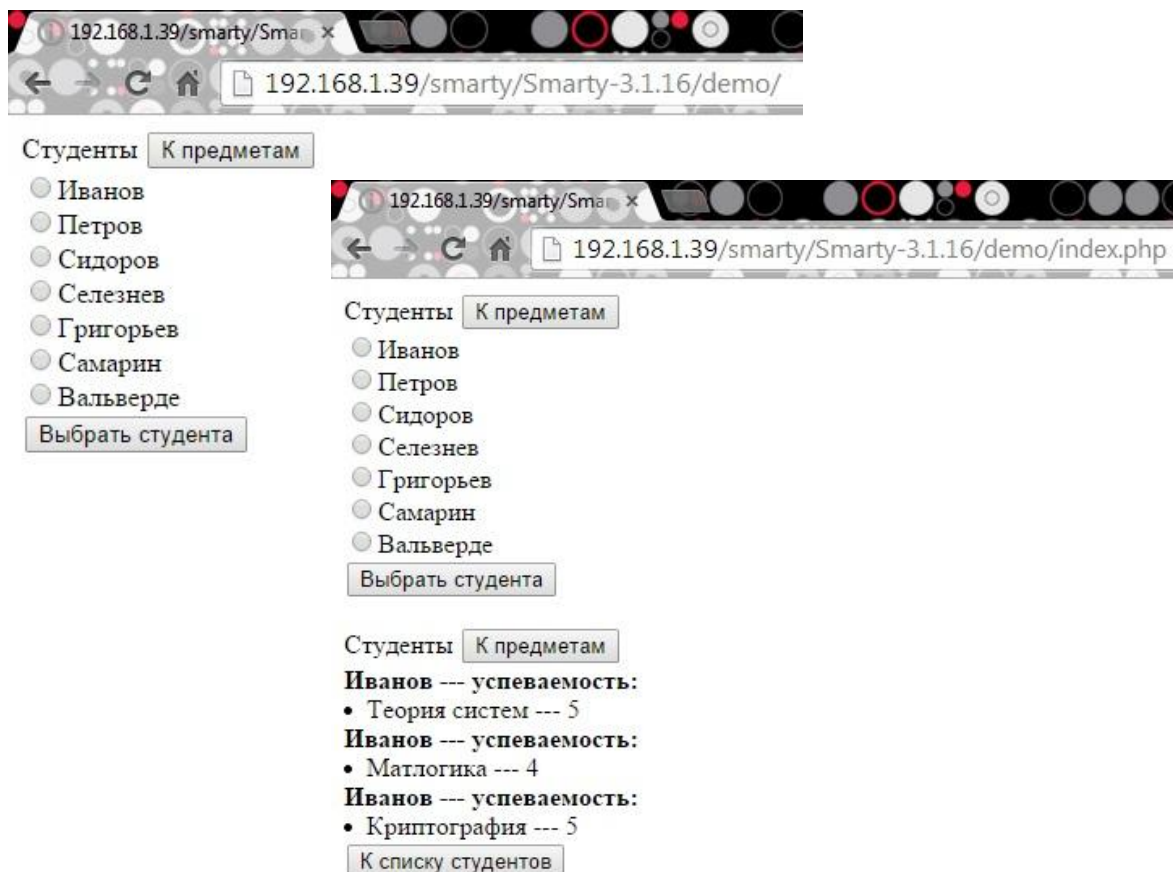


Рис. 6. Примеры экранных форм, сформированных на основе тестовых сведений из JSON-источника с помощью Smarty-объекта

Интерфейс пользователя. В рассматриваемом примере пользователю отображаются сведения из сформированного Smarty-объекта на основе JSON-данных. Начальное состояние отображает форму запроса выбранного студента, для которого с помощью Smarty-объекта запрашиваются данные по всем сдамкам. Результат выбора выводится в браузер. На рис. 6 приведены примеры форм, которые генерируются в результате выбора студента (в данном случае выбран студент Иванов, который имеет по результатам выбора 3 сдачи с полученными оценками по «теории систем», «матлогике» и «криптографии»). Первая форма рис. 6 демонстрирует список студентов в ситуации «СписокСтудентов», имеющих в JSON-файле, вторая форма появляется в результате нажатия кнопки «Выбрать студента» в ситуации «ВыбранСтудент». Возврат к предыдущей ситуации происходит при нажатии кнопки «К списку студентов». Кнопки «К предметам» на обеих формах предусмотрены для перехода к выбору предметов и запроса сведений из JSON-источника для вывода количества сдач по выбранному предмету. Аналогичные формы предусмотрены и для ситуаций

