

ДОСТУП К БАЗАМ ДААННЫХ

Стандартные системы доступа к базам данных

Существует несколько способов доступа к данным из средств разработки и клиентских приложений [1, 2, 3]. Подавляющее большинство систем управления БД содержит в своем составе библиотеки, предоставляющие специальный *прикладной программный интерфейс* (*Application Programming Interface — API*) для доступа к данным этой СУБД. Обычно такой интерфейс представляет собой набор функций, вызываемых из клиентского приложения. В случае настольных СУБД эти функции обеспечивают чтение/запись файлов БД, а в случае серверных СУБД инициируют передачу запросов серверу БД и получение от сервера результатов выполнения запросов или кодов ошибок, интерпретируемых клиентским приложением. Библиотеки, содержащие API для доступа к данным серверной СУБД, обычно входят в состав ее клиентского программного обеспечения, устанавливаемого на компьютерах, где функционируют клиентские приложения.

Использование клиентского API (или клиентских СОМ-объектов) является наиболее очевидным (и нередко самым эффективным с точки зрения производительности) способом манипуляции данными в приложении. Однако в этом случае созданное приложение сможет использовать данные только СУБД этого производителя, и замена ее на другую (например, с целью расширения хранилища данных или перехода в архитектуру «клиент — сервер») повлечет за собой переписывание значительной части кода клиентского приложения — клиентские API и объектные модели не подчиняются никаким стандартам и различны для разных СУБД.

Другой способ манипуляции данными в приложении базируется на применении *универсальных механизмов* доступа к данным. Указанный механизм обычно реализован в виде библиотек и дополнительных модулей, называемых *драйверами* или *провайдерами*. Библиотеки содержат некий стандартный набор функций или классов, укомплектованный согласно той или иной спецификации. Дополнительные модули, специфичные для различных СУБД, реализуют непосредственное обращение к функциям клиентского API конкретных СУБД.

Отметим, что достоинством универсальных механизмов является возможность применения одного и того же абстрактного API, а во многих случаях — СОМ-серверов, компонентов, классов для доступа к разным типам СУБД. Поэтому приложения, использующие универсальные механизмы доступа к данным, легко модифицировать при необходимости замены СУБД. Причем модификация, как правило, затрагивает не код приложения как таковой, а настройки доступа к данным, содержащиеся в реестре или внешних файлах. Недостатками универсальности являются невозможность доступа к уникальной функциональности конкретной СУБД, снижение производительности приложений,

Ниже приведены наиболее популярные среди универсальных механизмов доступа к данным:

- Borland Database Engine (BDE);
- Open Database Connectivity (ODBC);
- OLE DB;
- ActiveX Data Objects (ADO).

