

## Распределенные источники данных для создания систем поддержки принятия решений (OLAP-систем)

В.В. Городилов, А.А. Залеская\*

**Аннотация.** В данной статье выделены следующие тенденции развития СППР, как распределенных OLAP-систем:

- осознание Интернет-источников и Интернета в целом как распределенного Хранилища данных для реализации СППР;
- роль XML (Extensible Markup Language), как универсального формата данных (и метаданных) для построения распределенных хранилищ и передачи данных в СППР;
- появление спецификаций, поддерживающих создание, хранение, доступ и обслуживание данных и метаданных в OLAP-серверах (XML for Analysis, JOLAP).

### Введение

OLAP-технологии зарекомендовали себя, как эффективный способ анализа информации и довольно широко применяются при построении систем поддержки принятия решений (СППР). В качестве источника данных для OLAP систем, как правило, используются хранилища данных. Наиболее существенными особенностями хранилищ данных является консолидация в них данных из различных источников (на основе метаданных), их неизменность, наличие избыточности.

Однако развитие технологий и GRID-технологий в особенности вносит коррективы в сложившиеся представления о методах построения хранилищ данных и OLAP-систем. Интернет становится мощным источником данных, пригодных для OLAP анализа. На данный момент подавляющее большинство интернет-источников построены на базе языка HTML, описывающего форматирование данных, но не содержащего информации об этих данных. Соответственно, неструктурированные данные с Web-сайтов очень трудно перенести в хранилище данных. Но использование в представлении интернет-информации языка XML (Extensible Markup Language) обеспечивает решение данной проблемы путем присвоения информационных тэгов данным Web-сайта. Добавляя контекст к содержанию Web-сайта, язык XML дает возможность предприятиям помещать данные Web-сайта в хранилища данных, а так же непосредственно использовать их в процессе обработки аналитических запросов в GRID-среде. XML, как промежуточный формат данных, безусловно, вносит единообразие и удобство погружения и извлечения данных из хранилища данных. Интеграция технологий хранилищ данных и XML – это новый виток развития

\*Институт дискретной математики и информатики СО РАН, Новосибирск.

OLAP систем базирующихся на хранилищах данных и позволяющих интегрировать все многообразие систем этого класса, а также применять новые, до этого неиспользуемые источники информации.

### **Спецификации, поддерживающие создание, хранение, доступ и обслуживание данных и метаданных в OLAP-серверах**

XML for Analysis представляет собой API, основанный на SOAP (Simple Object Access Protocol), предназначенный для стандартизации доступа клиентских приложений к OLAP-данным через Интернет. Спецификация XML for Analysis разработана компаниями Hyperion Solutions Corporation и Microsoft Corporation. Сегодня данный стандарт поддерживают производители OLAP-средств такие как Brio Technology, Business Objects, Cognos, Comshare, Crystal Decisions, MicroStrategy, SAP. При традиционном доступе к данным с помощью универсальных механизмов доступа к данным OLE DB или ODBC на клиентский компьютер требуется установить специальные клиентские компоненты, связанные с сервером, поставляющим данные (например, библиотеки MDAC). Это накладывает определенные ограничения на клиентскую и серверную платформы, операционные системы, языки программирования. Отметим, что эти ограничения не позволяют создавать надежные многоплатформенные приложения, доступные для любых, в том числе мобильных, устройств, не требующие загрузки дополнительных компонентов на клиентское устройство. XML for Analysis развивает концепцию OLE DB путем предоставления универсального механизма доступа к аналитическим данным (OLAP-кубам и моделям Data Mining), доступным через Интернет, без необходимости установки клиентских компонентов, предоставляющих соответствующие COM-интерфейсы. В дополнение к преимуществам OLE DB, XML for Analysis оптимизирован для применения в Web-число обращений к серверу в случае применения средств, поддерживающих эту спецификацию, минимизировано, а клиентское приложение является не хранящим состояния (stateless), что позволяет создавать масштабируемые распределенные аналитические приложения. XML for Analysis можно применять не только для реализации Web-запросов, но и для осуществления взаимодействия между серверами. XML for Analysis имеет может стать не зависящим от производителей OLAP-средств универсальным механизмом запросов к OLAP-данным, но не заменяет полностью исходные API для доступа к многомерным данным, специфические для конкретных продуктов, так как уступает им в производительности. Тем не менее такой API позволит создавать универсальные клиентские OLAP-средства, поддерживающие одновременно несколько серверных OLAP-продуктов, и роль XML for Analysis для аналитических приложений будет так же важна, как и роль ODBC в мире реляционных СУБД.