«СписокПредметов» и «ВыбранПредмет». Для выбора одного из предметов предусмотрены «радиокнопки». Задача данного интерфейса – отправка пользователю соответствующей формы в каждой из ситуаций, а также представление данных соединения сведений из JSON-источников.

Динамическая модель. Основой приложения служит модель нового уровня абстракции, которая оснащена новыми элементами и атрибутами. Динамическая модель данного приложения применительно к XML рассматривалась в работах [2–6]. Основными отличительными признаками модели веб-приложения в данной работе является задание документов JSON с помощью элементов "doc". Введены новые спецификации шаблонов типа tpl для отображения Smarty-объектов. Примеры данных средств приведены на рис. 7 в состоянии «Студенты-Предметы». Конструкция, обозначаемая символом `{tpl}`, указывает интерпретатору, что шаблон, запрашиваемый в подсостоянии «ВыбранСтудент» субмодели «Студенты», требует вывода Smarty-объекта «УспеваемостьСтудента» с помощью tpl-шаблона «УспеваемостьСтуден-

та», который находится на внешней памяти в папке шаблонов Smarty TPL/uspevstud.tpl. У каждого Smarty-объекта могут быть один или несколько вложенных JSON-источников, кроме того, они могут быть различного происхождения. Для этого был введен тип данных type, в котором указывается, откуда может быть принят к обработке контент. Самым распространенным типом данных источника, используемого в приложении, является массив (в модели он обозначается как array). Для приемника контента (символ **rcv**) предусмотрен новый метод вывода данных (method = "tpl"). Метод вывода tpl предполагает вывод Smarty-объекта с помощью компилируемого шаблона tpl.

Модель контента для загрузки Smarty-объекта. Контент, загружаемый в Smarty-объект, формируется на основе данных JSON-источников. Пример такой модели данных представлен на рис. 8. Сведения для загрузки

в Smarty-объект «СписокСтудентов» по студентам выбираются из документа doc «СписокСтудентов» по ссылке path="JSON/stud.json". Сведения для загрузки в Smarty-Объект «СписокПредметов» по студентам выбираются из документа doc «СписокПредметов» по ссылке path="JSON/predm.json". Для формирования результирующего контента (рис. 8, в, г) сведения о выбранном студенте извлекаются по кодам студента и предмета.

По этим же кодам выбираются сведения по сдамч: «ФИО студента», «специальность», «группа», «оценка» для ситуации «ВыбранСтудент» (рис. 8, а, б). Что же касается ситуации «ВыбранПредмет», выбираются дополнительные сведения: «название», «цикл», «специальность». Ключевые элементы могут быть перенесены в JSON-массив атрибутов, обозначаемый как @attributes, в этом случае выбор может быть произведен по атрибутам [15–18], более детально это отображено на (рис. 5, а, б, в).