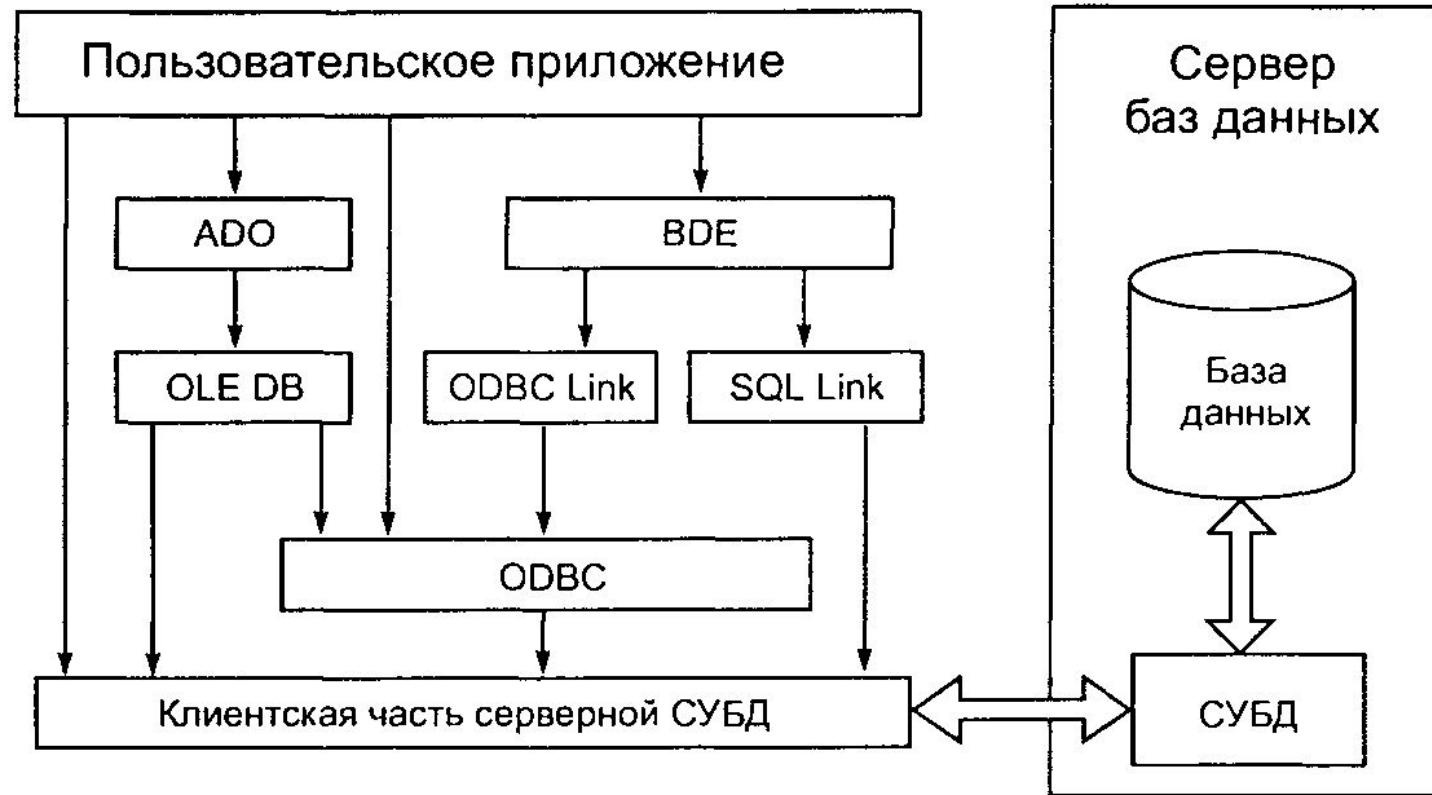


Рис. 4.1. Возможные механизмы доступа к данным из приложений

В общем случае приложение, использующее БД, может применять следующие механизмы доступа:

- непосредственный вызов функций клиентского API (или обращение к COM-объектам клиентских библиотек);
- вызов функций ODBC API (или применение классов, инкапсулирующих подобные вызовы);
- непосредственное обращение к интерфейсам OLE DB;
- применение ADO (или применение классов, инкапсулирующих обращение к объектам ADO);
- применение ADO + OLE DB + ODBC;
- применение BDE + SQL Links (или применение классов, инкапсулирующих обращение к функциям BDE);
- применение BDE + ODBC Link + ODBC.

Технология BDE

Borland Database Engine (BDE) — универсальный механизм доступа к данным, применяемый в средствах разработки фирмы Borland (а именно — Delphi и C++ Builder), а также в некоторых других продуктах, например Corel Paradox, Corel Quattro Pro, Seagate Software Crystal Reports.

Технология VDE

Физически VDE представляет собой набор библиотек доступа к данным, реализующих VDE API — набор функций для манипуляции данными, вызываемых из приложения. Эти функции, в свою очередь, могут обращаться к функциям клиентского API (в случае, например, Oracle, Informix, IBM Database)

Технология VDE

Для доступа к БД с помощью VDE на компьютере с клиентским приложением следует установить библиотеки VDE общего назначения, а также *VDE-драйвер* для данной СУБД. В случае серверных СУБД такие драйверы носят название *SQL Links*. Эти драйверы содержат VDE API — стандартный набор функций, созданных на основе функций клиентских API соответствующих СУБД.

Механизм ODBC

Open Database Connectivity (ODBC) — широко распространенный программный интерфейс фирмы Microsoft, удовлетворяющий стандартам ANSI и ISO для интерфейсов обращений к БД (Call Level Interface — CLI). Для доступа к данным конкретной СУБД с помощью ODBC, кроме собственно клиентской части этой СУБД, нужен ODBC Administrator (приложение, позволяющее определить, какие источники данных доступны для данного компьютера с помощью ODBC, и описать новые источники данных) и *ODBC-драйвер* для доступа к этой СУБД. ODBC-драйвер представляет собой динамически загружаемую библиотеку (*DLL*), которую клиентское приложение может загрузить в свое адресное пространство и использовать для доступа к источнику данных. Для каждой используемой СУБД нужен собственный ODBC-драйвер, так как ODBC-драйверы используют разные функции клиентских API для различных СУБД.

