

УДК 004.738.5:004.65

ФУНКЦИИ АНАЛИТИКИ В ВЕБ-ПРИЛОЖЕНИЯХ НА ОСНОВЕ СИТУАЦИОННО-ОРИЕНТИРОВАННЫХ БАЗ ДАННЫХ

Е. С. МАКАРОВА¹, В. В. МИРОНОВ²

¹es_makarova@list.ru, ²mironov@list.ru

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

Поступила в редакцию 22.04.2013

Аннотация. Рассматривается организация функций аналитики в веб-приложениях, функционирующих на основе серверных ситуационно-ориентированных баз данных (СОБД) и OLAP-клиентов. Обсуждается взаимодействие СОБД и OLAP-клиента путем формирования промежуточной контент-таблицы. Строится схема формирования контент-таблицы из XML-документов, хранящихся в СОБД. Предлагаются дополнительные компоненты динамической модели СОБД, облегчающие организацию сервиса для формирования контент-таблицы. Полученные результаты иллюстрируются на примере реализации функций аналитики на базе OLAP-клиента FlexMonster Pivot Table & Charts Component и контент-таблицы в формате CSV в веб-приложении, обслуживающем деятельность диссертационных советов вуза.

Ключевые слова: веб-приложение; ситуационно-ориентированная база данных; динамическая модель; XML; XSLT; CSV; Web OLAP; OLAP-клиент; FlexMonster; информатизация диссертационных советов

Оперативная аналитическая обработка данных (On-Line Analytical Processing – OLAP) ориентирована на проведение сложного анализа больших объемов данных с высокой производительностью. Традиционно для этого строились крупные хранилища данных со специальной организацией и предварительной агрегацией информации. В настоящее время с появлением достаточно универсальных OLAP-клиентов возможности оперативной аналитической обработки стали доступны и для мало- и среднемасштабных информационных систем. В связи с этим появились новые термины: «настольный OLAP», «Web OLAP» и т. п. [1]. Так, настольный OLAP предполагает построение и хранение гиперкуба на компьютере пользователя, позволяя проводить анализ данных в однопользовательском режиме. Web OLAP предполагает выполнение аналитических функций в многопользовательском режиме через Веб. В обоих случаях формирование гиперкубов, как правило, происходит «на лету» («on the fly»), из базы данных OLTP без предварительной агрегации.

В работе [2] авторами на концептуальном уровне (без учета особенностей и возможностей

среды реализации) предложен подход к проектированию многомерной модели данных для решаемых «на лету» задач Web OLAP. Подход отличается тем, что за основу берется ER-модель базы данных OLTP, в ней дополнительно вычленяются в отдельные сущности – атрибуты, существенные в плане анализа, после чего связи типа «многие-ко-многим» рассматриваются как потенциальные гиперкубы.

Дальнейшие исследования в этом направлении связаны с переходом с концептуального на логический уровень абстракции, учитывающий особенности и возможности платформы реализации базы данных. В качестве такой платформы авторами рассматриваются ситуационно-ориентированные базы данных (СОБД) [3, 4], в которых данные хранятся в виде документов в формате XML и обрабатываются в контексте текущего состояния динамической модели.

Необходимо построить отображение многомерной модели OLAP-куба на XML-модель хранилища документов СОБД, а также создать алгоритмическое и программное обеспечение, формирующее гиперкубы «на лету» для используемого OLAP-клиента.

В данной статье рассматривается, как в веб-приложениях, основанных на СОБД [5–7],

в контексте текущей ситуации сформировать «на лету» гиперкуб и загрузить его в OLAP-клиент веб-браузера.

ВВОДНЫЕ ЗАМЕЧАНИЯ

СОБД мы рассматриваем как набор следующих компонентов [8]: HSM (HSM Library) – библиотека динамических моделей, содержащая набор иерархических ситуационных моделей HSM (Hierarchical Situational Models) в виде иерархии графов переходов с конечным числом состояний (Finite State Model); CSM (Current State Memory) – память текущего состояния, в которой хранятся сведения о текущих состояниях динамических моделей; ADM (Associated Document Memory) – память ассоциированных документов – хранилище XML-данных, соотнесенных (ассоциированных) с различными состояниями динамической модели; AFL (Associated Functions Library) – библиотека ассоциированных функций, хранящая функции обработки данных, соотнесенные с состояниями динамической модели; HSMI (HSM Interpreter) – интерпретатор динамической модели, который в ответ на внешний запрос Q формирует ответ R путем обработки некоторой динамической модели HSM из HSM на основе отслеживания ее текущих состояний, сохраняемых в CSM, обработки ассоциированных данных из ADM и выполнения ассоциированных функций из AFL.

Если СОБД используется на веб-сервере как основа для интернет-приложения, то в качестве входного запроса Q выступает набор параметров, получаемый вместе с URL (например, параметры формы в режиме POST), а в качестве результата R – ответный HTML-код, отправляемый клиенту. Параметры запроса задают обрабатываемую динамическую модель и необходимость смены ее текущих состояний [5, 6].

OLAP-клиент мы рассматриваем как программное средство, позволяющее отображать загруженные в него данные в виде сводной таблицы. Сводная таблица (Pivot Table) [9] – это программное средство визуализации исходных табличных данных в рамках концепции OLAP, позволяющее пользователю-аналитику в интерактивном режиме сортировать, укрупнять, детализировать данные, получать агрегированные (сводные) показатели: количества, суммы, средние значения и т. п.

Обычно OLAP-клиенты поддерживают загрузку анализируемых данных 1) из OLAP-серверов по XMLA-стандарту, 2) из реляцион-

ных баз данных по SQL-интерфейсу и 3) из других видов хранилищ в формате CSV.

Мы ориентируемся на последний случай – формат CSV (Comma-Separated Values – «разделенные запятыми значения»). Как правило, при использовании этого формата анализируемые данные загружаются в клиент в виде неагрегированных фактов, агрегация выполняется средствами самого OLAP-клиента.

Для практической реализации в данной работе используется свободно распространяемая версия OLAP-клиента FlexMonster Pivot Table & Charts Component [10].

Загрузка OLAP-клиента из СОБД. Таким образом, предполагается, что в определенных ситуациях, соответствующих заданным текущим состояниям динамической модели, СОБД должна на основе данных из хранилища документов ADM сформировать данные, предназначенные для анализа, в CSV-формате и загрузить их в OLAP-клиент (рис. 1).

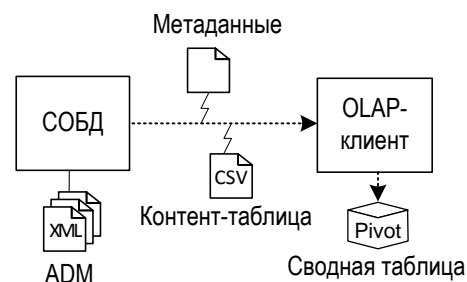


Рис. 1. Взаимодействие СОБД и OLAP-клиента с помощью контент-таблицы

Исходные данные для анализа, загружаемые в OLAP-клиент для формирования сводной таблицы, будем называть *контент-таблицей*. Помимо контент-таблицы, содержащей собственно анализируемые данные, в OLAP-клиент могут передаваться *метаданные*, задающие структуру сводной таблицы, отображаемой пользователю.

Таким образом, задача включает два этапа:

- определение структуры контент-таблицы – какие в ней должны быть столбцы;
- формирование ее содержимого – какие в ней должны быть строки.

КОНЦЕПЦИЯ ФОРМИРОВАНИЯ КОНТЕНТ-ТАБЛИЦЫ

Структура контент-таблицы

CSV-формат предписывает плоскую структуру контент-таблицы: первая строка задает имена столбцов, последующие строки задают

построчные кортежи значений. Следовательно, исходный гиперкуб данных, предоставляемый пользователю для анализа в виде сводной таблицы, должен быть преобразован в плоскую форму.

Если модель гиперкуба уже построена (например, в соответствии с подходом, изложенным в работе [2]), то задать столбцы контент-таблицы не представляет труда. Нужно выписать через запятую заголовки интересующих мер и измерений, помечая их соответствующим образом (в FlexMonster, например, заголовки мер помечаются префиксом «-», а измерений – «+»). Мы используем термин «заголовок», а не «имена», имея в виду, что имя – это характеристика объекта, предназначенная для разработчика, а заголовок – внешняя характеристика, отображаемая пользователю.

Модель хранилища документов ADM

Исходные данные, из которых должно формироваться содержимое контент-таблицы в рассматриваемой задаче, могут находиться в разнотипных XML-документах, размещенных в хранилище ADM. Процесс формирования, таким образом, потребует, в общем случае, извлечения данных из множества документов.

На рис. 2 приведен пример модели ADM, иллюстрирующий возможную взаимосвязь типов документов, служащих источниками исходных данных контент-таблицы.

