ПРОГРАММНЫЕМЕХАНИЗМЫ УПРАВЛЕНИЯ ИНТЕГРИРОВАННЫМИ ИНФОРМАЦИОННЫМИ РЕСУРСАМИ

А.С. Шундеев
Научно-исследовательский институт механики МГУ имени М.В. Ломоносова
Россия, 119192, Москва, Мичуринский проспект, д.1
E-mail: alex.shundeev@gmail.com

И.С. Першин
Научно-исследовательский институт механики МГУ имени М.В. Ломоносова
Россия, 119192, Москва, Мичуринский проспект, д.1
E-mail: pershin.is@gmail.com

Основная область исследований, результаты которых представлены в докладе, относится к технологиям управления и интеграции данных. Обобщаются подходы к решению задачи интеграции данных, используемые при построении корпоративных информационных систем, для случая межкорпоративного электронного взаимодействия.

PROGRAM MECHANISMS FOR THE INTEGRATED INFORMATION RESOURCES MANAGEMENT / A.S. Shundeev, I.S. Pershin (Institute Of Mechanics, Lomonosov Moscow State University, 1 Michurinskiy pr., Moscow 119192, Russia). The main area of the researches, which results are presented in the report, concerns technologies of the data management and integration. Approaches to the decision of the task of integration of the data, used are generalized at creation of corporate information systems, for a case of intercorporate electronic interaction.

Введение

Ежегодно авторитетная консалтинговая компания Gartner публикует аналитический отчет [1], который содержит рейтинг производителей программных решений в области интеграции данных. Для каждой компании, попавшей в рейтинг, оценивается эффективность и полнота предлагаемых ею технических решений. Весь спектр практически значимых задач в области интеграции данных описывается в рамках семи сценариев, к которым относятся:

- создание федеративных представлений для данных из множества различных источников;
- сервисы данных в контексте сервисно-ориентированной архитектуры;
- унификация структурированных и неструктурированных данных

и ряд других.

В дальнейшем под решением задачи интеграции данных будем понимать создание целевого программного приложения на основе трех выделенных сценариев интеграции данных. При этом будем отдельно рассматривать случай создания корпоративного приложения (корпоративный случай) и случай автоматизации межкорпоративного взаимодействия (межкорпоративный случай).

В отчете компании Gartner отмечается, что подавляющее большинство производителей, попавших в рейтинг, ориентируются на рынок создания корпоративных приложений. В этой области существуют апробированные, проверенные временем подходы и технические решения. Настоящая работа посвящена обобщению выработанных для корпоративного случая подходов к решению задачи интеграции данных на случай межкорпоративного взаимодействия.

Виртуальные базы данных

Рассмотрим задачу интеграции данных для корпоративного случая. Постановка этой задачи состоит в следующем. Имеется набор баз данных (далее - БД), относящихся к одной или разным предметным областям. Подобные БД принято называть *локальными*. Требуется разработать программный механизм, позволяющий взаимодействовать с набором локальных БД как с единой БД. Данный механизм принято называть *виртуальной* БД. При этом, требуется соблюсти следующее ограничение. Нельзя создавать «физические» копии локальных БД. Для запроса к виртуальной БД в оперативном режиме формируется план его выполнения, включающий в себя подзапросы к локальным БД, операции по обработке промежуточных результатов и по формированию итогового результата. Подобный план выполняется средствами виртуальной БД.

Задача интеграции данных в приведенной постановке остается актуальной на протяжении уже нескольких десятков лет. Имеются частные решения, обладающими своими преимуществами и недостатками. Следует упомянуть работу [2], относящуюся к 80-м годам прошлого столетия, в которой предлагались подходы к созданию виртуальных БД над наборами разнородных локальных БД (иерархических, сетевых, реляционных). При этом предлагалось решение, базирующееся на создании общей модели данных высокого уровня и преобразовании произвольных моделей данных в общую модель.

В последнее время, в связи с ростом популярности технологий управления полуструктурированными данными появились базирующиеся на них подходы к решению задачи интеграции данных. Это прежде всего касается языка XML - стандартного формата для представления полуструктурированных данных, и языка запросов XQuery [3] - средства манипулирования XML- данными. Одна из возможных реализаций данного подхода представлена в работе [4]. В данной реализации виртуальная база данных имеет XML, XQuery интерфейс и позволяет интегрировать традиционные реляционные БД и XML, XQuery БД.