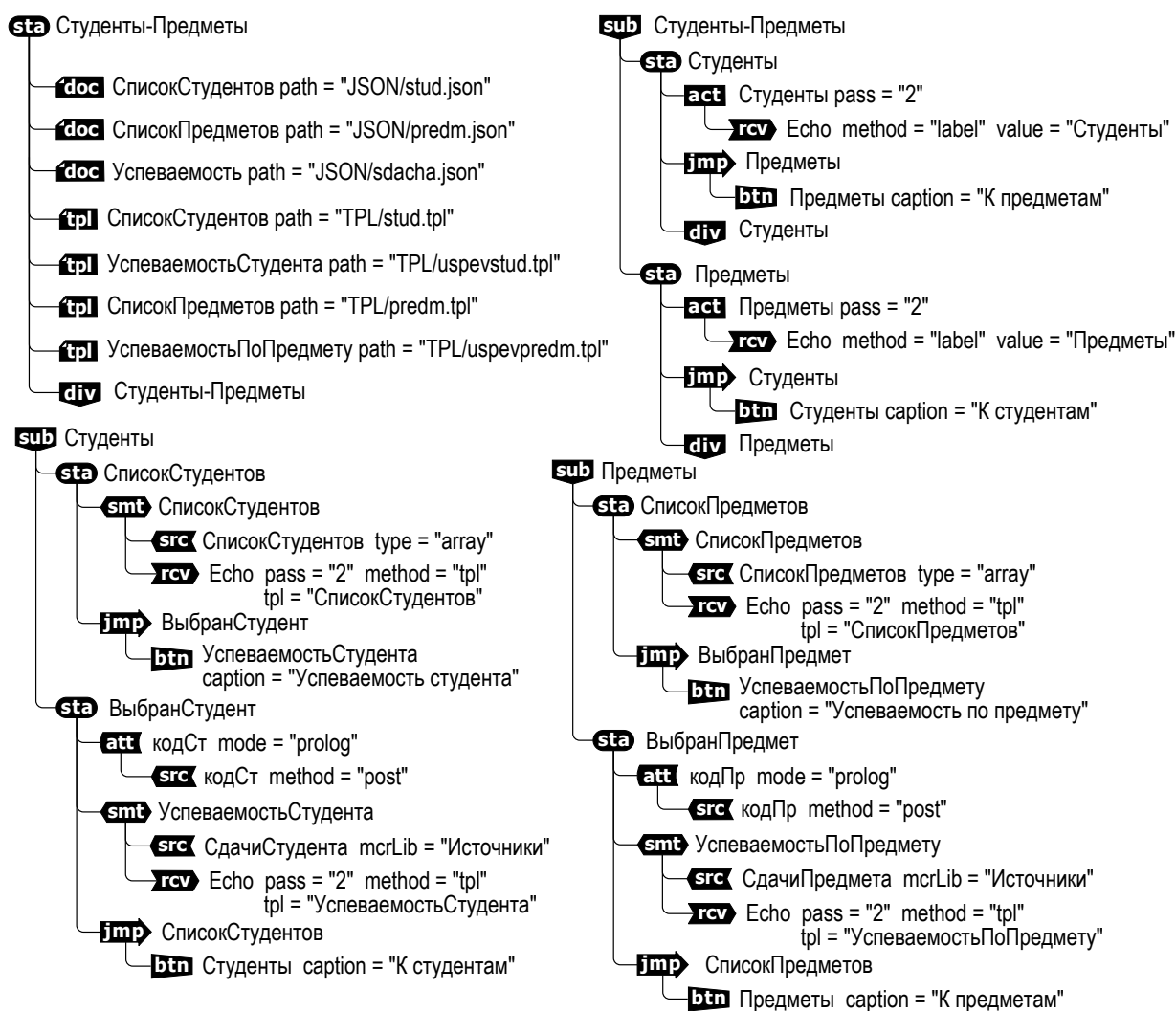


Рис. 7. Динамическая модель веб-приложения, оснащенная JSON-источниками и Smarty-объектами

Использование tpl-шаблонов для отображения контента Smarty-объекта. Суть использования tpl-шаблонов заключается в формировании Smarty-объекта, наполнении его актуальным контентом. В ситуации, когда все процедуры обработки завершены, происходит вывод контента в браузер клиента путем рендеринга объекта в шаблоне tpl. Шаблон tpl содержит спецификации на языке программирования PHP. В модели вместо XSLT-конструкции stylesheet используется Smarty-конструкция tpl, задающая имя шаблона для отображения контента.