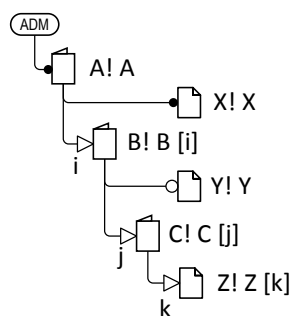


Рис. 2. Пример модели хранилища документов ADM

Нотация. В модели ADM папка обозначается символом \square , а документ – символом \square . Каждый объект – папка или документ – относится к некоторому типу и имеет имя. Имя типа объекта записывается перед именем объекта в виде префикса, оканчивающегося символом «!». Модель ADM – это модель *множеств* в том смысле, что она задает множества однотипных папок и документов. Символы \bullet , \circ , \triangleright , прикреп-

ленные слева к символам папок или документов, обозначают множества папок или документов. При этом темный кружок означает *обязательную однозначность*, т. е. что множество непустое и содержит ровно один элемент, светлый – *необязательную однозначность*, т. е. что множество может быть пустым или содержать один элемент, а светлый треугольник – *необязательную многозначность*, т. е. что множество может быть пустым или содержать несколько однотипных элементов.

В случае многозначности возникает необходимость идентификации отдельных элементов множества – папок или документов. Для этого символ треугольника помечается символом индекса, который также помещается в квадратных скобках после имени папки или документа.

По умолчанию подразумевается обязательная однозначность, поэтому далее на диаграмме модели ADM будем опускать темный кружок. Кроме того, будем опускать префикс типа в именах документов и папок, подразумевая, что папки или документы всегда относятся к определенному типу.

В рассматриваемой модели (см. рис. 2) все документы размещены в корневой папке с именем A, относящейся к типу A!. Корневая папка A содержит один обязательный документ X типа X! (формат XML по умолчанию) и одно необязательное множество B однотипных дочерних папок типа B! с именами B [i]. Каждая папка B [i], в свою очередь, может содержать один документ Y типа Y! и множество папок C [j]. В свою очередь каждая папка C [j] может содержать множество документов Z [k] типа Z!.

Таким образом, ADM имеет расщепленную структуру в том смысле, что однотипные данные хранятся в нескольких однотипных документах, размещенных в разных папках. В связи с этим возникает задача сбора данных, необходимых для формирования строк контент-таблицы, из расщепленных множеств однотипных документов.

Отображение контент-таблицы на хранилище документов

Задача извлечения данных из ADM при формировании строк контент-таблицы иллюстрируется на рис. 3. В этом примере каждая строка контент-таблицы формируется на основании данных, извлекаемых из трех документов: X, Y и Z. Из документов необходимо извлечь

значения определенных элементов данных (текстовые значения определенных XML-атрибутов или XML-элементов).

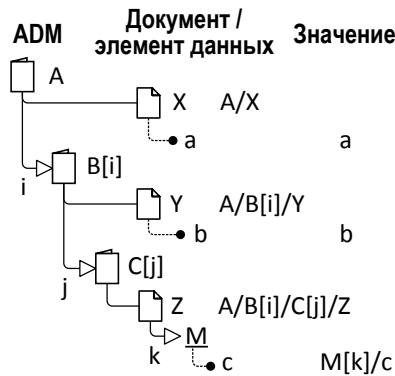


Рис. 3. Варианты извлечения данных из ADM для формирования строк контент-таблицы

1) Документ X размещен в корневой папке A и представлен в одном экземпляре. Путь к этому документу A/X. Из этого документа извлекается элемент данных «a», имеющий единственное значение в документе.

2) Документы Y размещены в папках множества B, вложенного в папку A. Путь к этим документам A/B [i]/Y, где i – индекс, пробегающий по всем папкам множества B. Таким образом, на один документ X приходится, в общем случае, несколько документов Y. Из документа Y извлекается элемент данных «b», имеющий единственное значение в документе. Таким образом, на каждое значение элемента данных «a» приходится, вообще говоря, несколько значений элемента данных «b».

3) Документы Z размещены в папках множества C, вложенных в папки множества B. Путь к этим документам A/B [i]/C [j]/Z, где j – индекс, пробегающий по всем папкам множества C, соответствующим фиксированному значению индекса i. Таким образом, на каждый документ Y приходится, в общем случае, несколько документов Z. Особенность этого случая состоит в том, что из документа Z извлекается элемент данных «c», имеющий, в общем случае, несколько значений в документе (на рис. 3 он представлен в виде атрибута «c» многозначного XML-элемента M). Таким образом, на каждое значение элемента данных «b» приходится, вообще говоря, несколько документов Z, каждый из которых может содержать несколько значений M [k]/c, где индекс k пробегает по всем экземплярам элемента M экземпляра документа Z.

Рассмотренная ситуация достаточно типична для задачи формирования контент-таблицы. Элементы данных «с» самого нижнего уровня соответствуют показателям фактов (мерам) OLAP-куба, элементы данных «b» – координатам измерений, а элементы данных «a» – укрупненным координатам измерений. Тем самым отражается то обстоятельство, что множество различных фактов может соответствовать одним и тем же координатам измерений.

Концептуальная схема формирования контент-таблицы

Рассмотрим теперь задачу формирования контент-таблицы в алгоритмическом плане: какие действия необходимо выполнить, чтобы сформировать множество строк контент-таблицы. Поскольку требуется извлекать данные из множества документов, необходимо организовать циклы перебора требуемых экземпляров в множествах однотипных папок и документов. При этом иерархическая модель ADM подсказывает, как организовать иерархию вложенных циклов перебора элементов из множества папок и документов.

При рассмотрении модели ADM в плане формирования контент-таблицы подразумевается стандартная процедура рекурсивного обхода дерева: обход ADM начинается с корневой папки; для каждой обрабатываемой папки последовательно обрабатываются вложенные в нее документы и другие папки в порядке их следования.

Для придания динамики статическая модель ADM дополняется элементами-действиями трех видов: цикла, сохранения промежуточных значений и вывода результата.

Элемент цикла (символ $\overline{\text{for}}$) обеспечивает последовательную обработку отдельных экземпляров – элементов множества папок или документов. Для элемента цикла задается условие отбора объектов обработки (в виде метки возле символа), индекс цикла (в разрыве соединительной линии, прикрепляющей дочернюю папку или документ). Действие элемента заключается в последовательном запуске обработки экземпляра дочерней папки или документа для всех значений индекса цикла.

Элемент сохранения промежуточных значений – это конструкция вида $x \rightarrow \$y$, где x – элемент данных, извлекаемый из документа, \$y – временная переменная, в которой сохраняется значение извлеченного элемента данных.

Элемент вывода результата (символ $\langle \text{ech} \rangle$) обеспечивает вывод сформированной строки контент-таблицы в выходной поток. Сформированная строка задается в виде кортежа временных переменных: $\langle \$x, \$y, \dots, \$z \rangle$.

На рис. 4 приведена концептуальная схема формирования строк контент-таблицы с использованием элементов-действий для рассмотренного выше примера ADM (см. рис. 3).

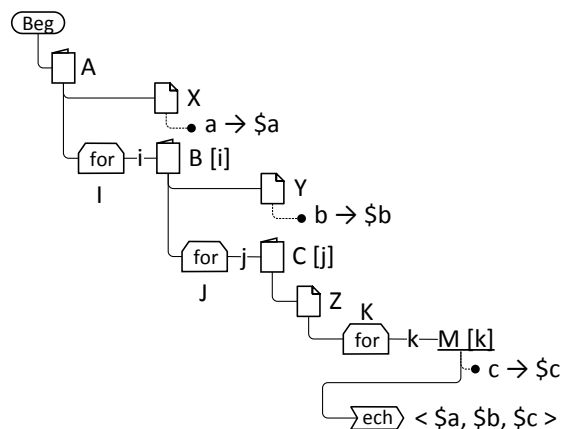


Рис. 4. Схема формирования строк контент-таблицы

В отличие от статической модели ADM схема на рис. 4 отражает динамику обработки документов ADM в процессе формирования строк контент-таблицы.

Обработка начинается с корневой папки А. Первым в папке А обрабатывается документ X, из которого извлекается элемент данных «а» и сохраняется во временной переменной \$а. После этого обрабатывается элемент цикла «I». Условие отбора экземпляров «I» по умолчанию предполагает обработку всех экземпляров дочернего множества объектов – папок В. Элемент цикла последовательно, одну за другой запускает на обработку папки В [i].

При обработке папки В [i] сначала обрабатывается вложенный документ Y, из которого извлекается элемент данных b и сохраняется во временной переменной \$b. Далее обрабатывается элемент цикла J, в результате чего последовательно запускается обработка папок C [j].

При обработке папки C [j] обрабатывается вложенный документ Z, для которого в свою очередь обрабатывается элемент цикла K. Результатом этого является последовательный запуск обработки экземпляров многозначного XML-элемента M [k].

