

Вісник Харківського національного університету
Серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи
управління»
УДК 004.652/.942 № 629, 2004, с.48-55

Об одном подходе к интеграции неоднородных информационных ресурсов

Альмхерат Ахмад Али (Абдель Карим), Г. Н. Жолткевич, С. Ю. Игнатов
Харьковский национальный университет им. В.Н. Каразина, Украина

In the present paper the problem of the information resources integration is considered. An approach to solving this problem is proposed. The approach is based on Publish/Subscribe strategy. Two modification of this strategy realization are considered.

Одной из основных проблем, с которой сталкиваются крупные организации в настоящее время, является проблема интеграции управленческих информационных ресурсов. Сложность этой проблемы усугубляется частыми изменениями управленческих взаимодействий и прав доступа как ограниченных, так и полных к информационным ресурсам. Наиболее разработанными и применяемыми для задач управления являются архитектуры корпоративных и распределенных баз данных (БД).

В отличие от традиционной, хорошо управляемой и безопасной среды вычислений предприятия, построенной на базе универсальной вычислительной машины (мейнфрейм) с подсоединенными к ней терминалами, среда локальных сетей персональных компьютеров (ПК) требует изощренных систем контроля, управления и безопасности. В то же время, развитые средства сетевой организации делают возможным разделение бизнес-информации как внутри групп индивидуальных пользователей, так и между такими группами, внутри и вне корпорации и облегчают организацию информационных процессов в масштабе предприятия. Чтобы ликвидировать брешь между отдельными локальными сетями ПК и традиционными средствами вычислений, а также для организации распределенных вычислений в масштабе предприятия, появилась модель вычислений на базе рабочих групп при единой (корпоративной) БД.

В современном подходе к реализации корпоративных БД взаимозаменяемо используются понятия серверов рабочих групп и сетевых серверов. Сервер рабочей группы может быть сервером, построенным на одном процессоре компании Intel, или суперсервером (с несколькими центральными процессорами), подобным изделиям компаний Compaq, HP, IBM и DEC, работающим под управлением операционной системы Windows NT. Это может быть также UNIX-сервер начального уровня компаний Sun, HP, IBM и DEC.

Основу следующего уровня современных информационных систем предприятий и организаций составляют корпоративные серверы различного функционального назначения, построенные на базе операционной системы UNIX. Архитектура этих систем варьируется в широких пределах в зависимости от масштаба решаемых задач и размеров предприятия. Двумя основными проблемами построения вычислительных систем для критически важных приложений, связанных с обработкой транзакций, управлением базами данных и

обслуживанием телекоммуникаций, являются обеспечение высокой производительности и продолжительного функционирования систем. Наиболее эффективный способ достижения заданного уровня производительности – применение параллельных масштабируемых архитектур. Задача обеспечения продолжительного функционирования системы имеет три составляющих: надежность, готовность и удобство обслуживания. Все эти три составляющих предполагают, в первую очередь, борьбу с неисправностями системы, порождаемыми отказами и сбоями в ее работе.

Понятно, что в общем случае для того, чтобы прикладная программа, выполняющаяся на рабочей станции, могла запросить услугу у некоторого сервера, требуется, как минимум, наличие некоторого интерфейсного программного слоя, поддерживающего такого рода взаимодействие (было бы, по меньшей мере, неестественно требовать, чтобы прикладная программа напрямую пользовалась сетевыми примитивами транспортного уровня). Из этого, собственно, и вытекают основные принципы системной архитектуры клиент/сервер.

Система разбивается на две части, которые могут выполняться в разных узлах сети – клиентскую и серверную части. Прикладная программа или конечный пользователь взаимодействуют с клиентской частью системы, которая в простейшем случае обеспечивает высокоуровневый логический протокол. Клиентская часть системы при необходимости обращается по сети к серверной части. Заметим, что в развитых системах сетевое обращение к серверной части может и не понадобиться, если система может предугадывать потребности пользователя, и в клиентской части содержатся данные, способные удовлетворить его следующий запрос (кэширование).

Альтернативой корпоративным являются распределенные БД, основной задачей систем управления которыми является обеспечение средств интеграции локальных баз данных, располагающихся в некоторых узлах вычислительной сети, с тем, чтобы пользователь, работающий в любом узле сети, имел доступ ко всем этим базам данных как к единой базе.

Таким образом, задача интеграции неоднородных БД состоит в предоставлении пользователям интегрированной системы некоторой глобальной схемы БД, представленной средствами какой-либо модели данных, и автоматическом преобразовании операторов манипулирования данными глобального уровня в операторы, понятные соответствующим локальным СУБД. В теоретическом плане проблемы преобразования решены, имеются реализации [1, 2].