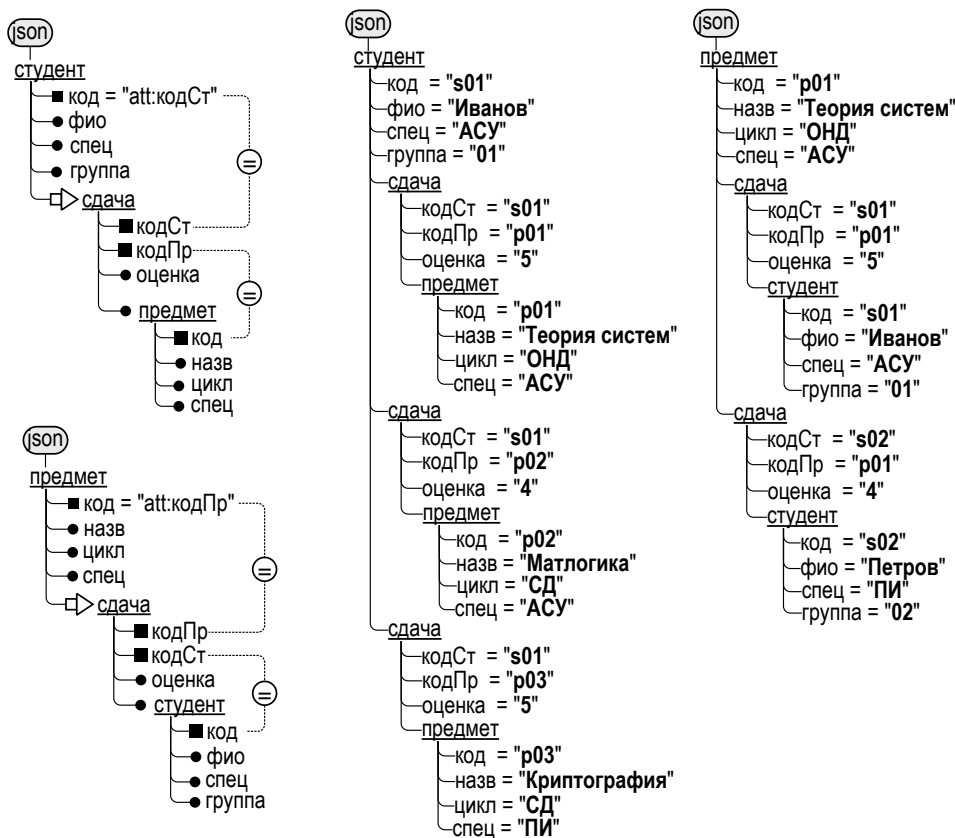
В рассматриваемом примере в исходном состоянии отображается список студентов, среди которых выбирается нужный студент для получения детальных сведений по сдачам и предметам. Список студентов сформирован на основе документа doc:СписокСтудентов. Для исходной ситуации «Студенты» предусмотрен шаблон tpl:СписокСтудентов. Шаблон компилируется, отображая каждого студента, концептуальная модель компиляции шаблона tpl: СписокСтудентов приведен на рис. 9, а.

При выполнении каждой итерации цикла FOREACH шаблона добавляется элемент INPUT, который содержит идентификационные сведения каждого студента. Элемент INPUT имеет тип type="radio", что отображается браузером в виде набора «радиокнопок» для выбора одного из студентов и кнопки BUTTON для перехода в следующее состояние «ВыбранСтудент», отражающее сведения об успеваемости выбранного студента. Для этой ситуации предусмотрен шаблон tpl:УспеваемостьСтудента, отображающий содержимое Smarty-объекта по выбранному студенту.

Концептуальная модель компиляции шаблона приведена на рис. 9, б. Результат компиляции второго шаблона «УспеваемостьСтудента» представляет собой список из элементов разметки LI сдач по каждому предмету.

ЗАКЛЮЧЕНИЕ

В данной статье рассмотрена задача использования JSON-контента в ситуационно-ориентированных базах данных, а также введены в динамическую модель конструкции, необходимые для обработки формирования JSON-контента.