При обработке экземпляра M [k] сначала извлекается XML-атрибут «с» и сохраняется во временной переменной \$c. Далее обрабатывается элемент вывода, в результате чего в выходной поток выводится сформированная строка контент-таблицы $\langle \$a, \$b, \$c \rangle$.

Таким образом, значения \$a являются одинаковыми для каждой строки контент-таблицы, а значения \$b – одинаковыми для групп строк, сформированных в пределах каждой папки В. На один документ Z приходится несколько сформированных строк по числу экземпляров XML-элемента M.

ФОРМИРОВАНИЕ И ЗАГРУЗКА КОНТЕНТ-ТАБЛИЦЫ СРЕДСТВАМИ ДИНАМИЧЕСКОЙ МОДЕЛИ

Возникает вопрос, как перевести рассмотренную выше концептуальную модель формирования контент-таблицы (построенную без учета среды реализации) в логическую модель, которая может быть реализована в рамках возможностей веб-приложения, функционирующего на базе СОБД. Иными словами, как обеспечить формирование контент-таблицы средствами динамической модели СОБД.

Возможности СОБД по обработке XML-документов

Динамическая модель HSM в составе СОБД предоставляет возможность обработки XML-документов с помощью так называемых динамических DOM-объектов [8, 11, 12].

Для порождения DOM-объектов в соответствующих состояниях **sta** дерева HSM в качестве дочерних размещаются dom-элементы **dom**, которые в свою очередь в качестве дочерних могут содержать элементы – источники **src** и приемники **rcv** данных. Элементы-источники специфицируют загрузку XML-документов из ADM в DOM-объект, а элементы-приемники – выгрузку XML-содержимого DOM-объекта (возможно, подвергнутого XSL-трансформации) в выходной поток, в ADM или в другой DOM-объект. Элементы-источники могут вкладываться друг в друга, что позволяет формировать в DOM-объекте XML-содержимое на основе нескольких исходных документов из ADM.

Арсенал средств динамических DOM-объектов создавался с ориентацией на задачи OLTP, и он недостаточен для решения рассматриваемых задач, ориентированных на OLAP.

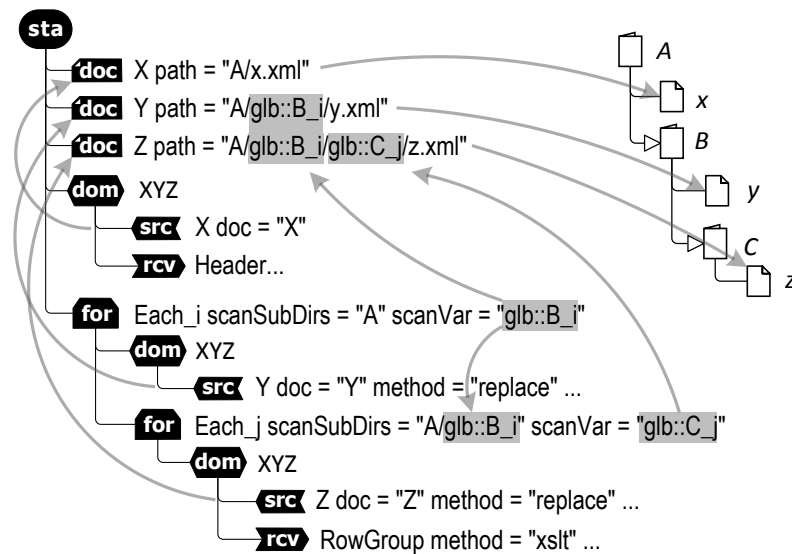


Рис. 5. Пример динамической модели HSM формирования контент-таблицы с использованием элементов цикла и селекции документов

В частности, возникают затруднения в случае необходимости циклической обработки множеств однотипных документов, размещенных в различных папках. Необходимо дополнить инструментарий HSM соответствующими возможностями.

Средства циклической обработки в динамической модели

Для упрощения циклической обработки множеств однотипных документов, размещенных в различных папках, в HSM введено два дополнения:

- элемент цикла **for**, или for-элемент, обеспечивающий многократную обработку dom-элементов;
- элемент селекции документа **doc**, или doc-элемент, задающий путь доступа к документам в ADM.

Элемент цикла располагается в HSM как дочерний элемент состояния и в свою очередь в качестве дочерних элементов может содержать dom-элементы и другие элементы цикла. Атрибут scanSubDirs этого элемента задает родительскую папку, дочерние папки которой подвергаются циклической обработке (если надо сканировать документы, то родительская папка задается атрибутом scanSubDocs). Спецификация пути в этом атрибуте может содержать ссылки на переменные охватывающих циклов и на элементы селекции. Атрибут scanVar задает переменную цикла, в которой хранится имя текущей

папки (или текущего документа), обрабатываемой на данном этапе выполнения цикла.

Элемент селекции располагается в HSM как дочерний элемент состояния. Атрибут path этого элемента задает путь к папке или документу в ADM. Важной особенностью этого атрибута является то, что спецификация пути может содержать ссылки на переменные циклов и на другие элементы селекции. Это обеспечивает гибкий доступ к папкам и документам в ходе выполнения цикла.

На рис. 5 иллюстрируется использование элементов цикла и селекции в динамической модели, предназначенной для формирования строк контент-таблицы, соответствующей рассмотренному выше примеру (см. рис. 4).

Единственное состояние динамической модели содержит три doc-элемента:

- элемент doc:X указывает на документ X в папке A;
- элемент doc:Y указывает на один из документов Y в папке B, какой именно – зависит от значения переменной цикла glb::B_i;
- элемент doc:Z указывает на один из документов Z в папке C, какой именно – зависит от значения переменных циклов glb::B_i и glb::C_j.

Далее следует элемент dom:XYZ, который порождает DOM-объект XYZ и с помощью элемента-источника src:X загружает в него документ X. Здесь же с помощью элемента-приемника rcv:Header в выходной поток выводится

строка заголовка контент-таблицы, содержащая CSV-список имен столбцов.

Следующий элемент цикла `for:Each_i` выполняет сканирование всех папок, размещенных в родительской папке А (т. е. папок В), имя текущей сканируемой папки сохраняется в переменной `glb::B_i`. Вложенный элемент `dom:XYZ` обеспечивает подзагрузку в ранее созданный DOM-объект XYZ документа Y, находящегося в сканируемой папке В. При этом новый экземпляр документа Y заменяет предыдущий (атрибут `method = "replace"`).

Вложенный элемент цикла `for:Each_j` выполняет сканирование всех папок, размещенных в сканируемой папке В (т. е. папок С), имя текущей сканируемой папки сохраняется в переменной `glb::C_j`. Вложенный элемент `dom:XYZ` обеспечивает подзагрузку в ранее созданный DOM-объект XYZ документа Z, находящегося в сканируемой папке С. Новый экземпляр документа Z заменяет предыдущий. Таким образом, к этому моменту DOM-объект XYZ содержит документы X, Y, Z, соответствующие сканируемым папкам.

Элемент-приемник `csv:RowGroup` обеспечивает формирование и вывод в выходной поток группы строк контент-таблицы, соответствующих текущему документу Z, путем XLS-трансформации документов X, Y, Z, загруженных к этому моменту в DOM-объект XYZ.

Таким образом, предложенные новые элементы HSM обеспечивают решение задачи формирования контент-таблицы.

Отметим, что в рассмотренном примере текущие документы, необходимые для формирования минимальной группы строк контент-таблицы, загружаются в общий DOM-объект для выполнения XSL-трансформации. Такой подход диктуется тем обстоятельством, что здесь на основе одного экземпляра документа Z формируется несколько строк контент-таблицы. В тех случаях, когда любая комбинация участвующих документов порождает только одну строку контент-таблицы, задачу можно решить, загружая в DOM-объекты по одному документу.

Такой случай иллюстрируется на рис. 6 в предположении, что один экземпляр документа Z порождает данные для одной строки контент-таблицы. Здесь документы X, Y, Z загружаются в отдельные DOM-объекты – X, Y, Z соответственно. Строки контент-таблицы формируются из трех последовательных фрагментов, получаемых в результате XSL-трансформации содержимого этих DOM-объектов.