Следует отметить, что при строгой интеграции неоднородных БД локальные системы утрачивают свою автономность. После включения локальной БД в федеративную систему все дальнейшие действия с ней, включая администрирование, должны вестись на глобальном уровне. Поскольку пользователи часто не соглашаются утрачивать локальную автономность, требуя, тем не менее, предоставления возможности взаимодействия со всеми локальными БД на одном языке и формулировать запросы с одновременным указанием разных локальных БД, развивается направление мульти-БД (например, System R и System R* [3, 4]). В системах мульти-БД не поддерживается глобальная схема интегрированной БД, и применяются

специальные способы именованя для доступа к объектам локальных БД. Как правило, в таких системах на глобальном уровне допускается только выборка данных. Это позволяет сохранить автономность локальных БД.

Реализации информационных систем управления с использованием описанных подходов содержат массу дорогостоящих аппаратно-технических и программно-системных решений сопоставимых по стоимости с затратами на запуск небольшого спутника.

При условии одновременного проектирования территории, зданий, сооружений, управляющих узлов и коммуникаций организации, ее бизнес-процессов подход, базирующийся на корпоративной БД, является предпочтительным по сравнению с подходом, в основе которого лежит распределенная БД. Распределенные же БД, в силу сложности их сопровождения и администрирования, не находят применения в информационных управляющих системах бизнес-структур. При условиях частых частичных модернизаций существующих систем и подсистем управления, реорганизации, создания или переподчинения подразделений существующей организации, не обладающей изначально корпоративной сетью, классические корпоративный или распределенный подходы являются дорогостоящей утопией.

Следует особо подчеркнуть, что исходная посылка разработки описанных выше архитектурных решений состоит в требовании предоставления физической возможности доступа к любой информации, хранящейся в БД, любому пользователю. Ограничение доступа обеспечивается программным путем за счет специальной подсистемы управления безопасностью. В то же время, в реальной системе управления предприятием не существует ни одного рабочего места (управляющей единицы), которое объективно нуждалось бы в полном доступе ко всем информационным ресурсам общей управляющей системы.

Это противоречие, лежащее в основе концепции обеспечения доступа к информационным ресурсам как для подхода, базирующегося на корпоративной БД, так и для подхода, основанного на распределенной БД, стало одним из побудительных мотивов возникновения проекта, направленного на создание системы интеграции гетерогенных информационных ресурсов SIRIUS.

Роль второго побудительного мотива сыграла необходимость интеграции управленческих информационных ресурсов в Харьковском национальном университете имени В. Н. Каразина, особенно остро проявившаяся в связи с расширением международного сотрудничества университета, модернизацией основных процессов и управления ими на основе принципов, сформулированных в Magna Charta Universitatum [5, 6, 7].

Сразу следует оговорить, что проект SIRIUS базируется на допущении, что информационная управляющая система является квазистатичной, иными словами изменение состояния отдельного информационного ресурса не приводит к необходимости немедленной (или достаточно быстрой) реакции на это изменение. Например, в случае управления контингентом при изменении базы данных студентов на основании приказа об отчислении, время полной реакции системы может составлять несколько дней без существенного нарушения целостности данных.

Интеграция управленческих приложений в рамках проекта SIRIUS основывается на архитектурном инварианте бизнес-приложений, модель которого показана рис. 1 [8].



Рис. 1. Архитектура бизнес-приложения

При этом физически бизнес-приложение может быть реализовано либо на локальном компьютере, либо в локальной сети, либо в корпоративной сети, в том числе обеспечивающей доступ к информационному ресурсу посредством Web.

В настоящей работе под информационным ресурсом понимается любая структура данных, удовлетворяющая какой-либо модели хранения данных, которая, как правило, но необязательно, является базой данных.

Под программным интерфейсом информационного ресурса понимается программный комплекс, обеспечивающий доступ к элементам данных информационного ресурса на уровне модели хранения данных. Для информационных ресурсов, реализованных средствами стандартных систем управления базами данных (СУБД) таким программным интерфейсом может быть сама СУБД или соответствующие библиотеки объектов доступа к данным, например, ODBC, DAO, JDBC и тому подобное. В терминологии работы [9] программный интерфейс информационного ресурса реализует слой функций доступа к данным системы.

Бизнес-логика приложения представлена программными объектами, моделирующими процессы и информационные потоки в системе управления и обеспечивающими обработку данных информационного ресурса в соответствии с правилами, установленными для объекта управления. Это соответствует слоям прикладных функций и интегральной логики системы [9].

