

**БАЛТИЙСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ «ВОЕНМЕХ» имени Д. Ф. Устинова**

**Интерфейсы взаимодействия
информационных систем с базами данных.**

Автор: Рассолов М. М.

Санкт-Петербург
2009 год

Сегодня одной из самых больших возможностей повышения эффективности работы компании является использование различных информационных систем. Внедрение ИС в работу позволяет оптимизировать работу предприятия практически во всех областях его деятельности: от маркетинговых исследований и управления цепочками поставок до организации работы с клиентами.

Федеральный закон Российской Федерации от 27 июля 2006 г. N 149-ФЗ «Об информации, информационных технологиях и о защите информации» даёт следующее определение информационной системы: «Информационная система — совокупность содержащейся в базах данных информации и обеспечивающих ее обработку информационных технологий и технических средств». Таким образом, основой любой ИС является информация, содержащаяся в базе данных, а задача программной и аппаратной частей ИС заключается в эффективной обработке этой информации.

Все ИС можно классифицировать по архитектуре:

- настольные, или локальные ИС, в которых все компоненты (БД, СУБД, клиентские приложения) работают на одном компьютере;
- распределённые ИС, в которых компоненты распределены по нескольким компьютерам.

Наибольшее практическое применение имеет клиент-серверная технология построения распределённых ИС, в основе которой лежат две идеи:

1. много организационно и физически распределённых пользователей, одновременно работающих с общими данными — общей базой данных (пользователи с разными именами, в том числе располагающимися на различных вычислительных установках, с различными полномочиями и задачами);
2. общие для всех пользователей данные на одном или нескольких серверах;

Правильное понимание этой технологии основано на выделении в структуре информационных систем на трёх компонентах:

- компонент представления, реализующий функции ввода и отображения данных, называемый иногда еще просто как интерфейс;
- прикладной компонент, включающий набор запросов, событий, правил, процедур и других вычислительных функций, реализующий предназначение автоматизированной информационной системы в конкретной предметной области;
- компонент доступа к данным, реализующий функции хранения, извлечения, физического обновления и изменения данных (машина данных).

Исходя из особенностей реализации и распределения (расположения) в системе этих трех компонентов различают четыре модели технологий «Клиент-сервер»:

- модель файлового сервера;
- модель удалённого доступа к данным;
- модель сервера базы данных;
- модель сервера приложений.

В этом реферате я рассмотрю различные принципы взаимодействия ИС с базой данных на примере ERP и CRM системы компании Microsoft под названием Dynamics NAV.

Microsoft Dynamics NAV представляет собой интегрированную комплексную систему управления предприятием, объединяющую возможности финансового управления, анализа состояния бизнеса, управления производством, дистрибуцией и электронной коммерцией, а также взаимоотношениями с клиентами.

Посмотрим, каким образом перечисленные выше модели клиент-серверной технологии реализованы в системе от Microsoft.

1. Модель удаленного доступа к данным.

Microsoft Dynamics NAV может работать на Microsoft SQL Server 2000 и Microsoft SQL Server 2005. При работе на Microsoft SQL Server вся бизнес-логика приложения помимо SIFT (SumIndexFields — технология индексного суммирования) выполняется на клиенте. Таким образом, такое построение системы соответствует модели удаленного доступа к данным клиент-серверной архитектуры. В такой модели компонент доступа к данным полностью отделен от двух других компонентов (компонента представления и прикладного компонента) и размещается на сервере системы. В нашем случае СУБД MS SQL размещается на центральном сервере (или нескольких серверах) и просто обеспечивает структурированный доступ к данным для клиентов, на которых расположен компонент представления и основная часть прикладного компонента. Основная, потому что несколько модулей системы интегрированы в MS SQL (например, SIFT и OLAP — OnLine Analytical Processing). Сделано это для значительного ускорения обработки больших объёмов данных. SIFT представляет собой индексное суммирование, благодаря которому БД быстро производит числовые вычисления над сложными данными, а OLAP — это средства оперативной аналитической обработки данных.

В данной модели клиентское приложение оказывается довольно «толстым»: функции по обработке поступающей информации он берёт на себя, поэтому от клиента требуются повышенные вычислительные мощности, сервер же остается не загруженным.

К преимуществам модели удаленного доступа к данным относятся масштабируемость и возможность интеграции с внешними приложениями. Масштабируемость проявляется в возможности увеличения размера БД практически до любого нужного размера, а также увеличении числа одновременно работающих пользователей/клиентов. Интеграция осуществима благодаря возможности работы MS SQL Server как с Microsoft Dynamics NAV,

так и с многими продуктами других производителей. Таким образом, все эти продукты могут без труда использовать единую БД и иметь прозрачный доступ к любой необходимой информации.

К недостаткам такой модели можно отнести плохую реализуемость или невозможность функций, которые могут быть органично включены во встроенный сервер БД. Кроме того, из-за того, что не все операции выполняются на сервере, много необработанной информации пересылается клиентам, что значительно повышает нагрузку на сеть.

2. Модель сервера базы данных.