Архитектура виртуальной базы данных

Опишем типовой подход к проектированию архитектуры виртуальной БД. В архитектуре выделяют два типа структурных элементов: *обертка (wrapper)* и *посредник (mediator)*. Элементы первого типа отвечают за непосредственное взаимодействие с локальными БД, а элементы второго типа - за выполнение планов запросов виртуальной БД. В рамках сервисно-ориентированной архитектуры обертки и посредники реализуются в виде программных сервисов.

В случае оберток, каждой локальной БД может быть сопоставлен индивидуальный программный сервис, отвечающий за взаимодействие только с данной БД. Другое возможное решение базируется на использовании одного программного сервиса (ограниченного набора сервисов), отвечающего взаимодействие сразу со всеми локальными БД в составе виртуальной БД. В этом случае обертки выступают в виде своеобразных «логических» сущностей (параметров настроек доступа) в рамках программного сервиса.

Аналогично, медиаторы могут реализовываться в виде одного или набора программных сервисов. Последний вариант, как правило, используется для организации распределенного выполнения планов запросов виртуальной БД.

Доступ к локальным базам данных

В простейшем случае обертка представляет собой адаптер для взаимодействия с локальной БД. В более сложных случаях данный элемент может решать задачи по унификации интерфейса взаимодействия (протокол доступа, модель и средства манипулирования данными) с локальной БД. Приведем примеры.

Рассмотрим задачу унификации интерфейса доступа к реляционным БД на основе технологий XML, XQuery. Любая реляционная БД имеет стандартное представление в виде

XML документа [6]. К подобному XML документу может быть применен XQuery запрос. Таким образом, язык XQuery, изначально разработанный как средство манипулирования XML документами, может также рассматриваться как язык запросов к реляционным данным. Отметим, что формирование «физической» копии реляционной БД в виде XML документа для выполнения индивидуального XQuery запроса является не самым эффективным решением.

В работе [5] описывается алгоритм трансляции XQuery запросов в серию SQL запросов, которые выполняются штатными средствами реляционной БД. Полученные ответы на SQL запросы затем комбинируются и формируют итоговый ответ на исходный XQuery запрос. Данный подход применим к достаточно представительному подмножеству языка XQuery. В работе [4] язык XQuery ограничивается подмножеством «эквивалентным» языку SQL. При этом каждый XQuery запрос транслируется ровно в один SQL запрос.

Сопоставление схем данных

Используя механизм оберток, можно модели и средства манипулирования локальных БД привести к единообразному виду. В результате задача интеграции данных несколько упрощается. Можно предполагать, что виртуальная и локальная БД базируются на одной модели данных, а также используют общий язык манипулирования данными.

Каждая БД представляет собой абстрактную модель некоторой предметной области. Для описания таких предметных областей используются схемы данных. Будем называть схему виртуальной БД *глобальной схемой*, а схемы локальных БД - *локальными схемами*.

Для того, чтобы корректно использовать механизм виртуальной БД, необходимо формально определить процедуру формирования состояния виртуальной БД. Для этого необходимо решить задачи *сопоставления* локальных схем данных на глобальную (подход global-as-view [7]). Подобное сопоставление может осуществляться как в «ручном», так и в автоматизированном режиме [8].

