

# ДАЙ ЖЕ ТЫ ВСЕМ ПОНЕМНОГУ,

# ИЛИ

## Динамическое распределение доступа к данным в ГИС

### О проблеме

Так всегда и бывает. Сталкиваясь с техническими и организационными проблемами, которые мешают нам создать "ГИС нашей мечты", мы думаем: "Вот сейчас, наконец, решим вопрос с бюджетом, введем "с бумаги" все данные, купим и установим все необходимые программы... и будет нам ГИС, и все увидят, как жизнь стала проще и радостнее".

Не тут-то было. Человек, купивший наконец автомобиль, становится более мобильным... но должен решать проблемы с парковкой и обслуживанием — иначе результаты усилий по накоплению и приобретению могут просто разочаровать.

Судите сами. "Заправлены в планшеты космические карты" — выбрали правильный путь: в мощное и эффективное хранилище на основе СУБД Oracle собраны все пространственные и описательные данные, которые заботливо и каждодневно обновляются. Само собой, для обработки сканированных карт использовался наш надежный Spotlight, данные попали в СУБД с помощью CS MapDrive. Мы об этом писали. А вы — читали.

Проект успешен, всё больше и больше людей стремится воспользоваться его результатами, и в этом-то успехе кроется следующая БОЛЬШАЯ ПРОБЛЕМА... решение для которой мы и предлагаем.

### Как быть

Итак, цифровая карта содержит несколько десятков слоев, с которыми связаны замысловатыми иерархическими связями многочисленные описательные данные. А доступ к этим данным необходимо распределить между доброй сотней разных пользователей так, чтобы каждый видел то и только то, что ему предназначено: никто ведь не отменял понятия ни государственной, ни коммерческой тайны. Что же делать — писать сто специализированных клиентских приложений, вводить в систему администрирования сотню уникальных пользователей? Боюсь, что тогда затраты и усилия на поддержание ГИС-проекта вырастут настолько, что сам бюджет на его создание перестанет поражать воображение...

Решение находится в русле развития нашей основной концепции: ГИС на основе СУБД.

Наш выбор в пользу Oracle, разумеется, остался неизменным: помимо уже многократно упоминавшихся достоинств общего порядка (высокая производительность, масштабируемость, мультиплатформенность), усилились специфические для Oracle достоинства. Уже известный нам механизм хранения пространственных данных Oracle Spatial/Oracle Locator в версии 10g пополнился специальными средствами для хранения и анализа топологии пространствен-

ных данных, а также возможностью хранить "тяжелые" растровые файлы с использованием щадящего для запрашиваемых вычислительных ресурсов принципа "пирамидальной разрешающей способности". Это означает, что для просмотра фрагмента растрового изображения совершенно не обязательно загружать в память это изображение целиком, чтобы потом указать необходимую область. При общем просмотре в память загружается только "внешний срез", а полностью растр загружается только в виртуальной окрестности запрошенной области.

Итак, ориентируемся на Oracle: развитие этой концепции прямо приводит к использованию механизма "ролей", суть которого заключается в определении неких базовых групп пользователей, которым присваиваются общие для ролевой группы права на доступ к хранимым в СУБД пространственным и описательным данным.

Издержки на администрирование сразу уменьшаются за счет исключения повторного описания идентичных по уровню доступа пользователей, входящих в одну группу. При входе в систему по имени и паролю доступность тех или иных информационных ресурсов определяется только "по членству", то есть принадлежности пользователя к той или иной ролевой группе. Таким образом перевод пользователя из группы в

группу, изменение прав всей группы и прочие понятные манипуляции системный администратор производит привычными стандартными средствами администрирования СУБД, а результат регламентирования доступа к пространственным данным очевиден сразу по окончании административных действий.

Это, повторим, несколько снижает издержки администрирования, но не решает проблемы полностью: просто вместо сотни стандартных наборов данных потребуются заготовить, например, пятьдесят... всё равно многовато. Ведь для получения доступа к пространственным данным через Autodesk MapGuide придется собрать для каждой роли собственный MWF-файл, чтобы определить номенклатуру дозволенных к просмотру информационных слоев и стили их отображения.

Такое развитие событий нас никак не устраивало, и выход был найден в написании специальной утилиты с использованием Autodesk Dynamical Authoring Toolkit, которая в реальном масштабе времени "собирает" персональную карту для каждого зарегистрированного в ролевых группах пользователя!

## Проблема выбора

Теперь, когда определен магистральный путь построения системы динамического распределения пользовательских прав, необходимо решить еще один важнейший вопрос: как построить клиентское место, на котором и будут визуализироваться динамически создаваемые карты и окна данных.

Принципиально возможны два наиболее общих подхода к проблеме.

