МНОГОНАПРАВЛЕННЫЕ ПРЕОБРАЗОВАНИЯ И ВЕБ-ПРЕДСТАВЛЕНИЯ РАЗНО-СТРУКТУРИРОВАННОЙ ИНФОРМАЦИИ

Колодин М.Ю.

УДК 006.72

Колодин М.Ю. **Многонаправленные преобразования и веб-представления разно-структурированной информации.**

Аннотация. Все более актуальным становится получение разнообразных представлений информации на основе одних и тех же наборов данных, прежде всего в Интернете и в интранет-системах. Одним из перспективных способов получения таких представлений является выполнение многоуровневых метапреобразований исходных данных в соответствии с динамически задаваемыми требованиями.

Ключевые слова: метасистемы, веб-представления, структурированная информация, преобразователи данных.

Kolodin M.Y. **Multidirectional transformations and Web-representations of differently-strtuctured information.**

Abstract. Various representations of the same source data are very actual in Internet and intranet systems. One of the perspective ways to obtain such representations is the method of multi-level metatransformations of the given data according to dynamically set requirements.

Keywords: metasystems, Web-representations, structured information, data transformers.

1. Введение. В последние годы все более актуальной становится задача многократного использования одних и тех же данных, получение многочисленных различных видов выходной информации на основе одинаковых однократно вводимых наборов исходной информации, причем во всех случаях она может быть различным образом структурирована [1, 5]. Такая задача возникает и в работе с локальными базами данных, и с Интернет-ресурсами, и в управляющих и информационных интранет-системах. Из этого следует, что нужно искать более удобные форматы и соответствующие им инструменты представления и преобразования данных помимо широко распространенных представлений в виде XML [2] и HTML.

Цель данного исследования состоит в разработке способов оптимального заполнения, преобразования и, главное, передачи и выдачи различным образом структурированной информации на основе таких наборов. Наиболее важными здесь являются задачи выбора оптимального представления данных, особенно для случаев данных больших объемов, данных переменной структуры, неполных данных, и построение инструментов для их преобразования, в том числе для показа в выдаче веб-браузеров.

2. Типичные примеры: «вуз» и «архив». Рассмотрим два примера, на которых будет хорошо видна сущность проблемы.

Пример 1. Есть вуз, в нем работает администрация; преподаватели разных кафедр нескольких факультетов обучают студентов, объединенных в группы, по нескольким специальностям, курсам, потокам, годам обучения; у студентов есть руководители, у кафедр и факультетов есть руководители и т. п. Нужно учитывать все эти объекты, а также занятия, оценки, зачеты и экзамены. Понятно, что попытка вывести для пользователя сразу всю информацию обречена на провал: даже если удастся выдать все данные, их будет невозможно воспринять даже зрительно. Следовательно, нужно в каждый момент предоставлять пользователю только то, что ему требуется, скрывая детали до тех пор, пока он явно или неявно их не запросит сам.

Пример 2. Имеется музыкальный (поэтический и т. п.) архив. Есть авторы и исполнители песен, различные объединения авторов; клубы и концертные площадки, где исполняются песни, характеризуемые названием, первой строкой, текстом, авторством текста, музыки и песни в целом; причем у строк есть варианты, у каждого исполнения есть свои атрибуты, в том числе вариативные, события (фестивали, концерты, пр.); у каждой архивной единицы хранения есть указания на все вышеуказанное, а также хозяин архива, авторы и технические параметры записей, оцифровок; для всего этого указаны места и даты, причем они могут быть указаны неточно, неполно, в диапазонах или вообще отсутствовать; есть и другие сведения. В настольном варианте такая база содержит десятки и сотни тысяч строк в таблице с более чем 100 полями, а в полном — реляционную базу данных с более чем 100 таблицами; полный объем информации в одной из таких баз, сейчас находящейся в эксплуатации, превышает 55 ТБ, а с учетом дополнительных распределенных и находящихся в разных городах баз — до 70 ТБ (включая описи и собственно материалы), и этот объем ежедневно растет.