Объекты пользовательского интерфейса ответственны за правильное отображение данных и обеспечение взаимодействия с пользователем. В работе [9] эта ответственность возложена на два слоя системы – собственно пользовательский интерфейс и презентационную логику.

В настоящей работе мы не проводим такого детального расслоения, поскольку бизнес-логика и пользовательский интерфейс остаются за рамками интеграционных процессов.

Для обеспечения интеграции отдельных бизнес-приложений предлагается минимальная модификация их архитектуры путем добавления слоев, ответственных за интеграционные процессы (см. рис. 2).

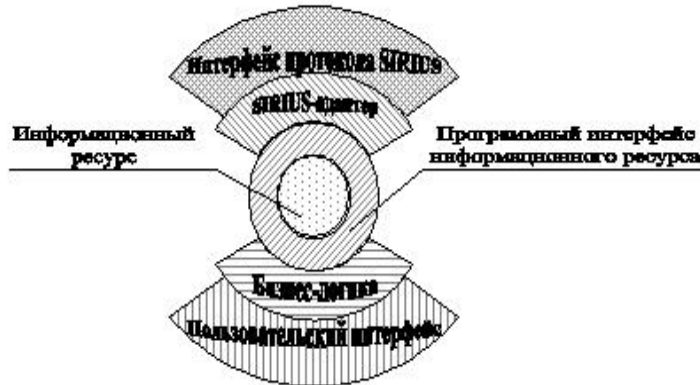


Рис.2. Архитектура информационного компонента в системе SIRIUS.

Бизнес-приложение, реализованное как информационный компонент, обязано поддерживать интерфейс протокола системы SIRIUS, посредством которого обеспечивается интеграция этого приложения в единую систему управления.

Такая архитектура системы управления обеспечивает ее открытость и масштабируемость, однако ничего не говорит о механизмах взаимодействия информационных компонентов и синхронизации данных информационных ресурсов.

Эти механизмы должны обеспечить суверенитет каждого информационного компонента над своим информационным ресурсом. Это означает невозможность модификации данных информационного ресурса никакими программными средствами кроме тех, которые инкапсулированы в информационном компоненте, а также установление только средствами информационного компонента фильтров доступа к данным информационного ресурса со стороны других компонентов.

Такой суверенитет надежно можно реализовать путем отказа от предоставления возможности on-line доступа к данным информационного ресурса. Именно это соображение диктует ограничение на применимость архитектуры SIRIUS только в случае квазистатических систем.

В случае таких систем обеспечить решение задачи синхронизации данных можно путем использования стратегии Publish/Subscribe.

Стратегия Publish/Subscribe в обобщенном виде может быть описана следующим протоколом, участниками которого являются ИЗДАТЕЛЬ и ПОДПИСЧИК:

1. информационный компонент ИЗДАТЕЛЬ посылает *уведомление* всем информационным компонентам, известным как его ПОДПИСЧИКИ, о

- факте изменения данных, над которыми установлен суверенитет ИЗДАТЕЛЯ;
2. получив *уведомление*, ПОДПИСЧИК отправляет соответствующий *запрос* ИЗДАТЕЛЮ;
 3. получив *запрос*, ИЗДАТЕЛЬ проверяет права доступа автора запроса (ПОДПИСЧИКА), формирует соответствующий отчет и отправляет его этому ПОДПИСЧИКУ;
 4. получив *отчет*, ПОДПИСЧИК преобразует его в транзакцию собственного информационного ресурса и выполняет ее.

Этот протокол может быть реализован несколькими способами, что приводит к двум типам интеграции информационных систем, которые мы условно можем назвать *сообществами* и *федерациями*.

В сообществах поддерживается полностью симметричная стратегия, при которой взаимодействие между ИЗДАТЕЛЯМИ и ПОДПИСЧИКАМИ происходит напрямую, без ПОСРЕДНИКОВ. Основной проблемой этого типа интеграции является построение надежного механизма распознавания и адресации компонентов информационной системы – членов сообщества.