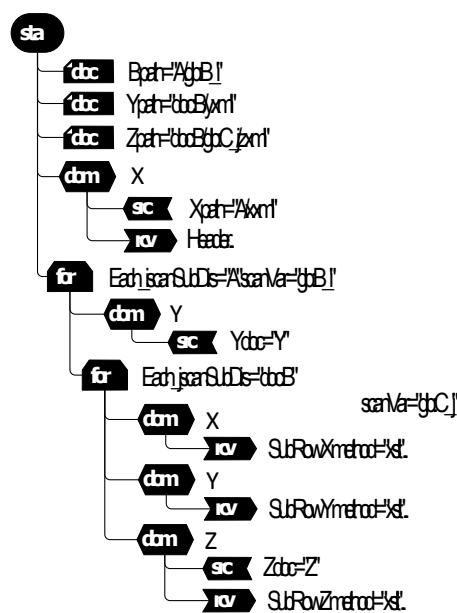


Рис. 6. Пример формирования по частям строк контент-таблицы

Другая особенность этого примера заключается в использовании ссылок на элементы селекции в других элементах селекции и элементах цикла. Так, элемент селекции `doc:V` задает путь к экземпляру папки В, сканируемой элементом цикла `for:Each_i`. Ссылка на этот элемент используется в элементах селекции `doc:Y` и `doc:Z` при задании пути к документам Y и Z, лежащим внутри папки В, а также в элементе цикла `for:Each_j` при задании родительской папки.

Реализация. Для реализации рассмотренных выше возможностей программный код интерпретатора динамических моделей HSMI был дополнен соответствующими модулями: модулем обработки элемента цикла и модулем обработки элемента селекции. Модули запрограммированы на языке PHP.

Организация взаимодействия СОБД и OLAP-клиента

Возникает вопрос о том, как организовать работу OLAP-клиента под управлением СОБД в составе веб-приложения, когда необходимо запускать клиент в браузере пользователя и загружать в него сформированную в СОБД контент-таблицу. При этом следует учитывать двухэтапный характер запуска OLAP-клиента, при котором:

1) веб-сервер отправляет в веб-браузер программный код и начальные данные, необходи-

мые для инициализации OLAP-клиента на стороне пользователя;

2) OLAP-клиент, выполнив инициализацию, с помощью запроса HTTP-request обращается к веб-сервису, в качестве ответа HTTP-response получает контент-таблицу, которую и отображает пользователю в виде сводной таблицы.

Таким образом, СОВД должна, во-первых, обеспечить в определенных состояниях запуск OLAP-клиента в браузере, во-вторых, предоставить OLAP-клиенту веб-сервис формирования контент-таблицы. Для этого в СОВД требуются две динамические модели: исходная, из которой запускается клиент, и динамическая модель сервиса контент-таблицы, которая формирует контент-таблицу. Общая схема взаимодействия СОВД и OLAP-клиента, ориентированная на использование клиента FlexMonster Pivot Table & Charts Component, приведена на рис. 7.

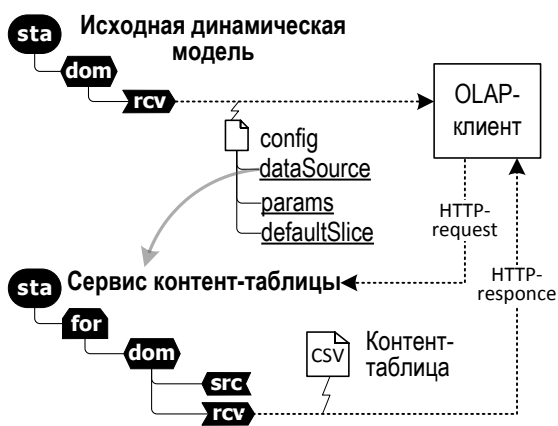


Рис. 7. Общая схема взаимодействия СОВД и OLAP-клиента

Исходная динамическая модель в определенном состоянии формирует стандартный набор данных, необходимых для запуска OLAP-клиента, и отправляет их через выходной поток в браузер пользователя. В том числе при этом клиенту передается файл конфигурации, содержащий, помимо прочего, адрес сервиса контент-таблицы.

Динамическая модель сервиса контент-таблицы в своем состоянии, соответствующем текущему состоянию исходной динамической модели, выполняет формирование и вывод в выходной поток строк контент-таблицы в соответствии с обсуждавшейся выше концепцией (см. рис. 5 и 6).

Синхронизация состояний динамических моделей. В реальных условиях возникает необ-

ходимость использовать OLAP-клиент в различных (не в единственном) состояниях исходной динамической модели, формируя для каждого случая свою контент-таблицу. В таком случае требуется построение нескольких веб-сервисов контент-таблиц, что неудобно.

Чтобы обойтись единственным сервисом, обслуживающим формирование контент-таблиц для всех вызовов OLAP-клиента, необходимо синхронизировать текущее состояние динамической модели сервиса с текущим состоянием исходной динамической модели. Чтобы достичь это, предлагается использовать следующий прием:

1) Структуру состояний динамической модели сервиса сделать соответствующей структуре состояний исходной динамической модели. Соответствие понимается в том смысле, что если, к примеру, в исходной HSM состояние $sta:Q$, запускающее OLAP-клиент, принадлежит субмодели $sub:M$ родительского состояния $sta:P$, то и в HSM сервиса контент-таблица должна формироваться в состоянии $sta:Q$, принадлежащем субмодели $sub:M$ родительского состояния $sta:P$.

2) При обращении к сервису контент-таблицы интерпретатор HSMI должен использовать память текущего состояния CSM исходной динамической модели, а не создавать свою независимую CSM, как в обычном случае.

Тогда при обращении к сервису контент-таблицы будет обработано то состояние, которое соответствует текущему состоянию исходной модели, запустившему OLAP-клиент.

ПРАКТИЧЕСКИЙ ПРИМЕР: ИНФОРМАЦИОННАЯ СИСТЕМА ДИССЕРТАЦИОННЫХ СОВЕТОВ

В качестве содержательного примера, иллюстрирующего обсуждавшиеся выше концепции, рассмотрим реализацию аналитической функциональности в информационной системе, обслуживающей деятельность диссертационных советов вуза. Информационная система представляет собой веб-приложение, в серверной базе данных которого хранятся сведения о диссертационных советах организации, о диссертантах и другие сведения, относящиеся к диссертационному процессу. Веб-приложение функционирует на основе СОВД, обеспечивающей как хранение данных в виде XML-документов, так и взаимодействие с пользователями через веб-браузеры. Изначально система была ориентирована на OLTP-задачи, связанные

с вводом, хранением и выдачей сведений об отдельных диссертантах, их диссертациях и защитах [5–7]. Однако управление как отдельным диссоветом, так и всей их совокупностью требует разноаспектного анализа ситуаций по совокупности рассматриваемых диссертаций на протяжении различных периодов времени. Требуется предусмотреть функции аналитики в общей функциональности информационной системы.

В работе [2] авторы рассматривали эту задачу на концептуальном уровне – в рамках многомерной модели было построено множество потенциальных гиперкубов, которые в принципе можно сформировать из данных, хранимых в базе данных системы. В этой работе требуемые гиперкубы реализуются на базе СОВД и OLAP-клиента FlexMonster Pivot Table & Charts Component.

Формируемые OLAP-отчеты. В рамках исследовательского прототипа системы реализованы 4 OLAP-отчета:

1) Диаграммы – представляет в виде сводных диаграмм сведения об активности диссертантов (стадии, этапы, показатели диссертаций и диссертационного процесса по различным диссоветам, специальностям и т. д.).

2) Активность диссертантов – представляет в виде сводной таблицы сведения об активности диссертантов (аналогично диаграммам, но в виде сводных таблиц).

3) Активность участников – представляет в виде сводной таблицы сведения об активности лиц-участников диссертационного процесса (в качестве членов диссоветов, научных руководителей, оппонентов и т. п.).

4) Активность организаций – представляет в виде сводной таблицы сведения об активности организаций в диссертационном процессе (ведущие организации, отзывы, сотрудники-участники и т. п.).

Структура контент-таблиц. В соответствии с потребностями анализа, с учетом многомерных моделей гиперкубов, полученных в работе [2], составлены заголовки столбцов контент-таблиц для OLAP-отчетов. В качестве примера в табл. 1 приведены заголовки столбцов для отчета «Диаграммы» (измерения отмечены знаком «+», а меры – знаком «-»).

Структура хранилища документов. Хранилище XML-документов ADM, на основе которых формируются контент-таблицы для отчетов

тов OLAP, иллюстрируется в виде модели, представленной на рис. 8.

