

торцевого динамического генератора. На рис. 5. (а, б, в) представлен ротор торцевого динамического генератора.

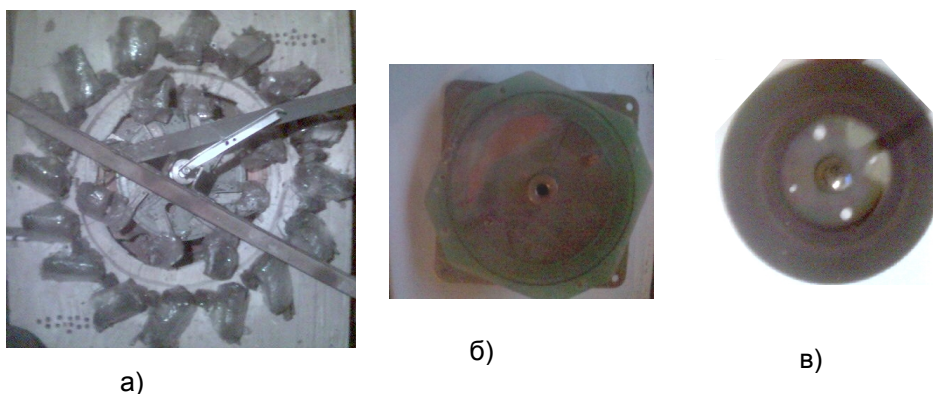


Рис. 5. Ротор торцевого динамического генератора

Выводы. Представлены некоторые полученные аналитические соотношения, полученные в результате исследований. Рассчитаны рассеянные СВЧ - электромагнитные поля отражателей ретрансляторов, в том числе цилиндрических и полуцилиндрических уголкового отражателей. Изображены рассчитанные и полученные в результате физических натурных испытаний картограммы внешнего поля систем постоянных магнитов динамических генераторов. Показаны фотографии экспериментальных макетов динамических генераторов на постоянных магнитах.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Савин В.И., Ашихмин В.Л., Юханов Ю.В., Огурцов Е.С. и др. Заявка на изобретение «Устройство для измерения и калибровки диаграмм направленности светоизлучающих устройств в плоскости». Заявка № 2007119053 от 22.05.2007.
2. Огурцов Е.С., Огурцов С.Ф. «Конструкционный синтез импедансных цилиндрических уголкового отражателей». Вестник МГОУ. – № 4. – 2001 г.
3. Огурцов Е.С., Огурцов С.Ф. Тезисы доклада «Конструктивный синтез импедансных круговых отражателей». РГУПС, 2002.
4. Огурцов Е.С., Огурцов С.Ф. Заявка на изобретение 2004129506/09 от 06.10.2004 «Электродинамическое устройство».
5. Огурцов Е.С., Огурцов С.Ф. Заявка на изобретение «Электродинамическое двухполюсное устройство» Заявка № 2007114262 от 16.04.2007.

УДК 004.658

В.А. Солодов

ТЕХНОЛОГИИ ДОСТУПА К БАЗАМ ДАННЫХ ИНФОРМАЦИОННЫХ ЭНЕРГЕТИЧЕСКИХ СИСТЕМ

Введение. Проектирование информационной инфраструктуры энергетики является одним из приоритетных направлений в области высоких технологий,

создание которой, наряду с единой энергетической системой, оказывает колоссальное влияние не только на развитие и обеспечение жизнедеятельности государства, но и на экономический потенциал страны. С развитием Интернет – ориентированных систем информационная инфраструктура принимает принципиально новый ряд компонент: средства удаленного доступа к вычислительным ресурсам, гипертекстовые базы данных, средства электронной почты, интерактивные серверные приложения – все это позволяет реализовать концепцию электронного документооборота с механизмом защищенных коммерческих транзакций. При создании информационно-управляющих систем в различных отраслях промышленности, в том числе и топливно-энергетическом секторе, жесткие требования предъявляются к технологиям хранения и доступа к данным, в большинстве случаев речь идет о технологиях баз данных. На сегодняшний день существует большое количество технологий публикации баз данных, предпочтительность которых зависит от требований, предъявляемых к приложениям, среди них: размер информационного хранилища, сложность структуры хранимых данных, быстродействие, безопасность, стоимость и др. В данной работе рассмотрены технологии публикации баз данных в приложениях, имеющих архитектуру «клиент-сервер».