а, б

в

г

Рис. 8. Контент Smarty-объекта, сформированный на основе сведений о студентах, предметах и сдачах из JSON-источников

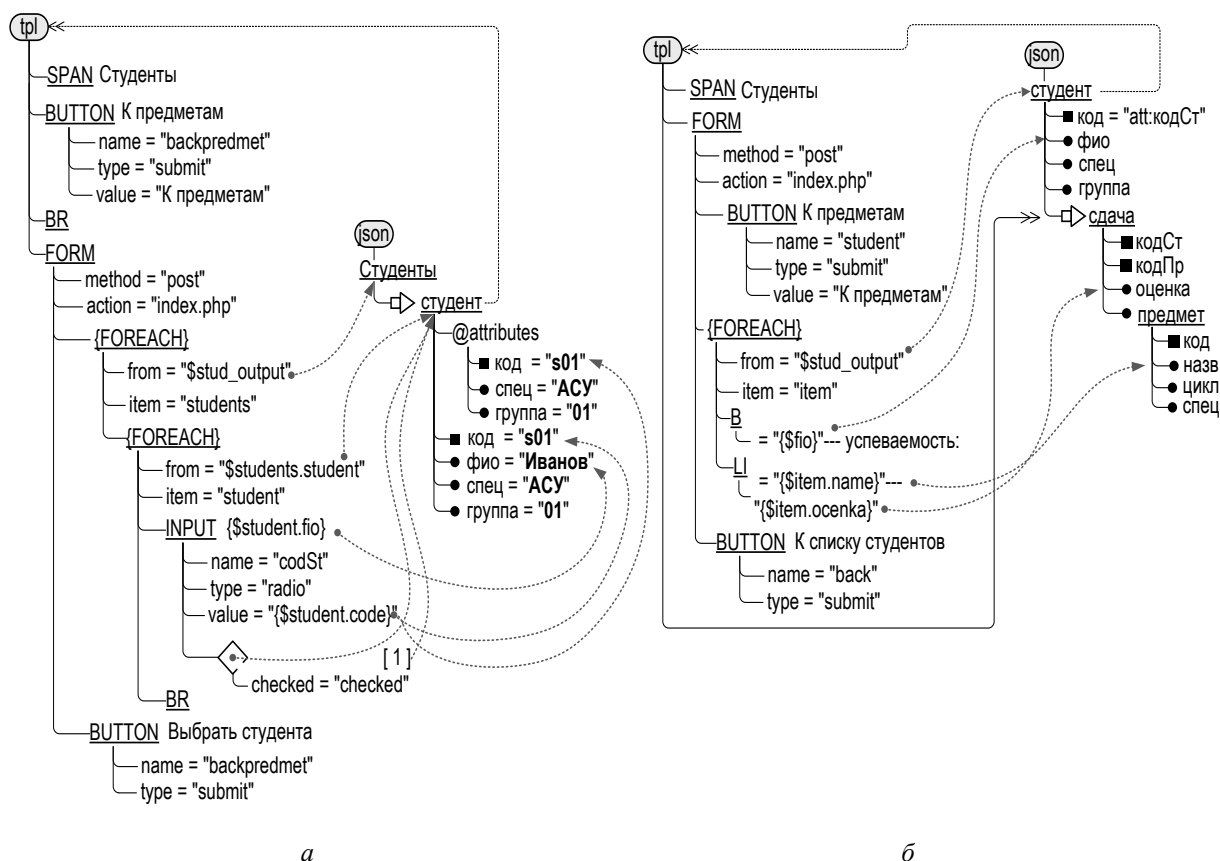


Рис. 9. Концептуальные модели компиляции шаблона TPL отправляемого в браузер на основе тестовых JSON-сведений из Smarty-объекта: *а* – формирование списка студентов; *б* – формирование списка сдач выбранного студента

Сформирована концепция, суть которой заключается в использовании Smarty-объектов для отображения JSON-результата выполнения функций. Примеры, приведенные и используемые в модели, рассмотрены без учета конкретной реализации на серверной платформе.

Предложено привязывать Smarty-объекты к состояниям динамической модели, при этом их создание в ходе интерпретации производится автоматически, осуществляется загрузка, фильтрация и очищение, а также слияние с учетом заданных условий. Конечный JSON-результат отображается с помощью заранее подготовленного шаблона tpl.