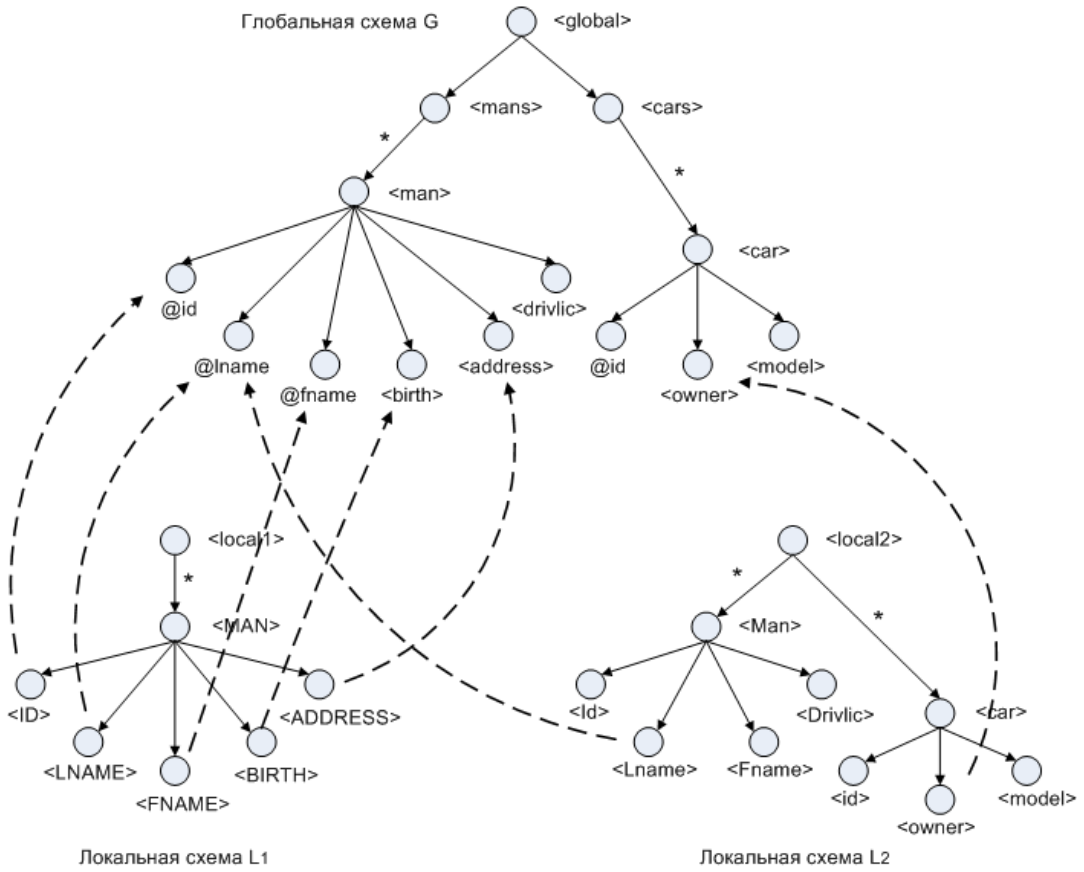

Рис.1. Пример отображения локальных схем данных на глобальную.

Рассмотрим пример сопоставления схем данных, схематично изображенный на Рис. 1. Имеются две локальные БД, задаваемые схемами L_1 и L_2. Схема L_1 содержит структуру данных <MAN>, моделирующую персональные сведения о людях, а схема L_2 содержит структуры данных <Man> и <Car>, моделирующие соответственно сведения о водителях и транспортных средствах. Виртуальная БД, задаваемая глобальной схемой *G*, одновременно «содержит» персональные сведения о людях, водителях и транспортных средствах. Отметим, что структура <man> глобальной схемы G формируется на основе структуры <MAN> локальной схемы L_1 и структуры <Man> локальной схемы L_2. Данная структура одновременно содержит персональные сведения о человеке и сведения о человеке как о водителе.

Приведенный пример показывает, что при решении задачи интеграции данных типичной является ситуация, когда одни и те же объекты реального мира моделируются при помощи разных структур данных. При этом подобные структуры содержат неполные (частичные) сведения.

Планы выполнения распределенных запросов

Настройка механизма построения планов выполнения запросов к виртуальной БД напрямую зависит от результата сопоставления локальных схем данных на глобальную. План выполнения запроса представляет собой дерево, вершинам которого приписаны операции над данными. Операции могут выполняться в произвольном порядке, но с учетом соблюдения следующего принципа. Сначала выполняются операции, приписанные дочерним узлам дерева, а за тем - родительским. Входными параметрами родительской операции служат результаты выполнения ее дочерних операций. Результат выполнения запроса по определению совпадает с результатом выполнения операции, приписанной корню его плана.

При проектировании виртуальной БД на основе платформы XML, XQuery можно ограничиться тремя типами операций: `SimpleQuery`, `Merge` и `Reconstruct`. На Рис. 2 схематично изображен план выполнения запроса к виртуальной БД из ранее рассмотренного примера. Результатом данного запроса является текущее состояние виртуальной БД (все возможные записи, касающиеся персональных сведений о людях и сведений о транспортных средствах).

В рамках операций `SimpleQuery` выполняются запросы к локальным БД. Для формирования элементов `car` достаточно обратиться только к локальной БД L_2, а для формирования элементов `man` нужно выполнять одновременно подзапросы к двум локальным БД. Операция `Merge` осуществляет слияние результатов подзапросов к локальным БД L_1 и L_2 по атрибутам `@lname`, `@fname`. Корневая операция `Reconstruct` создает итоговый XML документ. В основе реализации операции `Reconstruct` лежит функциональность стандартного XQuery процессора.