Проблемы модели удаленного доступа к данным решает встроенный сервер БД. В такой модели компонент доступа к данным и прикладной компонент тесно связаны друг с другом и располагаются на одном сервере. Приложения на клиентских станциях содержат в себе только компонент представления, что существенно снижает требования к вычислительной установке клиента. Такой клиент называется тонким, т. е. на него возложена относительно простая функция запроса и отображения данных. Пользователь через интерфейс системы на клиентской установке направляет на сервер базы данных только лишь вызовы необходимых процедур, запросов и других функций по обработке данных. Все затратные операции по доступу и обработке данных выполняются на сервере и клиенту направляются лишь результаты обработки, а не наборы данных, как в модели удаленного доступа к данным. Этим обеспечивается существенное снижение трафика сети в этой модели по сравнению с моделью удаленного доступа к данным.

В системе Microsoft Dynamics NAV наиболее эффективной с точки зрения производительности является конфигурация со встроенным сервером, которая и представляет собой реализацию модели сервера базы данных. Благодаря тесной связи компонента доступа к данным и прикладного компонента в рассматриваемой системе удалось реализовать дополнительные функции.

Например, систему контроля версий, позволяющую каждому пользователю или приложению работать со своей уникальной версией данных, которая является «снимком» БД в момент начала определённого действия. Она обеспечивает работу большого количества пользователей, не нарушая целостности данных и не блокируя доступ к данным других пользователей. Важно отметить, что в конфигурации системы под управлением MS SQL Server, т. е. в модели удаленного доступа к данным, вместо принципа контроля версий сервер обеспечивает лишь блокировку таблиц и оптимистическую конкуренцию.

Таким образом, на примере системы управления предприятием Microsoft Dynamics NAV мы видим, что модель сервера баз данных позволяет оптимизировать обработку больших и сложных по структуре данных за счет тесной интеграции компонента доступа к данным и компонента, обеспечивающего их обработку.

3. Модель сервера приложений.

Суть этой модели заключается в переносе прикладного компонента ИС на специализированный в отношении повышенных ресурсов по быстродействию дополнительный сервер системы. Как и в модели с сервером базы данных, на клиентских установках располагается только интерфейсная часть системы, т. е. компонент представления. Однако вызовы функций обработки данных направляются на сервер приложений, где эти функции совместно выполняются для всех пользователей системы.

На практике же, в частности в рассматриваемой нами Dynamics NAV сервер приложений (именуемый Microsoft Dynamics NAV Application Server) имеет несколько отличную функцию. Этот компонент помогают компаниям обмениваться информацией с партнерами, клиентами и филиалами, обеспечивая интеграцию Microsoft Dynamics NAV с другими приложениями и программами, а также открывая данное решение для электронной торговли. Запросы при этом поступают от других приложений именно к серверу

приложений, который импортирует этот запрос и, используя бизнес-логику своей системы, обращается к базе данных, обрабатывает полученную информацию и посылает ответ.

4. Модель файлового сервера.

Данная модель представляет собой довольно простую схему пассивного сервера, предоставляющего централизованный доступ к файлам, и толстых клиентов, содержащих в себе компоненты представления, доступа к данным и прикладной компонент. Схема может применяться для расширения персональных (пользовательских) СУБД в направлении многопользовательского режима, хотя сразу возникает проблема разделения доступа к данным и их безопасности. В связи с этим такая модель практически не применяется в более-менее крупных ИС и в Microsoft Dynamics NAV не представлена.

Выше были рассмотрены основные модели клиент-серверной технологии построения ИС. Каждая модель приносила свой взгляд на проблему взаимодействия ИС с БД. Далее рассмотрим одну особенность построения систем, дающих дополнительные возможности работы с данными из ИС.

С целью обеспечить наибольшую гибкость в обработке и сравнении данных большинство систем поддержки процессов предприятия содержат инструменты для создания и изменения компонентов приложения. Для этого предоставляется API и довольно часто удобная среда разработки. В системе Microsoft Dynamics NAV довольно далеко ушли в этом направлении. Её ядром является интегрированная среда разработки в архитектуре клиент-сервер (Client/Server Integrated Development Environment, или C/SIDE).

Эта среда служит основой для всех функциональных средств системы Microsoft Dynamics NAV. Она содержит семь видов объектов, которые и составляют весь функционал приложения. Из подобных объектов состоят все модули Microsoft Dynamics NAV, в результате чего достигается унификация и целостность интерфейса. Таким образом, любой пользователь системы, обладающий соответствующими знаниями, имеет возможность самостоятельно

разрабатывать модули прикладного компонента, настраивая их под любые особенности работы компании. То есть с базой данных пользователь конечно по-прежнему не может работать напрямую, но у него есть достаточно функциональный API, позволяющий ему манипулировать данными практически любым необходимым образом.

В конце реферата хотелось бы подчеркнуть, что почти все возможности (кроме связанных с MS SQL Server) по интеграции рассмотренной системы с другими системами осуществимы только с приложениями от Microsoft. Поэтому, рассматривая вопрос совместного доступа различных ИС к одним данным, важно учитывать особенности их взаимодействия для каждого конкретного случая. То есть каждая ИС имеет свои особенности реализации различных моделей клиент-серверной технологии, и в связи этим взаимодействие различных систем становится проблематичным, и тесная интеграция наиболее часто оказывается возможной только между приложениями от одного производителя.

На основе рассмотренного примера мы видим, что современные ИС стараются дать пользователю возможность выбора конфигурации взаимодействия с БД. Однако принятие каждого варианта несет в себе как плюсы, так и минусы, и чаще всего становится результатом выбора между масштабируемостью и своего рода универсальности системы (в смысле интеграции и добавлении функционала) с одной стороны, и скоростью и простотой обслуживания с другой.