Основные отличительные признаки концепции:

- 1) элементы состояния динамической модели ситуационно-ориентированной базы данных содержат ассоциированные с ними Smarty-объекты, специфицируемые с помощью введенных в данной работе smarty-элементов;
- 2) в процессе интерпретации динамической модели объекты создаются автоматически, загружаются JSON-контентом, затем в соответствии с заданными условиями обработки формируется результирующий контент;

3) в динамической модели предусмотрены средства для подключения и обработки JSON-контента в ситуационно-ориентированной базе данных.

Применение этой концепции позволит разработчику проектировать приложения без трудоемкого программирования, подключая в состояниях динамической модели JSON-источники и специфицируя для них условия обработки для получения результирующего контента.

Дальнейшее развитие данного направления связано с разработкой программной реализации на платформе HSMI.

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). [С. Strauch and W. Kriha (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, "Situationally-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 situationally-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: автор. 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 situationally-oriented databases based on dynamic DOM-objects" 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 инфо-коммуникационные 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 situationally-oriented database based on dynamic DOM-objects" in *Proc. 8th Workshop of winter school postgraduates and young researchers*, Ufa: UGATU, 2013, vol. 1, pp. 115-118.]

7. **Канахин 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. **Канахин 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. **Канахин 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. **Канахин 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 data-bases," *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. **He W., Zhai J.** Application of the indent conversion based on XML and DOM // Paper presented at the Proceedings – 2013 International Conference on Computational and Information Sciences, ICCIS'2013, P. 411–413. [W. He and

J. Zhai, "Application of the indent conversion based on XML and DOM," *Paper presented at the Proceedings - 2013 International Conference on Computational and Information Sciences (ICCIIS' 2013)*, vol. 1, pp. 411-413, 2013.]

19. **Leveraging** spatial join for robust tuple extraction from web pages / Han, W. et al. // *Information Sciences*, (10 March 2014) 2014. V. 261, P. 132-148. [W. Han, et al., "Leveraging spatial join for robust tuple extraction from web pages", *Information Sciences*, vol. 261, pp. 132-148, 2014.]

20. **Pokorný J.** New database architectures: Steps towards big data processing // *Proceedings of the IADIS International Conference Intelligent Systems and Agents 2013 ECDM' 2013*, (Prague, Czech Republic, 22-24 July, 2013). Prague, Czech Republic, 2013. P. 3-10. [J. Pokorný, "New database architectures: Steps towards big data processing," in *Proc. IADIS International Conference Intelligent Systems and Agents (ECDM' 2013)*, Prague, Czech Republic, pp. 3-10, 2013.]

21. **Kaur K., Rani R.** Modeling and querying data in NoSQL databases // *Proc. of IEEE Int. Conf. on Big Data 2013* (Santa Clara, CA, United States, 6-October 2013). Santa Clara, United States, 2013. P. 1-7. [K. Kaur and R. Rani, "Modeling and querying data in NoSQL databases," in *Proc. of IEEE Int. Conf. on Big Data 2013*, Santa Clara, United States, pp. 1-7, 2013.]

22. **Static** and dynamic semantics of NoSQL languages / Benzaken, V. et al. // *40th Annual ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages POPL' 2013* (Rome, 23 through 25 January 2013) 2013. C. 101-113. [V. Benzaken, et al., "Static and dynamic semantics of NoSQL languages," (in English), in *Proc. 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL' 2013)*, pp. 101-113, Rome, Italy, 2013.]

23. **Design** and evaluation of lifelog mashup platform with NoSQL database / K. Takahashi et al. // *Proc. of 15th Int. Conf. Information Integration and Web-Based Applications and Services, iiWAS'2013* (Vienna, Austria, 2-4 Dec., 2013) C. 133-139. [K. Takahashi, et al., "Design and evaluation of lifelog mashup platform with NoSQL database", (in English), in *Proc. 15th Int. Conference on Information Integration and Web-Based Applications and Services (iiWAS -2013)*, pp. 133-139, Vienna, Austria, 2013.]