Интерфейсы доступа к базам данных. Для унификации доступа к базам данных из клиентских приложений используются интерфейсы доступа к базам данных.

Интерфейс ODBC (Open Database Connectivity) был разработан компанией Microsoft для поддержки максимальной интероперабельности приложений, которая обеспечивает унифицированный доступ любого приложения, использующего ODBC, к различным источникам данных. Так, если приложение, соответствующее стандарту ODBC и SQL, первоначально разрабатывалось для работы с базой данных Microsoft Access, а затем таблицы этой базы были перенесены в базу данных Microsoft SQL Server или базу данных Oracle, то приложение сможет и дальше обрабатывать эти данные без внесения дополнительных изменений. Архитектура ODBC представлена на рис.1.

ODBC принимает от клиентских приложений вызовы функций, при этом выполняются следующие задачи: управление запросами к СУБД; выполнение передачи данных от приложения-клиента в СУБД и из базы данных в приложение-клиент; возвращение приложению-клиенту стандартной информации о выполненном вызове ODBC-функции в виде кода возврата; управление коммуникационными протоколами между приложением-клиентом и источником данных; управление транзакциями. В качестве аргументов функций, осуществляющих доступ к базам данных через ODBC, используются запросы на языке SQL (Structured Query Language).

Унифицированный доступ приложению-клиенту к различным источникам данных может быть обеспечен также при помощи OLE DB провайдеров. В отличие от доступа, предоставляемого посредством драйверов ODBC, OLE DB позволяет реализовывать доступ к источникам данных, как с применением языка SQL (к SQL-серверам), так и к любым другим произвольным источникам данных.

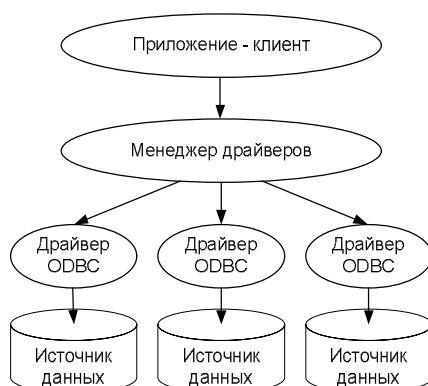


Рис.1. Архитектура ODBC

OLE DB – это метод доступа к любым данным через стандартные COM-интерфейсы, вне зависимости от типа данных и места их расположения. В качестве данных могут выступать базы данных, простые документы, таблицы Excel и любые другие источники данных. Так как архитектура OLE DB основана на COM, то механизм создания результирующих наборов состоит из последовательностей следующих шагов: создание объекта, запрос указателя на интерфейс созданного объекта, вызов метода интерфейса. Технология OLE DB лежит в основе интерфейсов доступа к базам данных DAO (Data Access Objects), RDO (Remote Data Objects), ADO (ActiveX Data Object).

Объектная модель DAO является первым объектно-ориентированным интерфейсом, который работал с Microsoft Jet и позволял программисту подключаться напрямую к таблицам Access (или к остальным БД через ODBC). DAO считается наиболее удобным средством для однопользовательских систем и небольших разработок. DAO поддерживает два различных окружения БД (рабочих пространства): Microsoft Jet и ODBCDirect.

Рабочее пространство Microsoft Jet позволяет получить доступ к базам данных Microsoft Jet, к источникам данных ODBC, которые позволяют подключаться через Microsoft Jet, а также к установленным источникам данных ISAM всех остальных форматов, например Paradox или Lotus 1-2-3. Ядро Microsoft Jet позволяет использовать базы данных MDB или прочие настольные БД ISAM и предоставляет такие возможности, как объединение данных из различных форматов БД.

Рабочее пространство ODBCDirect позволяет вам получить доступ к СУБД через ODBC без загрузки и использования ядра Microsoft Jet. ODBCDirect предоставляет возможность выполнять запросы или хранимые процедуры на сервере, а также использовать специфические возможности ODBC, такие как пакетное обновление и асинхронное исполнение запросов. С точки зрения написания клиент-серверных приложений наибольший интерес представляет рабочее пространство ODBCDirect.