Так называемый "толстый" клиент предполагает установку на сервере Oracle Server и Web Server (Microsoft Internet Server — IIS), а на каждом рабочем месте — Oracle Client и специально разработанного пользовательского приложения, созданного на любом из стандартных языков программирования (C++, Delphi). Преимуществом такого подхода является потенциально очень высокая степень учета пользовательских запросов, так как приложение разрабатывается индивидуально. Правда, рентабелен и обоснован этот подход только при небольшой номенклатуре клиентских рабочих

мест; в случае же разработки единой муниципальной геоинформационной системы номенклатура рабочих мест столь велика, что не может быть оценена даже ориентировочно. В подобном случае на первый план выступает то обстоятельство, что для внесения любых изменений требуется либо постоянно заказывать дополнительные работы разработчику системы, либо держать в штате высококвалифицированного программиста, который будет вносить все коррективы в исходный код программы и контролировать своевременное копирование обновленных программных модулей на все рабочие места, что делает систему весьма уязвимой.

"Тонкий" клиент — это технологии, предусматривающие наличие на рабочем месте конечного пользователя только программ просмотра Internet-страниц (например, MS Internet Explorer) и не требующие написания специального программного обеспечения. Одно из основных достоинств такого подхода — значительное упрощение доступа к данным и процесса администрирования: Internet-браузеры являются стандартным программным обеспечением, доступ к данным может осуществляться с любого рабочего места, подключенного к сети. При этом все изменения производятся на сервере.

Поскольку хранение данных (как картографических, так и атрибутивных) базируется на технологиях Oracle, желательное использование тех же самых технологий и для реализации доступа. Здесь предлагаются четыре возможных подхода, различающиеся используемыми инструментами, стоимостью, трудоемкостью реализации и сложностью поддержки.

- *Использование JavaScript как базового инструмента разработки.* Сразу оговоримся, что JavaScript — неотъемлемый инструмент воплощения любого из подходов, разница состоит в количестве функциональных особенностей, для реализации которых он может использоваться. В данном случае атрибутивные данные формируются непосредственно из HTML-страницы, что обеспечивает экономию, но ведет к очень низкому уровню безопасности системы и сильной зависимости от версии

браузера (например, Netscape и Microsoft имеют весьма различающиеся версии JavaScript).

- *Стандартный пакет для работы с HTTP-протоколом.* Этот пакет включается в стандартную поставку Oracle SQL Server и позволяет осуществлять доступ к HTTP-протоколу из PL/SQL, то есть собственно формировать web-страницы с данными внутри SQL Server. На первых этапах внедрения такой подход кажется самым недорогим, поскольку он не предполагает затрат на приобретение дополнительного серверного программного обеспечения. При этом довольно значительные средства придется вложить в разработку, отладку и поддержку прикладного серверного обеспечения: данная реализация является "низкоуровневой" и не имеет средств интерактивного дизайна (если они не были разработаны специально).

- *Использование Oracle Application Server,* являющегося мощной основой для функционирования и разработки Internet-приложений. Единственным недостатком этого пути является его достаточно высокая стоимость. Среди значительных преимуществ следует отметить наличие средств интерактивной разработки приложений и простота их администрирования. В отличие от предыдущего подхода, от администратора не требуется глубоких и детальных знаний в области внутреннего функционирования системы. Многие незначительные изменения могут быть сделаны "на лету" благодаря системе разработки, базирующейся на механизме "мастеров", которые тоже выполняются как "тонкий" клиент — без установки и настройки дополнительного программного обеспечения. "Обычные" пользователи, не владеющие специальными знаниями (например, в сфере HTML), имеют возможность публиковать собственные документы. Система настройки позволяет как пользователям, так и администраторам персонализировать многие параметры и сохранять их как параметры данного пользователя, данного клиентского компьютера и т.п. Весь интерфейс системы, тексты в диалоговых окнах, сооб-