В отличие от сообщества, в случае федерации в системе присутствует специальный информационный компонент – ПОСРЕДНИК, задачей которого является централизованная координация взаимодействия между всеми компонентами информационной системы. Это приводит к следующей модификации протокола Publish/Subscribe (участником его, кроме ИЗДАТЕЛЕЙ и ПОДПИСЧИКОВ, является ПОСРЕДНИК):

1. информационный компонент ИЗДАТЕЛЬ посылает *уведомление* ПОСРЕДНИКУ о факте изменения данных, над которыми установлен суверенитет ИЗДАТЕЛЯ;
2. получив уведомление, ПОСРЕДНИК определяет ПОДПИСЧИКОВ автора этого уведомления и рассылает им это уведомление;
3. получив уведомление, ПОДПИСЧИК отправляет соответствующий *запрос* ПОСРЕДНИКУ;
4. получив запрос, ПОСРЕДНИК проверяет права доступа автора запроса (ПОДПИСЧИКА) и отсылает его ИЗДАТЕЛЮ – автору уведомления, в ответ на которое был отправлен запрос;
5. получив запрос, ИЗДАТЕЛЬ формирует соответствующий отчет и отправляет его этому ПОСРЕДНИКУ;
6. получив отчет, ПОСРЕДНИК отсылает его ПОДПИСЧИКУ – автору запроса;
7. получив отчет, ПОДПИСЧИК преобразует его в транзакцию собственного информационного ресурса и выполняет ее.

В отличие от случая сообщества, федеративный протокол решает проблему распознавания, синхронизации и адресации информационных компонентов, но предъявляет определенные требования к централизации вычислительной сети организации, а также ее администрированию. Это, однако, не нарушает суверенитет информационных компонентов над своими информационными ресурсами. Федеративный подход позволяет также решить проблему запроса обязательных отчетов специальными компонентами безотносительно к тому, происходили ли какие-либо изменения данных информационных ресурсов. Это

свойство является существенным для систем управления организациями, поскольку правила бизнес-логики предусматривают предоставление таких отчетов.

Кроме того, следует отметить, что построение распределенной информационной системы, базирующейся на стратегии Publish/Subscribe, дает возможность восстановления существенной части информационного ресурса в случае его разрушения, как для подписчиков, так и для издателя.

Итак, по мнению авторов, построение распределенной информационной системы, базирующейся на стратегии Publish/Subscribe, позволяет:

1. интегрировать информационные ресурсы, сохраняя реализацию обработки данных средствами соответствующих информационных компонентов;
2. обеспечить суверенитет информационных компонентов над собственными информационными ресурсами;
3. обеспечить дополнительные возможности восстановления разрушенных информационных ресурсов;
4. снизить затраты на интеграцию информационных ресурсов и средств обработки данных за счет максимального использования существующих в организации локальных и сетевых решений.

Следует также подчеркнуть, что федеративный подход к интеграции информационных ресурсов на базе стратегии Publish/Subscribe является более адекватным для информационных управляющих систем по сравнению с подходом на основании протокола сообщества.

Открытыми проблемами предложенного подхода являются количественные оценки эффективности передачи данных при разных конфигурациях взаимодействия между информационными компонентами и оценка общего количества шаблонов уведомлений и запросов, циркулирующих в построенной таким образом системе.

ЛИТЕРАТУРА

1. Client/Server Software Architectures. An Overview. –Software Engineering Institute. – http://www.sei.cmu.edu/str/descriptions/clientserver_body.html.
2. Ozsu M. T., Valduriez P. Principles of Distributed Database Systems (2nd Edition). – Prentice Hall, 2004. – 658 p.
3. Traiger I. L., Gray J., Galtiel C. A., Lindsay B. G. Transactions and Consistency in Distributed Database Systems. – ACM Trans. Database Syst., vol. 7, № 3. – 1982. – Pp. 323-342.
4. Mohan C., Lindsay B., Obermarck R. Transaction management in the R* distributed database management system. – ACM Trans. Database Syst., vol. 11, №4. – 1986. – Pp. 378-396.
5. Magna Charta Universitatum. – <http://www.magna-charta.org/magna.html>.
6. Концепція стратегічного розвитку Харківського національного університету імені В. Н. Каразіна у 2005 – 2010 рр. – ХНУ ім. В. Н. Каразіна. – 2004. – <http://www-ukr.univer.kharkov.ua/information/rozvytok.php>.
7. Основні напрями розвитку інформаційних і комунікаційних технологій у Харківському національному університеті імені В. Н. Каразіна на період до

2010 року. – ХНУ ім. В. Н. Каразіна. – 2004. – <http://ckt.univer.kharkov.ua/index.php?page=22&lng=3>.

8. Васкевич Д. Стратегии КЛИЕНТ/СЕРВЕР. – К.: Діалектика, 1996. – 384 с.
9. Мацяшек Л. А. Анализ требований и проектирование систем. Разработка информационных систем с использованием UML. – М.: Вильямс, 2002. – 428 с.