Объектная модель DAO (ODBCDirect) представлена на рис.2.

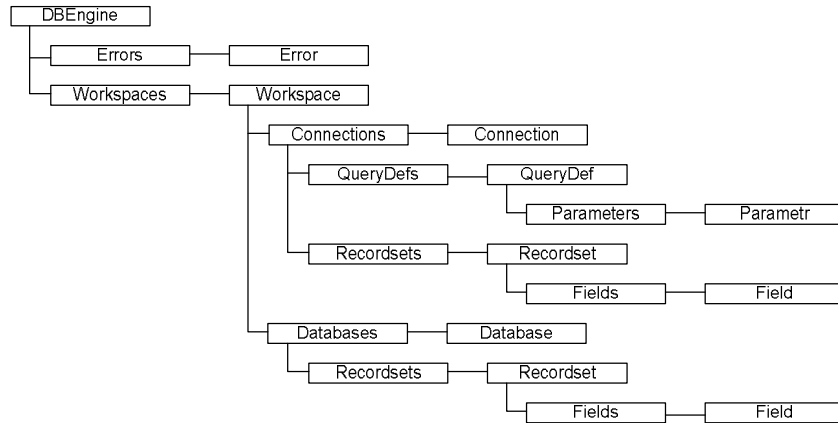


Рис. 2. Объектная модель DAO (ODBCDirect)

Объектная модель RDO является лишь объектно-ориентированной оболочкой ODBC API, а непосредственный доступ к данным выполняет ODBC-драйвер. Объектная модель RDO похожа на технологию DAO, основное отличие заключается в том, что RDO не требует дополнительной памяти для поддержки локальной базы данных. RDO предоставляет такие дополнительные функции по сравнению с DAO, как серверные курсоры, отсоединенные наборы записей и асинхронную обработку. Как и DAO, для доступа к своим объектам RDO применяет COM-интерфейсы. В настоящее время RDO практически не используется, так как ADO обеспечивает более эффективный и удобный доступ к данным. RDO является переходным между DAO и ADO, поэтому часть концепций наследуется от DAO, а часть концепций перешла впоследствии в ADO.

Объектная модель ADO призвана обеспечить доступ к наиболее часто применяемым функциям OLE DB. ADO состоит из трех основных компонентов: Connection, Command и Recordset. Объектная модель ADO приведена на рис.3.

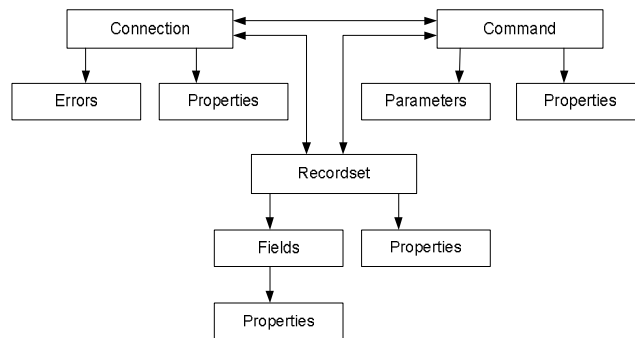


Рис. 3. Объектная модель ADO

Объект Connection устанавливает соединение между приложением и внешним источником данных, например SQL Server. Кроме того, он отвечает за инициализацию и создание подключения, выполнение запросов и механизм транзакций. Объект Command формирует запросы на выборку записей из источников данных, учитывая заданные пользователем параметры. Объект Recordset обеспе-

чивает доступ к записям, выбранным SQL-запросом, его применяют для редактирования, добавления или удаления записей в источнике данных. Три набора: Errors, Parameters и Fields, поддерживаемые ADO предоставляют приложениям дополнительные возможности. Помимо собственных свойств объекты ADO поддерживают набор Properties, состоящий из объектов Property, которые содержат специфическую информацию о поставщике данных, представленном объектом.

Варианты написания клиентских программ для доступа к базам данных. На более высоком программном уровне доступ к информации в базах данных может осуществляться следующими способами: преобразование содержимого БД в статические документы, динамическое создание html-страниц на основе информации из БД, использование серверных сценариев для доступа к БД, создание информационного хранилища на основе высокопроизводительной СУБД со встроенным SQL. Рассмотрим подробнее технологии, на основе которых могут быть реализованы все эти варианты.