Таблица 1
Заголовки столбцов контент-таблицы для отчета «Диаграммы»

Тип	Заголовок	Содержание
+	Диссовет	Шифр диссовета
+	Степень	Искомая степень
+	Стадия	Стадия процесса защиты
+	Состояние	Состояние процесса защиты
+	Специальность	Шифр специальности (осн.)
+	Диссертант	Идентификатор диссертанта
+	Год	Год защиты
+	ДатПриема	Дата приема к защите
+	ДатЗащиты	Дата защиты
-	Любых	Кол-во защит
-	Своих	Кол-во защит своих дис-тов
-	Чужих	Кол-во сторонних защит
-	Успешных	Кол-во успешных защит
-	Неудачных	Кол-во неудачных защит
-	ВАК-запрос	Кол-во запросов из ВАК
-	ВАК-вызов	Кол-во вызовов в ВАК
-	ВАК-замеч	Кол-во замечаний ВАК
-	Прогулов	Кол-во отсутствовавших членов диссовета
-	Выступл	Кол-во выступавших
-	Вопросов	Кол-во задававших вопросы
-	Внедрен	Кол-во актов внедрения
-	ПублВАК	Кол-во публикаций из перечня ВАК

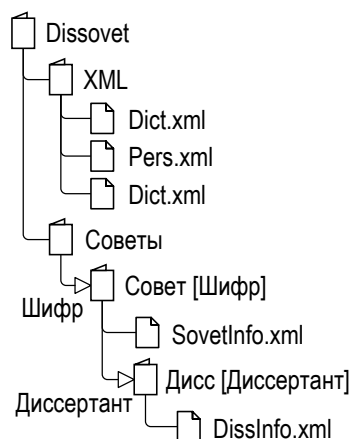


Рис. 8. Структура хранилища документов

Хранилище включает 4 типа папок и 5 типов XML-документов, размещенных в корневой папке Dissovet. Папка XML содержит справочники: словарь Dict.xml, глобальный справочник персон Pers.xml и глобальный справочник организаций Org.xml. Папка Советы содержит оперативные данные о диссертационных советах

и защищающихся в них диссертантах. Папки из множества папок Совет содержат сведения, относящиеся к отдельному совету (имя папки – шифр совета). Каждая такая папка включает документ SovetInfo.xml, содержащий сведения о совете, и множество Дисс, содержащее папки со сведениями о диссертантах, защищающихся в этом совете (имя папки – идентификатор диссертанта). Каждая такая папка включает документ DissInfo.xml с детальными сведениями о диссертанте, диссертации и ходе защиты.

Структура основных документов. Основные документы – источники данных для формирования контент-таблиц OLAP-отчетов – это SovetInfo.xml (сведения о совете), и DissInfo.xml (сведения о диссертанте).

Схема документа SovetInfo.xml (рис. 9) достаточно проста: сведения об утверждении и полномочиях совета, перечень специальностей, по которым совет присуждает степени и перечень членов совета.

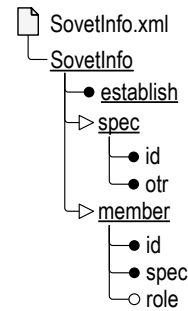


Рис. 9. Схема документа со сведениями о диссертационном совете

Схема документа DissInfo.xml (рис. 10) гораздо сложнее и включает персональные данные диссертанта, различные сведения о диссертации, о событиях диссертационного процесса, в том числе – процесса защиты диссертации, о результатах защиты и др. Здесь также присутствуют локальные базы персон и организаций, имеющие то или иное отношение к данному диссертанту.

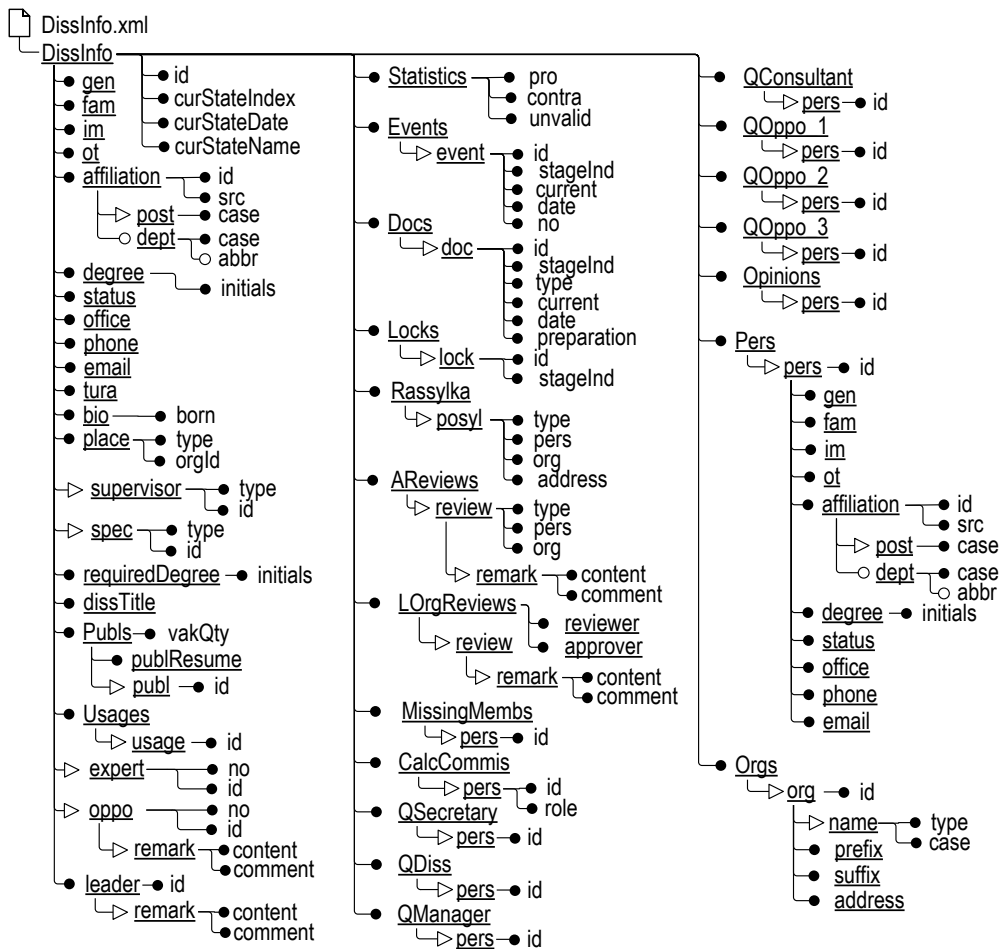


Рис. 10. Схема документа со сведениями о диссертанте

Исходная динамическая модель. В листинге 1 представлен код исходной динамической модели в виде ориентированной на функции аналитики субмодели `sub:Analysis`, являющейся частью общей динамической модели СОБД.

Модель содержит 5 состояний, первое из которых, `sta:Список_отчетов`, обеспечивает выбор OLAP-отчета, а остальные, `sta:Форма_1...`–`sta:Форма_4...`, обеспечивают запуск OLAP-клиента для вывода выбранного отчета (в листинге представлен код только двух таких состояний). Во всех состояниях предусмотрена

навигация для перехода в другие состояния (`jmp`-элементы); вывод информации в браузер пользователя осуществляется единообразно, путем XSL-трансформации (приемник `rcv:echo`) содержимого DOM-объекта, заданного соответствующим `dom`-элементом.

Динамическая модель сервиса контент-таблиц. В листинге 2 представлен код динамической модели, включающей 4 состояния, обеспечивающие сервис формирования контент-таблиц для четырех OLAP-отчетов (для компактности представлен код формирования двух отчетов).

Листинг 1. Исходная динамическая модель с состояниями функций аналитики

```
<sub:Analysis>
  <sta:Список_отчетов class = "PivotFuncs">
    <jmp:PivotFuncs targс = "Список_отчетов Форма_1_Диаграммы Форма_2_Диссертанты
      Форма_3_Участники Форма_4_Организации" mode = "regular">
      <sel:PivotFuncs optType = "LIST" curOptIndex = "0" activator = "onChange" place = "Admin"/>
    </jmp:PivotFuncs>
    <doc:PivotReport-Stylesheet type = "xml" path = "XSL/PivotReportsList-Gen.xml" />
    <dom:Info srcDoc = "PivotReport-Stylesheet">
      <rcv:echo pass="2" method="xslt" styleSheet="doc:PivotReport-Stylesheet"/>
    </dom:Info>
  </sta:Список_отчетов>

  <sta:Форма_1_Диаграммы class = "PivotFuncs">
    <jmp:PivotFuncs targс = "Список_отчетов Форма_1_Диаграммы Форма_2_Диссертанты
      Форма_3_Участники Форма_4_Организации" mode = "regular">
      <sel:PivotFuncs optType = "LIST" curOptIndex = "1" activator = "onChange" place = "Admin"/>
    </jmp:PivotFuncs>
    <doc:PivotReport-Stylesheet type = "xml" path = "XSL/PivotReport_1-Gen.xml" />
    <dom:SovetInfo>
      <rcv:echo pass="2" method="xslt" styleSheet="doc:PivotReport-Stylesheet"/>
    </dom:SovetInfo>
  </sta:Форма_1_Диаграммы>

  <sta:Форма_2_Диссертанты class = "PivotFuncs">
    . . . . .
  </sta:Форма_2_Диссертанты>

  <sta:Форма_3_Участники class = "PivotFuncs">
    . . . . .
  </sta:Форма_3_Участники>

  <sta:Форма_4_Организации class = "PivotFuncs">
    <jmp:PivotFuncs targс = "Список_отчетов Форма_1_Диаграммы Форма_2_Диссертанты
      Форма_3_Участники Форма_4_Организации" mode = "regular">
      <sel:PivotFuncs optType = "LIST" curOptIndex = "4" activator = "onChange" place = "Admin"/>
    </jmp:PivotFuncs>
    <doc:PivotReport-Stylesheet type = "xml" path = "XSL/PivotReport_4-Gen.xml" />
    <dom:SovetInfo>
      <rcv:echo pass="2" method="xslt" styleSheet="doc:PivotReport-Stylesheet"/>
    </dom:SovetInfo>
  </sta:Форма_4_Организации>
</sub:Analysis>
```