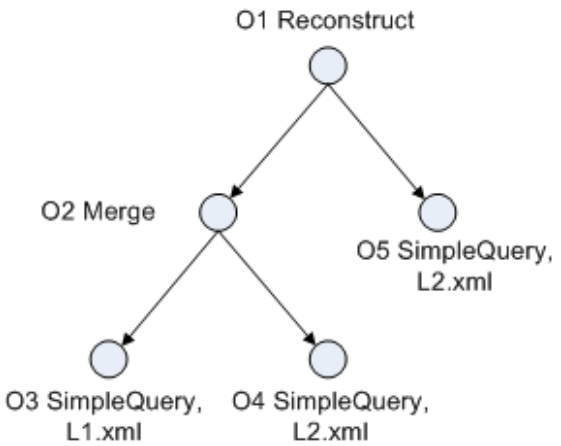

Рис.2. Пример плана выполнения запроса.

От баз данных к информационным ресурсам

Описанный подход к построению архитектуры виртуальной БД является типичным примером решения задачи интеграции данных при создании (модернизации) корпоративных приложений. Рассмотрим возможность распространения данного подхода на случай межкорпоративного взаимодействия.

Межкорпоративное взаимодействие

Одним из ограничений архитектуры виртуальной БД является отсутствие возможности заранее предсказать вид (охарактеризовать структуру) запроса к локальной БД. Такой запрос формируется в два этапа. Первый этап осуществляется посредником и связан с формированием плана выполнения запроса виртуальной БД. В результате формируются запросы, приписанные операциям `SimpleQuery`. Такие запросы не могут непосредственно выполняться в локальных БД. На втором этапе уже обертки транслируют запросы, приписанные операциям `SimpleQuery`, в запросы (серии запросов), заданные на языке манипулирования данными локальных БД.

Приведенное ограничение является существенным препятствием для использования механизмов виртуальных БД при организации межкорпоративного взаимодействия. Как правило, владелец информационной системы (далее - ИС) жестко фиксирует протокол взаимодействия с ней для внешних пользователей. Такой протокол включает ограниченный набор шаблонов запросов, которые допустимы к выполнению, а также бизнес-логику работы с ИС. Бизнес-логика задает допустимые последовательности выполнения запросов. О «произвольных» запросах, задаваемых с помощью таких выразительных языковых средств как SQL или XQuery, речь не идет. Это говорит о том, что математическое обеспечение (базовые модели и алгоритмы) основных механизмов, используемых при решении задачи интеграции данных, должно быть переработано для случая межкорпоративного взаимодействия.

Остановимся на следующей модели межкорпоративного взаимодействия. Выделим три типа субъектов взаимодействия: *владелец*, *поставщик* и *потребитель* информационного ресурса (далее - ИР). Владелец через свою ИС формирует, поддерживает в актуальном состоянии и предоставляет заинтересованным потребителям некоторую значимую информацию. В дальнейшем, под ИР будем понимать как сам массив подобной значимой информации, так и ИС владельца. В силу ряда причин (организационного, правового, финансового или технического характера) ИС потребителя может взаимодействовать либо напрямую с ИР, либо через ИС посредника. Информационная система посредника может расширять базовый стандартный перечень услуг по использованию ИР. Кроме того, ИС посредника может предоставлять доступ одновременно к нескольким ИР, относящимся к одной или разным предметным областям.

В рамках описанной модели ИР является прямым аналогом локальной БД, но уже на межкорпоративном уровне. Целевые задачи, которые призвана решать ИС посредника, являются обобщениями задач, поставленными перед виртуальной БД. Наконец, ИС потребителя является аналогом корпоративного приложения, в составе которого используются механизмы виртуальной БД и осуществляется интеграция локальных БД.

Виртуальный информационный ресурс

Опишем основные задачи, которые требуются решить для обобщения архитектурных, алгоритмических и программных решений, используемых при построении виртуальных БД, на случай межкорпоративного взаимодействия. По аналогии с понятием виртуальной БД будем использовать понятие *виртуального* ИР. Под виртуальным ИР будем понимать программный механизм, который может использоваться в составе ИС поставщиков. Для конечных потребителей подобный механизм позволит в унифицированном (возможно более доступном) виде взаимодействовать с одним или несколькими ИР.

Как и в случае с БД, важным этапом проектирования ИР является исследование предметной области и создание ее онтологии. В случае БД результатом является построение схемы БД.