24. **Kobayashi N., Ishii M., Takahashi S., Mochizuki Y., Matsushima A., Toyoda T.** Semantic-JSON: a lightweight web service interface for Semantic Web contents integrating multiple life science databases // *Nucleic Acids Research*. 2011. T. 39. C. W533-W540. [N. Kobayashi, M. Ishii, S. Takahashi, Y. Mochizuki, A. Matsushima, T. Toyoda, "Semantic-JSON: a lightweight web service interface for Semantic Web contents integrating multiple life science databases," *Nucleic Acids Research*, vol. 39, pp. W533-W540, 2011.]

25. **O'Connor B. D., Merriman B., Nelson S. F.** SeqWare Query Engine: storing and searching sequence data in the cloud // *Bmc Bioinformatics*. 2010. T. 11. C. 9. [B. D. O'Connor, B. Merriman, S. F. Nelson, "SeqWare Query Engine: storing and searching sequence data in the cloud," in *Bmc Bioinformatics*, vol. 11, P. 9, 2010.]

26. **Sladic G., Milosavljevic B., Konjovic Z., Vidakovic M.** Access Control Framework for XML Document Collections // *Computer Science and Information Systems*. 2011. T. 8, № 3. C. 591-609. [G. Sladic, B. Milosavljevic, Z. Konjovic, M. Vidakovic, "Access control framework for XML document collections" (in English), *Computer Science and Information Systems*, vol. 8, no. 3, pp. 591-609, 2011.]

27. **Sudarsan R., Gray J.** Metamodel search: Using XPath to search domain-specific models // *Journal of Research and Practice in Information Technology*. 2006. T. 38, № 4. C. 337-351. [R. Sudarsan, J. Gray, "Metamodel search: Using XPath to search domain-specific models," *J. Research and Practice in Information Technology*, vol. 38, no. 4, pp. 337-351, 2006.]

28. **Tang N., Yu J. X., Wong K. F., Li J. X.** Fast XML structural join algorithms by partitioning // *Journal of Research and Practice in Information Technology*. 2008. T. 40, № 1. C. 33-53. [N. Tang, J. X. Yu, K. F. Wong, J. X. Li, "Fast XML structural join algorithms by partitioning," *J. Research and Practice in Information Technology*, vol. 40, no. 1, pp. 33-53, 2008.]

29. **Chen Y. J., Huck G.,** On the evaluation of path-oriented queries in document databases // *Database and Expert Systems Applications*. 2001. T. 2113. C. 953-962. [Y. J. Chen, G. Huck, "On the evaluation of path-oriented queries in document databases," *Database and Expert Systems Applications*, vol. 2113, pp. 953-962, 2001.]

30. **Dekeyser S., Hidders J., Paredaens J.** A transaction model for XML databases // *World Wide Web-Internet and Web Information Systems*. 2004. T. 7, № 1. C. 29-57. [S. Dekeyser, J. Hidders, J. Paredaens, "A transaction model for XML databases," *World Wide Web-Internet and Web Information Systems*, vol. 7, no. 1, pp. 29-57, 2004.]

31. **Herskovits E. H., Chen R.** Integrating data-mining support into a brain-image database using open-source components // *Advances in Medical Sciences*. 2008. T. 53, № 2. C. 172-181. [E. H. Herskovits, R. Chen, "Integrating Data-Mining Support into a Brain-Image Database Using Open-Source Components," *Advances in Medical Sciences*, vol. 53, no. 2, pp. 172-181, 2008.]

32. **Jea K. F., Chang T. P., Chen S. Y.** A semantic-based protocol for concurrency control in DOM database systems // *Journal of Information Science and Engineering*. 2009. T. 25, № 5. C. 1617-1639. [K. F. Jea, T. P. Chang, S. Y. Chen, "A semantic-based protocol for concurrency control in DOM database systems," *J. Information Science and Engineering*, vol. 25, no. 5, pp. 1617-1639, 2009.]

33. **Kudrass T., Conrad M.,** Management of XML documents in object-relational databases // *Xml-Based Data Management and Multimedia Engineering-Edbt 2002 Workshops*. 2002. T. 2490. C. 210-227. [T. Kudrass, M. Conrad, "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.]