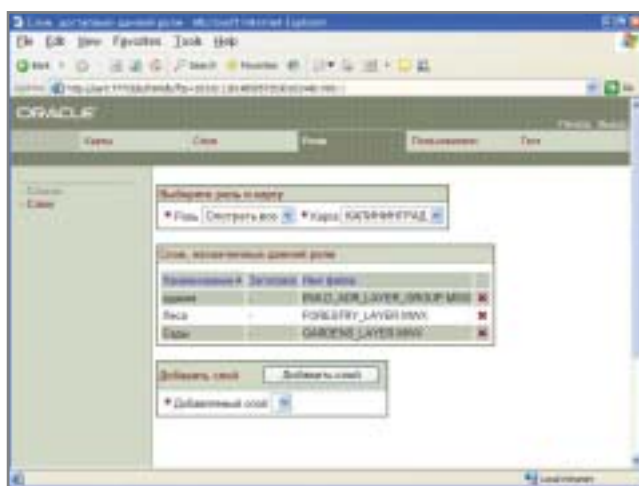


Рис. 1

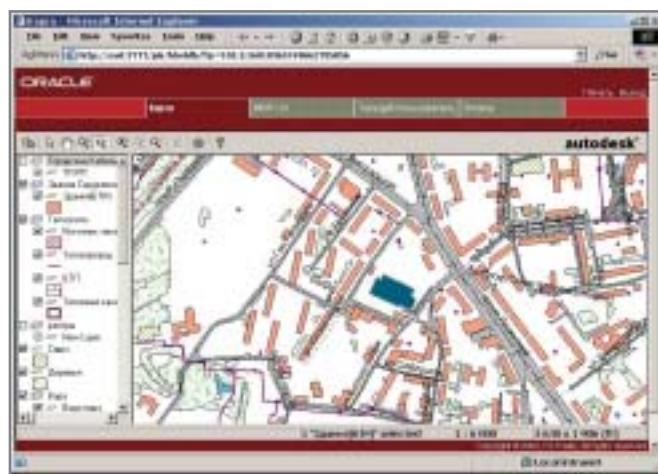


Рис. 2

щения и справочная документация являются многоязыковыми и поддерживают русский язык. В общем случае пользователь может установить язык интерфейса по своему выбору.

- *Active Server Pages (ASP) и MS .NET технологии.* Могут использоваться и как основной инструмент, и как дополнение к первому или второму подходу. Основное преимущество — относительно небольшая стоимость: не требуется покупать дорогостоящие решения от Oracle. ASP обеспечивают большую гибкость в создании пользовательского интерфейса и, в отличие от всех остальных, упрощают интеграцию с Autodesk MapGuide Server. С другой стороны, серьезным минусом этого подхода является необходимость постоянных высоких затрат на разработку и поддержку (как и при реализации второго подхода). Значительным недостатком ASP-сервера по сравнению с Oracle Application Server является установка только на одну из серверных версий операционной системы Windows с обязательной установкой Microsoft Internet Information Server 4.0 (IIS), Microsoft Personal Web Server (PWS) или WebSite Professional 2.0 фирмы O'Reilly & Associates. В то же время, исходя из соображений обеспечения безопасности информации, содержащейся в хранилище на основе СУБД, в ряде случаев могут быть использованы различные версии операционной системы UNIX, вплоть до ее наиболее распространенных бесплатных модификаций

Linux, использование наиболее защищенного Web Server Apache в его стандартной реализации или в реализации Oracle. Еще один серьезный недостаток технологии ASP — это последовательная обработка при одновременном обращении к ASP-серверу нескольких клиентов. Если один из клиентов обращается с длительным запросом, то остальные, даже в случае простых и нересурсоемких запросов, вынуждены ждать, а это создает впечатление зависания браузера. Теоретически возможно реализовать многопоточковый режим, при котором ASP-сервер обслуживает одновременно несколько клиентов, однако этот режим оказывается еще более ресурсоемким и системные переменные каждого класса инициализируются на каждое обращение, что значительно удлиняет время отклика. Добавим, что для создания программных модулей на ASP необходимо не только иметь навыки работы с VBScript, но и быть детально знакомым с принципами объектно-ориентированного программирования и с характеристиками конкретных web-протоколов.

При реализации же проекта на технологии Oracle Application Server необходимо знать только спецификации языка запросов SQL. Такое требование вполне реально для персонала муниципалитета, на который будет возложена ответственность за поддержку функционирования системы и внесение необходимых изменений.

Исходя из всего сказанного, предпочтительной технологией при

построении муниципальной геоинформационной системы на основе единого хранилища пространственных и описательных данных в СУБД Oracle является технология Oracle Application Server. При этом используется "тонкий" клиент, на рабочих местах пользователя находится только стандартный браузер для просмотра HTML-страницы, генерация которой происходит на сервере по запросу пользователя.

На сервере при этом устанавливаются Oracle Server, Oracle ApacheWebServer, Oracle Application Server, причем с целью минимизации затрат на приобретение базового программного обеспечения возможно использовать не весь комплект Oracle Application Server, а лишь один из его компонентов, называемый Oracle HTML DB.

### Что же в результате?

Итак, мы обретаем возможность обслуживать неограниченное количество пользователей, каждый из которых в динамике получает заранее определяемый его "ролью" набор слоев с возможностью наложения пространственных фильтров (не просто набор строений из слоя Buildings, а только их подмножество, принадлежащее к Центральному району).

Каждый из таких пользователей имеет еще и уникальное, определяемое той же "ролью" окно данных, в котором и состав данных, и доступные инструменты их анализа также уникальны для "роли".