Задача состоит в получении оперативного доступа к разнообразным выборкам из всей имеющейся информации, в том числе по сети (локально и через Интернет), причем существенно важно удобство обработки ее пользователем, не являющимся программистом.

3. Обычный подход к преобразованию и визуализации информации. Традиционно для обработки информации такого рода создаются реляционная база данных и набор форм с заранее заданными запросами к базе данных с точно определенными форматами вывода.

Для *примера 1* потребуется отдельно рассматривать вывод данных о студентах по группам, по каждому студенту — вывод его показателей, по каждой группе — вывод сводной информации по каждому экзамену и т. п.

Для *примера 2* вообще нет общепринятых наборов выходных данных: все нужно формулировать и рассчитывать заново. Действительно, данные структурированы и физически расположены в базах различным образом, запросы носят в общем случае непредсказуемый характер, объемы потенциально выдаваемой информации различаются на многие порядки, предпочтительные способы ее визуализации субъективно различны у различных пользователей, велико количество ошибок пользователей при обращении к системе.

Недостатком классического подхода является необходимость ручного программирования обработки и вывода для каждого запроса, следствием чего являются ограниченный набор таких запросов, большая длительность цикла разработки и отладки при изменении наборов данных и появлении новых запросов на типы выдачи. В то же время гарантируется достаточно высокое качество результата для простых запросов. Но совсем сложно учесть и адекватно вывести информацию, наличие, объем и тип которой заранее неизвестны.

В обоих случаях реляционные таблицы сильно разрежены, имеют большой объем в случае, если они реализованы централизованно; обращение к данным в базе сильно различается по частоте и сложности запросов: некоторые части изменяются постоянно, некоторые остаются неизменными годами или даже заполняются однократно и никогда не изменяются.

Необходимо реализовать более гибкий подход с обеспечением приемлемого качества результата.

4. Метаподход с элементами динамической обработки. Заметим, что во всех указанных (и многих других) случаях мы имеем дело с различным образом гиперструктурированной информацией и хотим получать ее срезы в удобном виде. Таким образом, важно знать следующее:

— какая именно информация используется;

— как информация обрабатывается, т. е. передается и преобразуется, как она представлена пользователю или другой компьютерной системе;

— когда (на каких этапах своего жизненного цикла) информация обрабатывается.

Есть и другие вопросы, требующие ответа, например, затраты ресурсов (процессорного времени, сетевой пропускной способности и т. п.).

Для *примера 1* интересное решение найдено в Новосибирском университете (руководитель разработки — проф. А. Г. Марчук, ИСИ СО РАН) [1]; в этом решении активно используются описания схем данных XML Schema и семантических отношений по RDF. Это решение можно развить и объединить с другими, прежде всего, в части динамического изменения моделей структур данных (не самих данных, а их представлений), структур запросов и соответственно структур вывода [3]. Действительно, каждый запрос на обработку или вывод информации основывается на некой явной или подразумеваемой иерархии данных, т. е. на их вложенности и перекрестных ссылках. Используя эту иерархию, можно строить запросы; изменяя иерархию, можно автоматически получать новые запросы. Наконец, имея правила построения иерархий, можно практически полностью управлять процессами поиска, выборки, обработки и выдачи информации пользователю.

Формат задания таких описаний может быть различен, от простых пар ключ—значение (возможно, с группами) до XML-подобных структур (но не собственно на языке XML, а на упрощенных языках типа YAML или JSON, которые к тому же легко встраиваются в программы-обработчики без вызова дополнительных XML-преобразователей; это возможно, поскольку нет необходимости в использовании всех возможностей XML).