Листинг. 2. Динамическая модель сервиса контент-таблиц

```

<sub:Analysis>
  <sta:Форма_1_Диаграммы>
    <doc:Pers type = "xml" path = "XML/pers.xml" />
    <doc:PivotData-Stylesheet type = "xml" path = "XSL/PivotData_1.xsl" />
    <doc:SovetInfo type = "xml" path = "Советы/glb::CurSovet/SovetInfo.xml"/>
    <doc:DissInfo type = "xml" path = "Советы/glb::CurSovet/Дисс/glb::CurDis/DissInfo.xml"/>
    <dom:Info srcDoc = "PivotData-Stylesheet">
      <rcv:echo pass = "1" method="text"
        value="+Диссовет,+Степень,+Стадия,+Состояние,+Специальность,+Диссертант,+Год,D+ДатПриема,
          D+ДатЗащиты,-Любых,-Своих,-Чужих,-Успешных,-Неудачных,-ВАК-запрос,-ВАК-вызов,-ВАК-замеч,
          -Прогулов,-Выступл,-Вопросов,-Внедрен,-ПублВАК"/>
      <rcv:echo pass = "1" method="text" value="#13;"/>
    </dom:Info>
    <for:EachSov scanSubDirs = "Советы" scanVar = "glb::CurSovet">
      <for:EachDis scanSubDirs = "Советы/glb::CurSovet/Дисс" scanVar = "glb::CurDis">
        <dom:DissInfo-Sovet srcDoc = "DissInfo">
          <src:SovetInfo doc="SovetInfo"/>
          <rcv:echo pass = "1" method="xslt" styleSheet="doc:PivotData-Stylesheet" parGlobRef="CurSovet"/>
        </dom:DissInfo-Sovet>
      </for:EachDis>
    </for:EachSov>
  </sta:Форма_1_Диаграммы>

  <sta:Форма_2_Диссертанты>
    .....
  </sta:Форма_2_Диссертанты>

  <sta:Форма_3_Участники>
    .....
  </sta:Форма_3_Участники>

  <sta:Форма_4_Организации>
    <doc:Pers type = "xml" path = "XML/pers.xml" />
    <doc:PivotData-Stylesheet type = "xml" path = "XSL/PivotData_4.xsl" />
    <doc:SovetInfo type = "xml" path = "Советы/glb::CurSovet/SovetInfo.xml"/>
    <doc:PivotData-Membs-Stylesheet type = "xml" path = "XSL/PivotData_4-Membs.xsl" />
    <doc:DissInfo type = "xml" path = "Советы/glb::CurSovet/Дисс/glb::CurDis/DissInfo.xml"/>
    <dom:Info srcDoc = "PivotData-Stylesheet">
      <rcv:echo pass = "1" method="text"
        value="-Кол-во,+Диссовет,+Организация,+Активн,+Диссертант,+Спец,+СтадияДис,+СтатусДис,+Степень,
          +Год,D+ДатПриема,D+ДатЗащиты"/>
      <rcv:echo pass = "1" method="text" value="#13;"/>
    </dom:Info>
    <for:EachSov scanSubDirs = "Советы" scanVar = "glb::CurSovet">
      <for:EachDis scanSubDirs = "Советы/glb::CurSovet/Дисс" scanVar = "glb::CurDis">
        <dom:DissInfo-Sovet srcDoc = "DissInfo">
          <src:SovetInfo doc="SovetInfo"/>
          <rcv:echo pass = "1" method="xslt" styleSheet="doc:PivotData-Stylesheet" parGlobRef="CurSovet"/>
        </dom:DissInfo-Sovet>
      </for:EachDis>
    </for:EachSov>
  </sta:Форма_4_Организации>
</sub:Analysis>

```

Структура динамической модели сервиса соответствует структуре исходной динамической модели: имена состояний, как и имя суб-модели, – те же, что и в исходной модели. Это позволяет с помощью общей динамической модели формировать различные контент-таблицы.

Контент-таблицы формируются на основе документов двух типов: *SovetInfo.xml* (сведения о совете) и *DissInfo.xml* (сведения о диссертанте). Для сканирования сведений обо всех диссертантах во всех советах предусмотрены два *for*-элемента: элемент *for:EachSov* сканирует советы, а вложенный в него элемент *for:EachDis* – диссертантов в совете.

Строки контент-таблицы последовательно формируются путем XSL-трансформации содержимого DOM-объекта, в который загружены по одному экземпляру документов *SovetInfo.xml* и *DissInfo.xml*. Для этого используются соответствующие таблицы стилей, составленные на языке XSLT. При этом для первых двух отчетов, отражающих активность диссертантов, каждому экземпляру документа *DissInfo.xml* соответствует одна строка контент-таблицы (посвященная одному диссертанту), а для остальных отчетов, отражающих активность лиц и организаций, – несколько строк. Последнее обусловлено тем обстоятельством, что с каждым диссертантом тем или иным образом связано несколько лиц и организаций (научный руководитель, оппонент, организация диссертанта, ведущая организация и т. п.).

Визуализация данных. На рис. 11–14 приведены скриншоты OLAP-отчета «Активность организаций», отображаемого в браузере пользователя.

На рис. 11 иллюстрируется исходная форма сводной таблицы, как она отображается сразу после запуска и загрузки контент-таблицы. В строках отображается иерархия измерений «Организация-Диссертант», в столбцах – показатели-меры, в области фильтра – поля-измерения «Стадия», «Степень» и «Год». Такая структура задана с помощью файла конфигурации *config.xml*, ссылка на который передается OLAP-клиенту при запуске из исходной модели.

На рис. 12 иллюстрируются возможности детализации данных (*Drill down*) в сводной таблице: для организации УрФУ раскрыты показатели ее активности по отношению к отдельным диссертантам.

На рис. 13 и 14 иллюстрируются возможности изменения структуры сводной таблицы пользователем. Окно формирования среза ги-

перкуба (рис. 14) позволяет задать поля, отображаемые в строках, столбцах или в области фильтра, а также другие особенности визуализации данных. На рис. 14 показан вид сводной таблицы после того, как пользователь поменял местами уровни иерархии в строках таблицы (теперь для каждого диссертанта возможна детализация показателей в плане задействованных организаций).

ЗАКЛЮЧЕНИЕ

Таким образом, в статье представлены результаты решения задачи реализации функций аналитики в веб-приложениях, основанных на применении ситуационно-ориентированных баз данных (СОБД).

Для решения задачи применен подход, состоящий в том, что из XML-документов серверного хранилища СОБД в определенных состояниях формируется «на лету» контент-таблица, которая отправляется в OLAP-клиент, загруженный в веб-браузер пользователя.

Предложенная концептуальная схема построения формирования контент-таблицы основана на вложенных циклах обработки множеств однотипных XML-документов из хранилища СОБД. Строки контент-таблицы формируются путем XSL-трансформации XML-документов хранилища, динамически загружаемых в DOM-объекты.

Введены дополнительные элементы динамической модели СОБД, позволяющие формировать строки контент-таблицы путем сканирования хранилища XML-документов: элемент цикла, обеспечивающий сканирование однотипных папок, и элемент селекции документа, обеспечивающий выбор экземпляров однотипных документов для загрузки в DOM-объекты в ходе сканирования.

Предложенные элементы динамической модели поддержаны программным обеспечением в виде модулей в составе интерпретатора динамической модели СОБД, ориентированным на контент-таблицы в формате CSV и OLAP-клиент *FlexMonster Pivot Table & Charts Component*.

Работоспособность предложенного подхода подтверждена путем реализации четырех типов OLAP-отчетов в рамках исследовательского прототипа распределенной информационной системы, предназначенной для управления комплексом диссертационных советов научной организации (вуза).