Специфика ИР состоит в том, что он включает в себя ограниченный набор шаблонов запросов, которые могут выполняться. Поэтому модель предметной области ИР должна включать описания входных и выходных параметров таких запросов, а также правил их формирования. Таким образом, предметная область может задаваться некоторой частичной алгеброй $D = (A; \xi_1, \xi_2, \ldots, \xi_n)$. Элементы алгебры выполняют роль допустимых входных параметров для запросов ИР, а конечные последовательности элементов алгебры используются для задания результатов выполнения запросов. Частичные операции позволяют моделировать процедуру формирования входных параметров запросов и процедуру обработки выходных параметров. Сам ИР может быть представлен в виде набора $R = (D; Q_1, Q_2, \ldots, Q_m)$, где Q_i $(i = 1, \ldots, m)$ - частичные функции, заданные на A со значением в A, представляют собой шаблоны запросов к ИР.

По своей сути, виртуальный ИР представляет собой механизм, позволяющий изменить (с целью улучшения, упрощения, дополнения) стандартный перечень услуг по использованию одного или нескольких реальных ИР. Он также может быть представлен в виде соответствующего набора $V = (D'; Q_1', Q_2', \ldots, Q_{m'}')$. Отличие состоит только в реализации.

В рамках выбранной модели межкорпоративного взаимодействия за реализацию шаблонов ИР отвечает его владелец. Посреднику известна только сигнатура подобных шаблонов. Обладая этой информации и возможностью выполнять запросы к ИР, он на их базе реализует собственный набор шаблонов запросов уже виртуального ИР.

Заключение

В заключение сформулируем список наиболее важных задач, требующих своего решения в рамках разработки механизмов создания виртуальных ИР:

- *Сопоставление ИР*. Заданы два ИР R_1 и R_2. Требуется создать перезаписи шаблонов запросов R_1 через шаблоны R_2. Данная задача является аналогом задач унификации моделей данных и сопоставления схем данных, возникающих при построении виртуальных БД. Ее решение является теоретической основой для построения программного механизма оберток ИР.
- Построение виртуального ИР над набором ИР из одной предметной области. Заданы n $(n > 1)$ копий одного ИР R. Требуется построить операцию «объединения» таких копий. Интерпретация этой задачи следующая. Значимая для внешних потребителей информация может быть распределена (возможно с дубликатами) между несколькими «однотипными» ИР. С практической точки зрения внешний потребитель заинтересован в работе со всем объемом информации и взаимодействием с одним ее источником, а не несколькими.
- *Построение виртуального ИР над набором ИР из разных предметных областей*. Данная задача является логическим продолжением предыдущих. Заданы два ИР R_1 и R_2, относящимся к «разным» предметным областям. Требуется построить виртуальный ИР V, реализация шаблонов запросов которого базируется на вызовах запросов к R_1 и R_2. Решение этой задачи должно базироваться создании языка определения виртуальных ИР.

Список литературы

[1] Friedman T., Beyer M., Bitterer A. Magic Quadrant for Data Integration Tools. Gartner RAS Core Research Note G00160825, 2008. - http://mediaproducts.gartner.com/reprints/sas/vol5/article4/article4.html

[2] Методы и средства интеграции неоднородных баз данных. Калиниченко Л.А. / Под ред. Л.Н. Королева. - М.: Наука. Главная редакция физико-математической литературы, 1983 – 424 с.
[3] D. Chamberlin, D. Draper, M. Fernandez, M. Kay. XQuery from the Experts. - Boston: Addison-Wesley, 2003.
[4] К.В. Антипин, А.В. Фомичев, М.Н. Гринев и др. Оперативная интеграция данных на основе XML: системная архитектура BizQuery. Труды Института системного программирования (5), 2004.
[5] M. Fernandez, Y. Kadyska, D. Suciu. SilkRoute: A Framework for Publishing Relational Data in XML. ACM Transactions on Database Systems, 27(4): 438–493, 2002.
[6] W3C. XML Representation of a Relational Database. - http://www.w3c.org/XML/RDB.html
[7] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, J. Widom. The TSIMMIS project: Integration of heterogeneous information sources. Proceedings of the 10th Meeting of the Information Processing Society of Japan. pp. 7-18. Tokyo, Japan, October 1994.
[8] D. Engmann, S. Massmann. Instance Matching with COMA++ BTW 2007 Workshop: Model Management und Metadaten. Verwaltung 2007.