5. Практические приемы и решения. В ходе работ по этой теме найдены несколько решений и перспективных направлений исследований. Прежде всего это использование имеющейся файловой системы для организации данных. Каждый каталог рассматривается как замкнутый информационный блок. В каждом блоке есть свой файл с метаописанием всего, что в нем находится. При переносе, копировании, модификации и удалении данных в каталоге и всего каталога соответствующая информация обновляется и включается в общую базу метаданных на уровне компьютера. Таким образом, мы легко переносим части базы данных, синхронизируем ее с остальными частями базы. Важно также, что метаописание включает в себя сведения о структуре данного блока, об иерархии входящих в него данных; таким образом, мы можем относительно легко сочетать различноструктурированные блоки [4].

Использование ссылок в операционных системах (ОС) семейства MS Windows в общем случае себя не оправдало в силу ограниченной и несколько странной реализации этого аппарата; поэтому ссылки применяются только для соединения однотипных частей архива, находящихся в том числе на разных физических носителях. Эта мера была введена скорее для удобства, тем более, что в ОС данного семейства многие программы работают со ссылками некорректно. В ОС семейства GNU/Linux проблем такого рода при работе со ссылками не было.

Предлагается оптимизация по времени поиска и обработки и сетевой пропускной способности. Сами большие архивы полностью копировать между всеми компьютерами нет возможности и необходимости, тем более что период их обновления существенно различен — от реального промежутка времени с быстрым откликом (не более нескольких секунд) до переносов больших объемов данных лишь раз в несколько лет. Нужны переносимые описания и метаописания данных. Это реализуется в виде блоков-каталогов, автоматизированно переносимых между компьютерами, в том числе по сети. Получается «большая система», в которой нет ни одного компьютера с полностью актуальной информацией, но которая в целом все же корректно функционирует и выдает правильные результаты; эта тема еще требует изучения.

Для веб-архивов можно пользоваться собственными программами препроцессирования информации [5] и встроенными в каждый браузер каскадными таблицами стилей. Полезный прием — сочетание стилей, при котором каждый уровень применяемой в данный момент иерархии дает свой класс форматирования, а веб-браузер по сформированной таким образом таблице стилей показывает требуемую информацию; при этом перечень сочетаний стилей может быть неполным, и браузер самостоятельно определит нужный стиль по имеющимся сведениям.

Для динамических веб-систем, которые функционируют в течение достаточно длительного времени [6], можно указать еще такой прием: учет реакции пользователя в дополнение к изначальным и его собственным настройкам (т. е. поскольку не всегда можно точно знать, в каком формате для данного набора данных пользователю удобно получить результат, можно предложить несколько форматов вывода и запомнить на будущее тот, который в конце концов выберет пользователь). Интересно также применять обновление системы по явному или неявному запросу пользователя: пусть некая часть «большой» системы не находится в актуальном состоянии, и к ней выполняется запрос; система дает ответ на основании имеющихся у нее в данный момент

сведений и после этого сама отправляет запрос к соответствующим подсистемам или к удаленным системам на актуализацию нужной порции информации; при следующем запросе пользователь получит более точный ответ.

6. Оценка эффективности метаподхода. После проведенных исследований можно сказать, что для систем «среднего» размера (содержащих до нескольких тысяч строк исходного кода, до сотен тысяч записей в базе данных) метод работает хорошо. Для «малых» систем (до нескольких сотен строк исходного кода, до нескольких тысяч записей в базе данных) затраты на построение многоуровневой системы часто не окупаются, особенно если выполняется разовая работа. Для оценки «больших» систем (параметры которых больше, чем указанные выше «средние», а также существенно распределенных и не доступных одновременно) нужны дополнительные эксперименты; однако применение метода использования метаописаний вместо собственно данных дает заметный положительный эффект. В последующих работах на основе одной и той же технологии использование уже выделенного метааппарата существенно ускоряет получение результата. Однако на первых порах и в «малых» системах накладные расходы на разработку метааппарата пока еще слишком велики; в таких случаях полезно не выписывать метапрограммы явно, а лишь выполнять проектирование в терминах метасистем, оставляя детали реализации дополнительных уровней до того времени, когда расширяющаяся система потребует такой детализации. Численно в типичных случаях экономия времени разработки составляла около 25–30% (единица учета — человеко-часы работы программиста-разработчика; сравнивалось время на проектирование и написание одинаковых по функционалу программ), время выполнения программ во всех случаях было примерно одинаковым и практически ничтожным по сравнению с затратами на работу окружения. Например затраты времени на работу веб-браузера, передачу данных между клиентом и сервером, работу веб-сервера, отправку статического HTML обратно клиенту и отображение информации на экране пользователя на порядки больше времени выполнения программ на исследуемом «метаэтапе». Однако для локальных вычислительных работ нужно получить более точные оценки (они будут выполнены по мере появления соответствующих задач и совершенствования методов оценивания); кроме того, нужно найти способы снижения затрат на первоначальное построение метасистем, а также повысить их практическую применимость для систем «малого» размера.