При преобразовании содержимого БД в статические документы информация из БД обрабатывается специальной программой, которая формирует статические связанные html-документы. Этот вариант эффективно работает только при отсутствии жестких требований к актуальности данных и совершенно не подходит для создания интерактивных приложений.

При динамическом создании html-документов доступ к БД осуществляется специальной CGI-программой (Common Gateway Interface), запускаемой Web-сервером в ответ на запрос клиента. Эта программа, обрабатывая запрос, просматривает содержимое БД и создает выходной HTML-документ, возвращаемый клиенту. Это решение эффективно для больших баз данных со сложной структурой, а также при необходимости частого обновления страниц. Недостатками этого метода является большое время обработки запросов, необходимость постоянного доступа к основной базе данных, повышенная нагрузка на сервер и относительно низкая безопасность. Для реализации такой технологии необходимо использовать взаимодействие WWW-сервера с запускаемыми программами CGI. Создание CGI-приложений возможно при помощи языков программирования (C, Pascal, Perl), а также при помощи интегрированных в СУБД средств (наподобие генераторов отчетов).

Технология написания серверных сценариев позволяет генерировать динамические интерактивные страницы. При помощи этой технологии можно получать информацию из HTTP запроса, посылать HTTP отклик назад Web-клиенту, сохранять пользовательскую информацию и определять возможности Web-клиента в области просмотра. Наиболее распространенными технологиями написания серверных сценариев являются ASP (Active Server Pages) – в качестве Web-сервера используется IIS (Internet Information Services) и PHP (в качестве Web-сервера используется Apache). В обоих случаях для доступа к информации, хранимой в базах данных, используются объекты ADO, о которых речь шла ранее. В простейшем случае ASP или PHP-страница – это типичный ASCII файл, которые содержит HTML и программный (VBScript или Jscript) коды, хотя в более сложных случаях решение на основе ASP может объединять Web-приложения, содержащие HTML, Jscript/VBScript, SQL, ADO и COM/COM+ компоненты среди прочих.

При использовании информационного хранилища для обработки разнообразных запросов, в том числе и от Web-сервера, используется промежуточная БД высокой производительности, информационное наполнение которой осуществ-

ляется специализированным программным обеспечением на основе содержимого основных баз данных.

При использовании информационного хранилища встает вопрос о выборе качественной платформы для создания информационного хранилища. Реляционная система управления базами данных фирмы Oracle является лидером на рынке СУБД. По производительности, надежности хранения данных, развитию семейства интерфейсов, объему серверных платформ продукты Oracle возглавляют многочисленные рейтинги. Существенным ограничением использование Oracle в сфере науки и образования является достаточно высокая цена.

Заключение. Таким образом, доступ к информации, хранящейся в БД, является важнейшей составляющей при создании информационной инфраструктуры энергетики. На сегодняшний день существует множество технологий различного уровня, позволяющих не только предоставлять информацию из баз данных, но и модифицировать ее через Web-приложения. Сочетание этих технологий позволяет создавать эффективные приложения электронного документооборота с учетом различных требований, предъявляемых к ним. Вместе с тем выбор средства доступа к базе данных, как и самой базы данных, является основой приложения и от этого выбора может зависеть не только производительность, скорость разработки, но и успех проекта в целом.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Баженова И.* Oracle 8/8i. Уроки программирования. Учеб. Пособие. – М.: Диалог-МИФИ, 2001. – 288 с.
2. *Гофман В., Хомоненко А.* Delphi 6 в подлиннике. – СПб.: БХВ-Петербург, 2001. – 1152 с.
3. *Постолит А.* Visual Studio.Net: разработка приложений баз данных. – СПб.: БХВ-Петербург, 2003. – 544 с.
4. *Сигнор Р., Стегман М.* Использование ODBC для доступа к данным. – М.: БИНОМ, 1995. – 384 с.
5. *Уильямс Э., Бербер К., Ньюкирк П.* Active Server Pages. – СПб.: БХВ-Петербург, 2001. – 672 с.