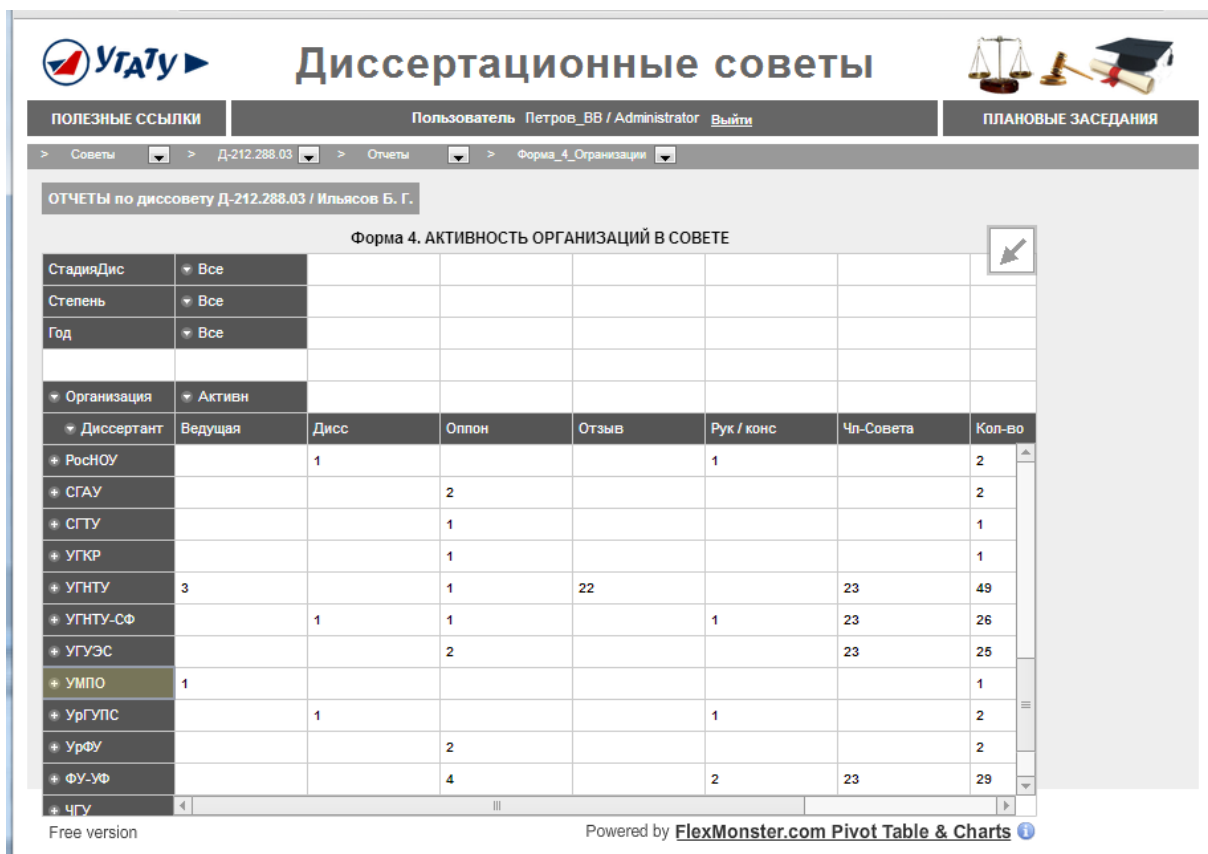


Рис. 11. Сводная таблица «Активность организаций» в окне веб-браузера

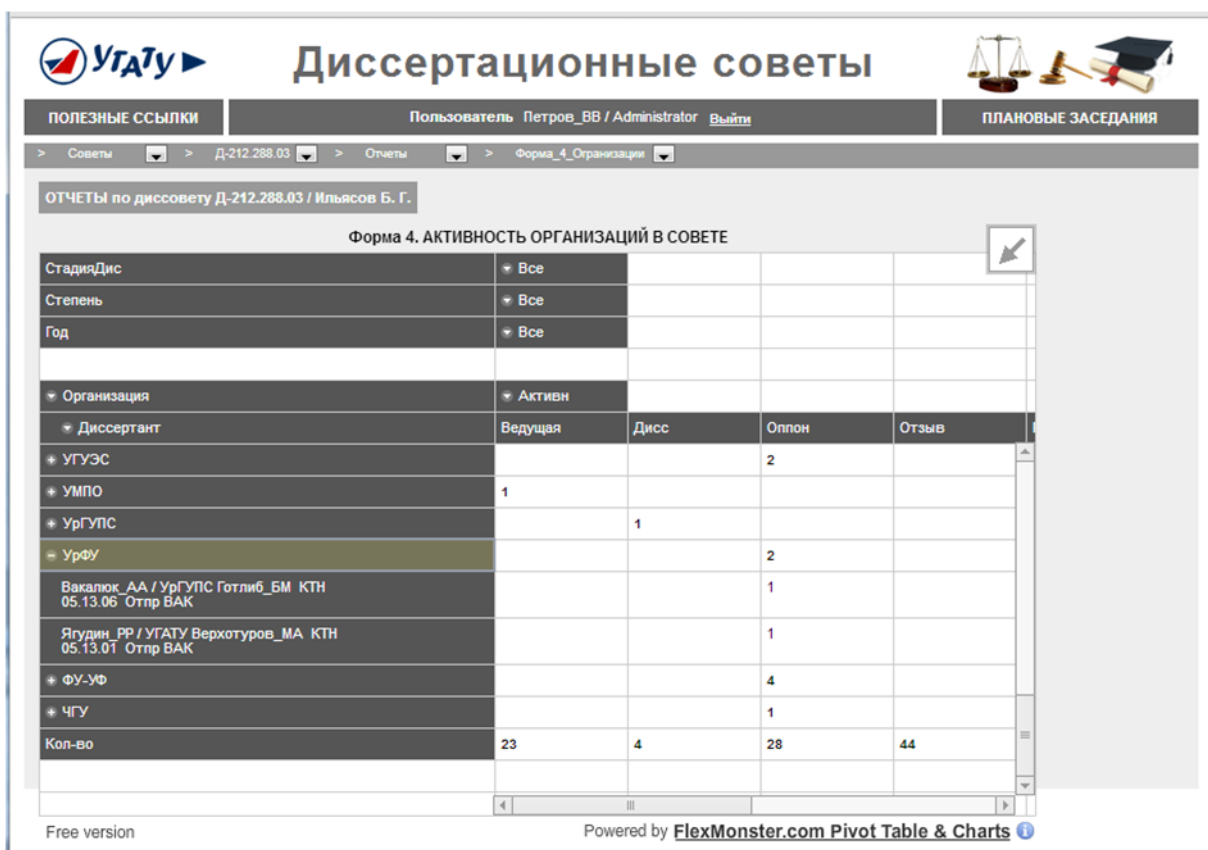


Рис. 12. Детализация данных в сводной таблице

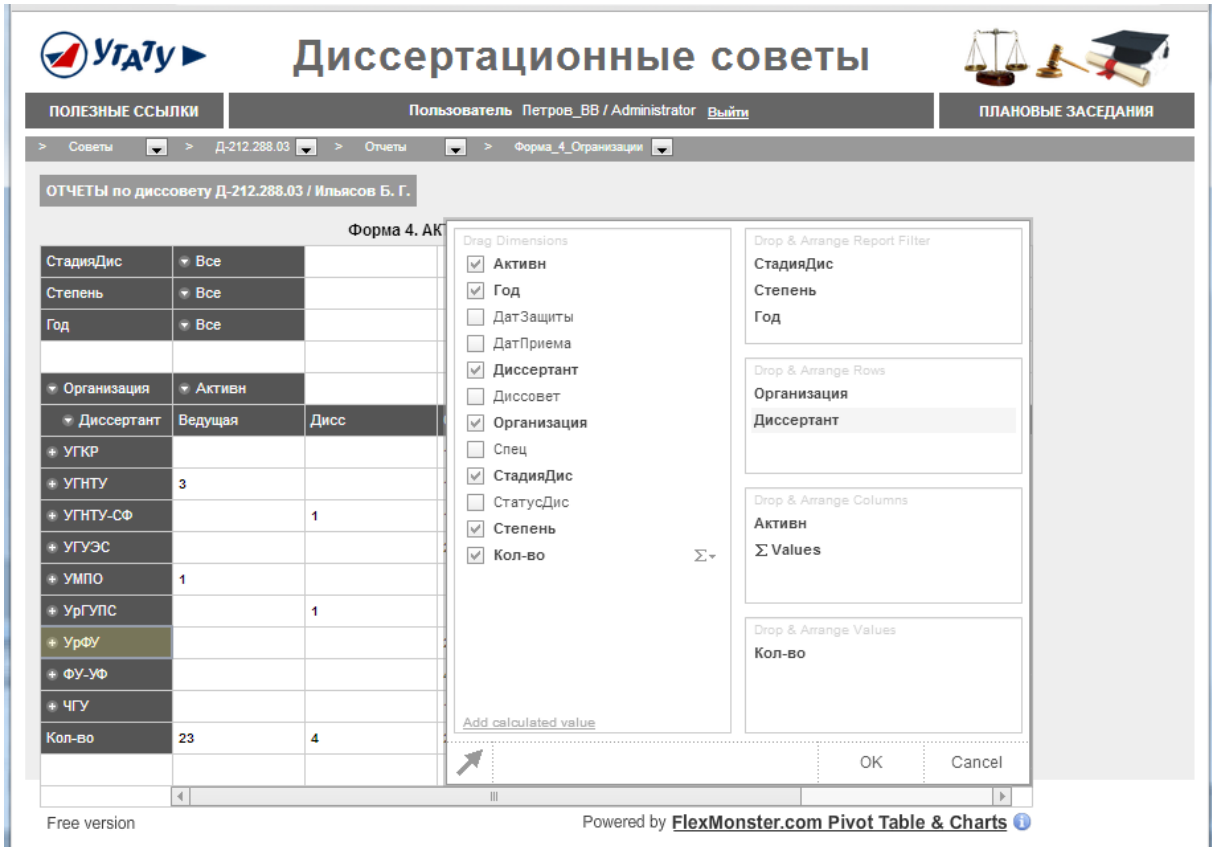


Рис. 13. Задание пользователем структуры сводной таблицы

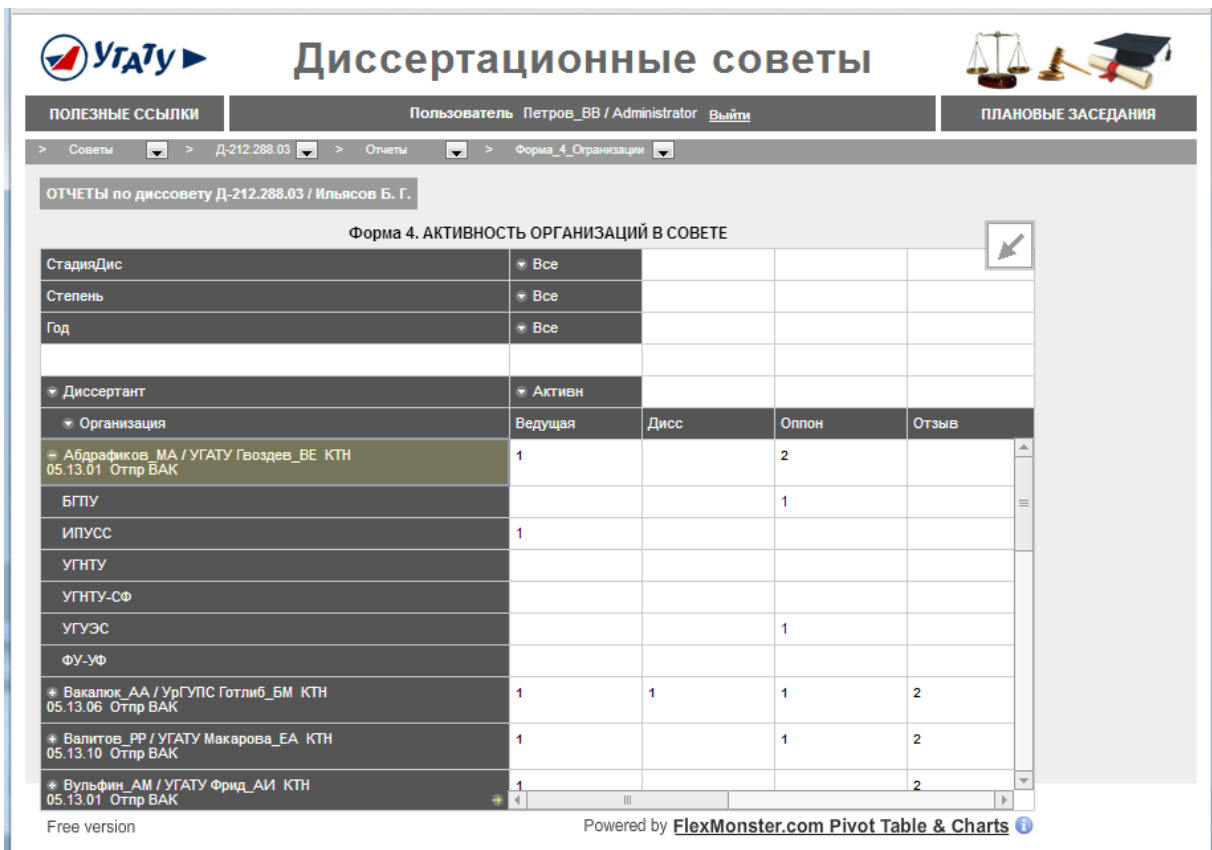


Рис. 14. Сводная таблица с измененной пользователем структурой

СПИСОК ЛИТЕРАТУРЫ

1. **Olap.ru.** Business Intelligence – Effective Data Mining & Analysis [Электронный ресурс]. URL: <http://olap.ru/> (дата обращения 07.08.2013).
2. **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.
3. **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.
4. **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.
5. **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.
6. **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.
7. **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.
8. **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.
9. **Pivot table** [Электронный ресурс] // Wikipedia: The Free Encyclopedia. URL: http://en.wikipedia.org/wiki/Pivot_table (дата обращения 01.08.2013).
10. **Flexmonster Pivot Table & Charts Component** [Электронный ресурс]. URL: <http://www.flexmonster.com/> (дата обращения 01.08.2013).
11. **Гусаренко А. С., Миронов В. В.** Динамические dom-объекты в ситуационно-ориентированных базах данных: лингвистическое и алгоритмическое обеспечение источников данных // Вестник УГАТУ. 2012. Т. 16, № 6 (51). С. 167–176.
12. **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.

ОБ АВТОРАХ

МАКАРОВА Екатерина Сергеевна, асп. каф. АСУ. Дипл. м-р техн. и технол. в обл. инф-ки и выч. техники (УГАТУ, 2010). Готовит дис. о технологиях Web-OLAP.

МИРОНОВ Валерий Викторович, проф. каф. АСУ. Дипл. радиоприемник (Воронежск. гос. ун-т, 1975). Д-р техн. наук по упр. в техн. системах (УГАТУ, 1995). Иссл. в обл. иерархич. моделей и ситуац. управления.

METADATA

Title: Analysis functions in web applications based on situation-oriented databases.

Authors: E. S. Makarova and V. V. Mironov.

Affiliation: Ufa State Aviation Technical University (UGATU), Russia.

Email: es_makarova@list.ru, mironov@list.ru.

Language: Russian.

Source: Vestnik UGATU (Scientific journal of Ufa State Aviation Technical University), vol. 17, no. 5 (58), pp. 150-165, 2013. ISSN 2225-2789 (Online), ISSN 1992-6502 (Print).