7.Заключение. Использование метаинформации, метапредставлений и метапреобразований данных показало свою полезность. Для некоторых наиболее актуальных приложений удалось найти подходящие форматы и их программные преобразователи. Дальнейшие исследования будут направлены на работу с «большими» системами. Особенно важно получить количественные оценки эффективности в каждом из изучаемых случаев и более точно определить границы указанных выше классов систем по размеру и сложности. Интересно также построить теоретические и практические решения для систем с динамически изменяющейся структурой данных.

Литература

1. *Марчук А.Г., Марчук П.А.* Архивная фактографическая система // Тр. 11 Всерос. науч. конф. «Электронные библиотеки: перспективные методы и технологии, электронные коллекции — RCDL-2009». Петрозаводск, 2009. С.177–185.
2. *Валиков А.Н.* Технология XSLT. СПб: БХВ-Петербург, 2002. 544 с.
3. *Колодин М.Ю.* Инструментальные средства разработки, реализации и сопровождения гипертекстовых преобразователей // Тр. СПИИРАН. 2008. Вып. 7. С. 64–69.
4. *Колодин М.Ю.* Синтаксические и семантические особенности метасистем // Тр. СПИИРАН. 2009. Вып. 9. С. 168–177.
5. *Колодин М.Ю.* Межформатные преобразования гипертекста // Тр. СПИИРАН. 2008. Вып. 6. С. 168–170.
6. *Колодин М.Ю.* Разработка, реализация и сопровождение веб-сайта научной организации // Тр. СПИИРАН. 2003. Вып. 1, т. 3. С. 217–223.

Колодин Михаил Юрьевич — научный сотрудник исследовательской группы информационных технологий в образовании (ИГИТО) Учреждения Российской академии наук Санкт-Петербургский институт информатики и автоматизации РАН (СПИИРАН). Область научных интересов: метасистемы, свободное программное обеспечение, образовательные технологии. Число научных публикаций — 74. myke@iias.spb.su, www.myke.spb.ru; СПИИРАН, 14-я линия В.О., д.39, Санкт-Петербург, 199178, Россия; р.т. +7(812)328-0382. Научный руководитель — канд. физ.-мат. наук, доцент, ведущий научный сотрудник А.Л. Тулупьев.

Kolodin Mikhail Yurievich — researcher, Research Group for Information Technologies in Education (RGITE) of Institution of the Russian Academy of Sciences St. Petersburg Institute for Informatics and Automation RAS. Research interests: metasystems, free and open source software, educational technologies. The number of publications — 74. myke@iias.spb.su, www.myke.spb.ru; SPIIRAS, 39, 14th Line V.O., St.Petersburg, 199178, Russia; office phone +7(812)328-0382. Scientific supervisor — Ph.D., associate professor, leading researcher A.L. Tulupiev.

Рекомендовано группой междисциплинарных проблем информатики СПИИРАН. Научный руководитель А.Л. Тулупьев, канд. физ.-мат. наук, доцент, ведущий научный сотрудник.
Статья поступила в редакцию 14.12.2009.