Механизм ODBC

ODBC позволяет манипулировать данными любой СУБД (и даже данными, не имеющими прямого отношения к БД, например данными в файлах электронных таблиц или в текстовых файлах), если для них имеется ODBC-драйвер. Для манипуляции данными используют как непосредственные вызовы ODBC API, так и другие универсальные механизмы доступа к данным, например OLE DB, ADO, BDE, реализующие стандартные функции или классы на основе вызовов ODBC API в драйверах или провайдерах, специально предназначенных для работы с любыми ODBC-источниками.

Механизм ODBC

Резюмируя сказанное выше, отметим, что по сравнению с другими универсальными механизмами доступа к данным применение VDE оправданно, если:

- для хранения данных используются СУБД, ранее принадлежавшие фирме Borland (Paradox, dBase);
- для создания приложений используются ранние версии средств разработки Borland вместе с Informix, IB Database, DB2, Sybase, а также ранними версиями Access и Microsoft SQL Server;
- используются объектные расширения Oracle и средства разработки Borland двух последних версий.

Компоненты для доступа к ODBC-источникам

Поскольку на данный момент ODBC является самым используемым универсальным механизмом доступа к данным, перспективы применения подобных компонентов очевидны. Производительность приложений, использующих компоненты для доступа к ODBC-источникам, обычно выше производительности приложений, которые используют BDE и ODBC Link, за счет отказа от использования дополнительных библиотек BDE. По этой же причине упрощена поставка таких приложений, поскольку не требуется включать BDE в дистрибутив и обеспечивать его настройку на компьютере пользователя.

Компоненты для доступа к ODBC-источникам

ODBCExpress (Korbitec) [4] представляет собой набор компонентов и классов для Delphi и C++ Builder, применяемых для доступа к ODBC-источникам данных и инкапсулирующих вызовы ODBC API. Для работы приложений, использующих эти компоненты и классы, требуются библиотеки ODBC (доступные на Web-сервере Microsoft) и ODBC-драйвер для выбранной СУБД. Сами компоненты и классы ODBCExpress располагаются внутри исполняемого файла приложения.

Компоненты прямого доступа к ORACLE

Oracle Data Access Components (ODAC) фирмы **CoRe Lab Software Development** представляет собой набор невизуальных компонентов для Delphi и C++ Builder, позволяющих осуществить доступ к Oracle без использования универсальных механизмов доступа к данным. Для доступа к объектам Oracle ODAC использует непосредственно API клиентской части Oracle — Oracle Call Interface (OCI). В результате повышается производительность приложений и, кроме того, при создании приложений Delphi Professional или C++ Builder Professional продукты более дешевые, чем Enterprise-версии [6].