34. **Nassis V., Dillon T. S., Rajagopalapillai R., Rahayu W.** An XML document warehouse model // *Database Systems for Advanced Applications: Proceedings*. 2006. T. 3882. C. 513-529. [V. Nassis, T. S. Dillon, R. Rajagopalapillai, W. Rahayu, "An XML document warehouse model," in *Database Systems for Advanced Applications, Proceedings*, vol. 3882, pp. 513-529, 2006.]

35. **Batory D.** Multilevel models in model-driven engineering, product lines, and metaprogramming // *IBM Systems Journal*. 2006. T. 45, № 3. C. 527-539. [D. Batory, "Multilevel models in model-driven engineering, product lines, and metaprogramming," *IBM Systems Journal*, vol. 45, no. 3, pp. 527-539, 2006.]

36. **Dejanovic I., Milosavljevic G., Perisic B., Tumbas M. A.** Domain-specific language for defining static structure of database applications // *Computer Science and Information Systems*. 2010. T. 7, № 3. C. 409-440. [I. Dejanovic, G. Milosavljevic, B. Perisic, M. A. Tumbas, "Do-

main-Specific Language for Defining Static Structure of Database Applications,” in *Computer Science and Information Systems*, vol. 7, no. 3, pp. 409-440, 2010.]

37. **Djukic V., Lukovic I., Popovic A., Ivancevic V.** Model execution: an approach based on extending domain-specific modeling with action reports // *Computer Science and Information Systems*. 2013. Т. 10, № 4. С. 1585–1620. [V. Djukic, I. Lukovic, A. Popovic, V. Ivancevic, “Model Execution: An Approach based on extending Domain-Specific Modeling with Action Reports,” *Computer Science and Information Systems*, vol. 10, no. 4, pp. 1585-1620, 2013.]

38. **Panach J. I., Juristo N., Pastor O.** Including Functional Usability Features in a Model-Driven Development Method // *Computer Science and Information Systems*. 2013. Т. 10, № 3. С. 999-1024. [J. I. Panach, N. Juristo, O. Pastor, “Usability Features in a Model-Driven Development Method,” *Computer Science and Information Systems*, vol. 10, no. 3, pp. 999-1024, 2013.]

39. **Strosnider J. K., Nandi P., Kumaran S., Ghosh S., Arsanjani A.** Model-driven synthesis of SOA solutions // *IBM Systems Journal*. 2008. Т. 47, № 3. С. 415-432. [J. K. Strosnider, P. Nandi, S. Kumaran, S. Ghosh, A. Arsanjani, “Model-driven synthesis of SOA solutions” (in English), in *IBM Systems Journal*, vol. 47, no. 3, pp. 415-432, 2008.]

ОБ АВТОРАХ

ГУСАРЕНКО Артем Сергеевич, асс. каф. автоматизированных систем управления. Канд. техн. наук (УГАТУ, 2013). Дипл. информатик-экономист (УГАТУ, 2010). Иссл. в обл. иерархических моделей и ситуационного управления.

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

METADATA

Title: Smarty-objects: use case of heterogeneous sources in situation-oriented databases.

Authors: A. S. Gusarenko¹, V. V. Mironov²

Affiliation:

Ufa State Aviation Technical University (UGATU), Russia.

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

Language: Russian.

Source: Vestnik UGATU (scientific journal of Ufa State Aviation Technical University), vol. 18, no. 3 (64), pp. 242-252, 2014. ISSN 2225-2789 (Online), ISSN 1992-6502 (Print).

Abstract: As part of the previously developed concept model DOM-elements in the dynamic model of situational-oriented database dynamic model proposed higher level of abstraction providing for visual and text tools specifications of data sources based on JSON. Specifications describes a method using JSON sources, JSON-elements associated with the states of the dynamic model, which regulates the filling Smarty-object and/or Smarty-array by JSON-content of the specified source when the parent state becomes the current state. It is proposed to implement the model on the platform PHP using tpl-templates of Smarty.

Key words: Web-application; dynamic model; NoSQL; JSON; 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).