РЕФЕРАТ

Колодин М. Ю. **Многонаправленные преобразования и веб-представления разно-структурированной информации.**

В последние годы все более актуальной становится задача многократного использования одних и тех же данных, получение многочисленных различных видов выходной информации на основе одинаковых однократно вводимых наборов исходной информации, причем во всех случаях она может быть различным образом структурирована. Это применяется и в работе с локальными базами данных, и с Интернет-ресурсами, и в управляющих и информационных интранет-системах.

Цель исследования состоит в разработке способов оптимального заполнения, преобразования и, главное, передачи и выдачи различным образом структурированной информации на основе таких наборов. Наиболее важными здесь являются задачи выбора оптимального представления данных, особенно для случаев данных больших объемов, данных переменной структуры, неполных данных, и построение инструментов для их преобразования, в том числе для показа в выдаче веб-браузеров.

На типовых примерах «вуз» и «архив» рассмотрены основные требования, трудности и способы решения поставленной задачи.

Есть несколько полезных приемов, прежде всего это использование имеющейся файловой системы для организации данных, применение описателей для информационных блоков на уровне каталогов, в том числе определяющих структуру находящейся в данном блоке информации, что позволяет правильно выделить и отобразить информацию, согласовать ее с информацией из других блоков с такой же или иной структурой и содержимым. Использование ссылок файловой системы было полезно при работе с ОС семейства Linux, но не вполне успешно для ОС MS Windows.

Очень полезным оказалось выделение из архивов метаинформации, с последующим обменом между серверами только метаинформации (это сведения о наличии некоторой информации определенного типа в архиве на данном сервере, краткий или полный перечень такой информации по некоторым признакам), с полуавтоматическим обновлением такой информации.

Удачной оказалась реализация представления данных на основе смеси стилей CSS. Включение информации и метаинформации на упрощенных языках типа YAML и JSON также способствовало повышению гибкости и быстродействия системы выборки и представления информации.

В целом экономия времени разработки в типовых случаях составила примерно 25–30% традиционного; однако это справедливо только для систем «среднего» размера; для «малых» и «больших» систем нужно провести дополнительные исследования. Нужно также более строго определить способы измерения эффективности и опробовать их для «больших» систем.

SUMMARY

Kolodin M. Y. **Multidirectional transformations and Web-representations of differently-strtuctured information.**

During the last years the problem of multiple usage of the same data becomes still more actual; we need to process and represent data in various formats, basing on the data that are entered once; moreover, these data may also be differently structured.

It is used both in work with local databases and with Internet resources, in control and information intranet systems.

The purpose of this research is to optimally fill, transform and, what is even more important, to send and output differently structured information based on such data sets. The most actual are the problems of selection of optimal data representation formats, especially for the cases of data of big size, data of variable structure, incomplete data, as well as building instruments for their transformation, including output for Web browsers.

The principal requirements, difficulties and ways of solving the given problem are studied on typical examples of «institute» and «archive».

There are several useful approaches to better organize work and solve the problem set. First of all, it is usage of file system for data organization, usage of descriptors for information blocks on the level of folders, including those that define the structure of information that is placed in the current block; it allows to correctly select and represent the information, properly connect it with information form other blocks with the same or other structure and contents. Usage of hard and soft file links was useful in operating systems of Linux family, but not so successful for MS Windows.

The principle of selection of metainformation from archives, with subsequent interchange of only such metainformation between servers was very useful; these are metadata about presence of some information in the archive on the given server, brief or full list of information items on some fields; with semiautomatic renewal of such information.

The data representation implementation based on CSS mix was also rather useful. Inclusion of information and metainformation in simplified languages like YAML and JSON also helped to improve flexibility and speed of information selection and representation system.

In general, the economy of development time in typical cases was about 25–30% of the traditional one; but is actual only for «middle» size systems; for «small» and «big» size systems additional study is necessary. We should also more accurately define ways of efficiency measurement and perform them for «big» systems.