Компоненты прямого доступа InterBase DataBase

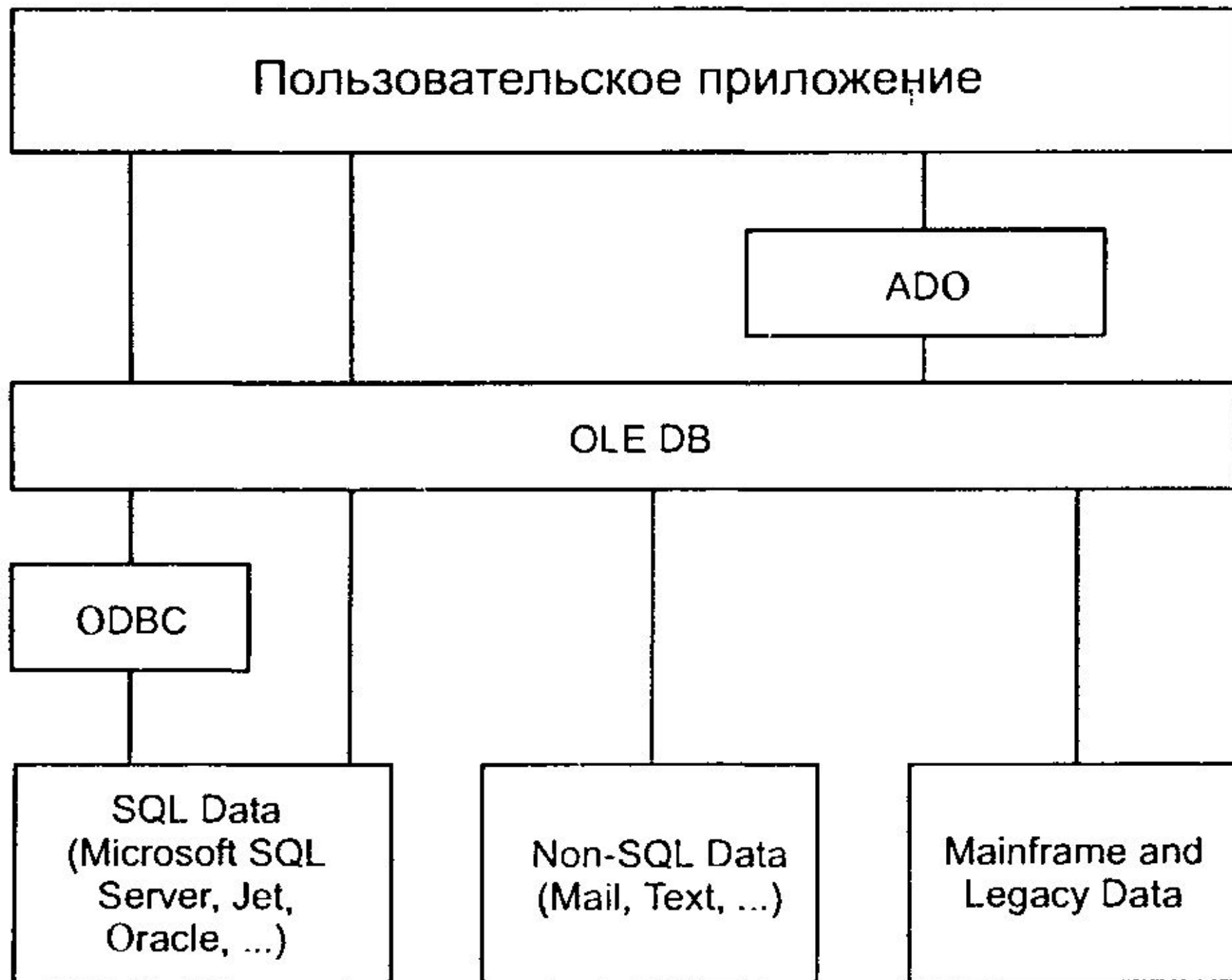
Компоненты прямого доступа к InterBase (IB) Database (как и аналогичные компоненты для Oracle) имеют неплохие перспективы применения, так как не требуют приобретения Enterprise-версий Delphi или C++ Builder, предоставляя дополнительную функциональность, специфичную для IB Database.

Компоненты Titan для доступа к различным СУБД

Компоненты *Titan* фирмы Reggatta Systems для доступа к Btrieve, Microsoft Access, Sybase SQL Anywhere используют клиентский API этих СУБД, что исключает необходимость использования других библиотек доступа к данным (BDE, ODBC и др.) и позволяет достичь приемлемой производительности [10].

Универсальный механизм доступа к данным

Универсальный механизм доступа к данным — Universal Data Access (Microsoft) [15] представляет собой стратегию предоставления доступа к любому типу информации. В результате обеспечен высокопроизводительный доступ к различным источникам информации (включая реляционные и нереляционные данные), в том числе к данным, хранящимся на серверах, данным электронной почты и файловой системы, текстовым, графическим и географическим данным и др. Вполне очевидно, что могут появляться новые форматы данных и способы их хранения, поэтому разумным требованием к универсальному механизму доступа к данным будет организация поддержки не только существующих в настоящее время форматов и источников данных, но и форматов данных в будущем.



Пользовательское приложение

ADO

OLE DB

ODBC

SQL Data
(Microsoft SQL
Server, Jet,
Oracle, ...)

Non-SQL Data
(Mail, Text, ...)

Mainframe and
Legacy Data