Администрирование всей системы производится стандартными средствами СУБД Oracle и Oracle HTML DB.



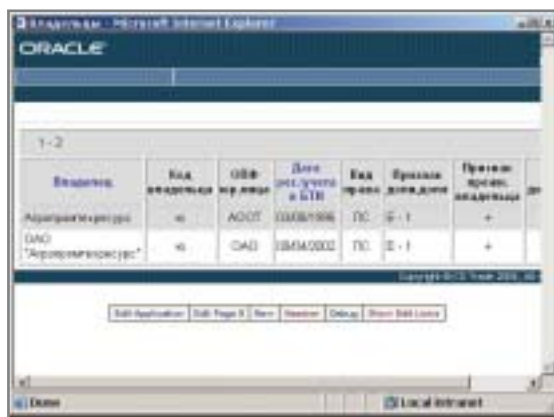


Рис. 3

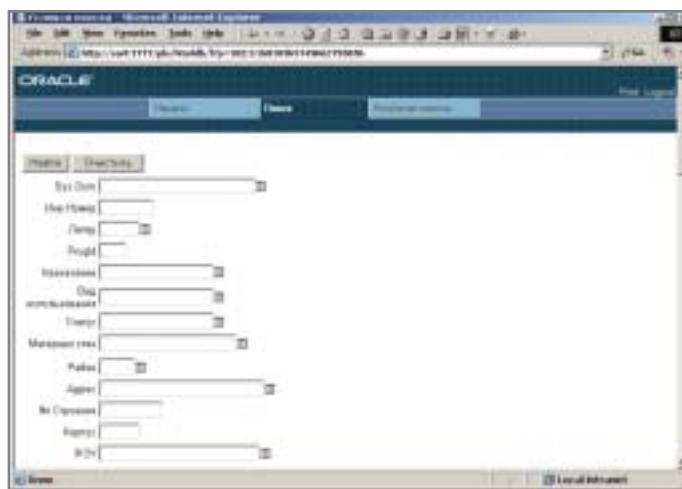


Рис. 4

Выглядит это следующим образом (на примере организации динамической карты, связанной с данными по строениям, предоставляемым городским БТИ). Администратор распределяет права и "роли" (рис. 1), а пользователь видит динамически сформированную карту (рис. 2) и динамически сформированное окно данных (рис. 3) с возможностью выполнения запросов (рис. 4).

### Краткое резюме

Описанный подход является естественным развитием концепции построения ГИС на основе СУБД Oracle. Он обеспечивает гарантированное развитие системы, независимое не только от количества одновременно работающих пользователей и от объема обрабатываемых данных, но и от лавинообразного роста различных по

функциональности рабочих мест. Все представленные решения уже функционируют в режиме опытной эксплуатации в муниципалитете Калининграда.

**Александр Ставицкий**  
к.т.н.,

директор по ГИС-направлению  
компании CSofT  
E-mail: asta@cssoft.com

## Комплексная автоматизация инженерного документооборота

**CSofT**  
Consistent Software

Москва, 121351,  
Молодогвардейская ул., д. 46, корп. 2  
Тел.: (095) 913-2222, факс: (095) 913-2221  
Internet: www.cssoft.ru E-mail: sales@cssoft.ru

Санкт-Петербург (812) 496-6929  
Воронеж (0732) 39-3050  
Екатеринбург (343) 215-9058  
Калининград (0112) 93-2000  
Краснодар (861) 254-2156  
Нижний Новгород (8312) 30-9025

Омск (3812) 51-0925  
Пермь (3422) 34-7585  
Тюмень (3452) 25-2397  
Хабаровск (4212) 41-1338  
Челябинск (351) 265-3704  
Ярославль (0852) 73-1756



### ИНЖЕНЕРНЫЕ МАШИНЫ И ПЛОТТЕРЫ Océ

Компания CSofT предлагает комплексные решения для автоматизации инженерного документооборота на базе системы управления техническими документами TDMS (www.tdms.ru), комплексов Océ (www.oce.ru), сканеров Contex (www.contex.ru), систем хранения данных, программных средств для эффективной работы со сканированными чертежами Raster Arts (www.rasterarts.ru).

Аппаратно-программные комплексы Océ (системы TDS300, TDS400, TDS600, TDS800 Pro) являются неотъемлемой частью современного технического документооборота. Компания Océ Technologies предлагает оборудование для печати (LED-плоттеры), сканирования и тиражирования широкоформатной документации, работающее автономно и в составе модульных репрографических систем. Производительность – от 2 до 10 листов формата A0 в минуту. Технологии Océ обеспечивают высокое качество и низкую стоимость копии, системы просты в обслуживании, нетребовательны к эксплуатационному помещению и расходным материалам.