О разработке еще одной спецификации – JOLAP (Java OLAP Interface) было объявлено в августе 2000 г. рядом лидеров рынка OLAP-средств, включая компании Hyperion Solutions, IBM, Nokia, Oracle, SAS Institute, Unisys. JOLAP представляет собой Java API для J2EE (Java 2 Enterprise Edition). В отличие от OLE DB for OLAP, XML for Analysis и MDAPI, спецификация JOLAP пре-

тендует на поддержку не только API для запросов к готовым OLAP-кубам, но и создания и модификации многомерных баз данных и содержащихся в них объектов. Спецификация JOLAP отвечает требованиям Pure Java API, поддерживающего создание, хранение, доступ и обслуживание данных и метаданных в OLAP-серверах и многомерных базах данных. Платформой для реализации JOLAP является Java 2 Platform, Enterprise Edition (J2EE).

Использование JOLAP позволяет конструкторам OLAP-систем предоставить единый стандартный API, который воспринимается множеством различных клиентских приложений и компонент, работающих на платформе J2EE. Аналогично, OLAP-клиенты могут быть запрограммированы на единый API, независимый от лежащего в его основе источника данных (например, “родная” многомерная база против реляционной звездной схемы). Конечной целью JOLAP является предоставление OLAP-системам тех же возможностей, которыми JDBC оснастил реляционные базы данных. JOLAP разрабатывается на базе высоко-обобщенной объектно-ориентированной концептуальной OLAP-модели. Эта модель поддерживает три концептуальных области, представляющих основной интерес для пользователей OLAP-систем: Метаданные, Данные и Запрос. Метаданные и Данные обеспечивают интерфейсы, поддерживающие манипуляции OLAP-данными и метаданными соответственно. Запрос определяет интерфейсы, поддерживающие общие OLAP-запросы (к метаданным и данным), а также управление и манипуляцию результирующими множествами. Объектная модель обеспечивает сервисы и интерфейсы на уровне ядра, доступные всем клиентам. Клиенты последовательно видят одни и те же интерфейсы и семантику и программируются для этих интерфейсов. Отдельная реализация объектной модели не обязательно поддерживает все интерфейсы и сервисы, определенные с помощью JOLAP. Спецификация JOLAP не предписывает никакой специальной стратегии реализации. Поставщики могут решать, встраивать ли JOLAP в свои продукты в качестве “родного” API или предпочесть разработку драйвера/адаптера, служащего связующим звеном между центральным JOLAP-уровнем и различными продуктами поставщиков. Чтобы гарантировать совместимость с J2EE и исключить дублирование работы, JOLAP использует существующие спецификации. В частности, JOLAP полагается на Java Connection Architecture в части обеспечения управления ресурсами, транзакциями и результирующими множествами, маршрутирования записей и безопасности. Кроме того, JOLAP использует создающийся Java Metadata Interface для центрального управления данными (т.е. JOLAP расширяет возможности центральных JMI-интерфейсов для представления понятий OLAP-метаданных, например, измерения и иерархии). JOLAP не имеет специальной операционной системы и не зависит от аппаратного обеспечения. Механизм безопасности в JOLAP будет организован за счет использования возможностей таких механизмов в J2EE и лежащих в основе JOLAP OLAP-системах.

## Заключение

Мы рассмотрели технологические аспекты связанные с построением распределенных OLAP-систем, позволяющих использовать Интернет-источники как распределенное хранилище данных, а XML в качестве связующей технологии

в процессе его построения. Однако следует отметить, что при этом заметно меняется и взгляд на ряд классических принципов построения OLAP-систем.

Так принцип неизменяемости данных в хранилище данных, сформулированный еще Биллом Инмоном, при извлечении данных для анализа из распределенных Интернет-источников вряд ли удастся сохранить в полном объеме. В то же время, все базовые признаки хранилища данных при построении его на распределенных источниках вполне могут быть реализованы:

- предметная ориентация,
- предварительная очистка и упорядочение данных,
- привязка ко времени.