Abstract: Analysis functions organization in the web applications functioning on the basis of the server situation-oriented databases (SOBD) and OLAP-clients is considered. SOBD and OLAP-client interaction by formation of the intermediate content table is discussed. The scheme of formation of the content table of the XML documents storing in SOBD is under construction. The additional components of the SOBD dynamic model facilitating the organization of content table service are offered. The results are illustrated for FlexMonster Pivot Table & Charts Component OLAP client.

Key words: Web application; situation-oriented database; dynamic model; XML; XSLT; CSV; Web OLAP; OLAP client; FlexMonster.

References (English Transliteration):

1. (2013, Aug. 07). Olap.ru. Business Intelligence – Effective Data Mining & Analysis [Online]. Available: <http://olap.ru>
2. 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.
3. 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.
4. 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.
5. 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.
6. 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.
7. 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.
8. V. V. Mironov and A. S. Gusarenko, “Situation-oriented databases: concept of XML data management based of dynamic DOM objects,” (in Russian), *Vestnik UGATU*, vol. 16, no. 3 (48), pp. 159-172, 2012.
9. (2013, Aug. 07). “Pivot table,” *Wikipedia: The Free Encyclopedia* [Online]. Available: http://en.wikipedia.org/wiki/Pivot_table
10. (2013, Aug. 07). Flexmonster Pivot Table & Charts Component [Online]. URL: <http://www.flexmonster.com/>
11. A. S. Gusarenko and V. V. Mironov, “Dynamic DOM objects in situation-oriented databases: lingware and knoware of data sources,” (in Russian), *Vestnik UGATU*, vol. 16, no. 6 (51), pp. 176-167, 2012.
12. V. V. Kanashin and V. V. Mironov, “Dynamic DOM objects in situation-oriented databases: lingware and knoware of data sources,” (in Russian), *Vestnik UGATU*, vol. 17, no. 2 (55), pp. 138-149, 2013.

About authors:

MAKAROVA, Ekaterina Sergeyevna, Postgrad. (PhD) Student, Automated Systems Dept.. Master Tech. and Technol. (UGATU, 2010).

MIRONOV, Valeriy Viktorovich, Prof., Automated Systems Dept. Dipl. Radiophysicist (Voronezh State Univ., 1975). Dr. (Habil.) Tech. Sci. (UGATU, 1995).