

УДК 004.658.6

И.В.Бутенко (6 курс, каф. ИУС), Д.Ф.Дробинцев, ст. преп.

ПРИНЦИПЫ ПОСТРОЕНИЯ МЕТАДАНЫХ ПРИ РАЗРАБОТКЕ АНАЛИТИЧЕСКОЙ СИСТЕМЫ

Наличие метаданных и средств их представления конечным пользователям является одним из основополагающих факторов успешной реализации аналитической системы (АС). Более того, без наличия актуальных, максимально полных и легко понимаемых пользователем описаний данных, Хранилище Данных (ХД), информация из которого берется для работы АС, превращается в обычный, но очень дорогостоящий электронный архив.

Первой же задачей, с которой сталкиваются при проектировании и реализации АС, является необходимость одновременной работы с самыми разнородными внешними источниками данных, несогласованными по структуре и формату, масштабам и количеству архивов, которые должны быть переработаны и загружены. При построении АС, разработчику сложно обойтись без высокоуровневых средств описания информационной модели системы. Причем, эта модель должна содержать описания не только целевых структур данных в базе данных ХД, но и структур данных в источниках их получения (различных информационных системах, архивах, электронных справочниках и т.д.), правила, процедуры и периодичность их выборки и выгрузки, процедуры и места согласования и агрегации.

Для реализации подобных задач нам необходимы данные, обеспечивающие отображение предметной базы конечных пользователей и предметной базы ХД. Эти данные называются метаданные, данные о данных. Обобщая, можно сказать, что метаданные – это «карта расположения данных» в хранилище.

В работе рассмотрен принцип построения метаданных, который базируется на многоуровневом описании, позволяющем локализовать понятийные базы конечных пользователей и разработчиков и обеспечить их взаимное отображение.

Сформулируем, что именно “хочет” знать конечный пользователь, кроме описания структур данных, их взаимосвязей и предвычисленных уровней агрегации.

- Источники получения данных. Аналитику желательно не просто знать о том, какие данные есть в системе, но и источники их получения и степень их достоверности. Например, одна и та же информация может попасть в Хранилище Данных из различных источников. В этом случае, пользователь должен иметь возможность узнать, какой источник выбран в качестве основного и каким образом выполнялась согласование и очистка исходных данных.

- Периодичность обновления. Пользователю желательно не просто знать, какому моменту времени соответствуют те или иные данные, но и когда они будут обновлены.

- Собственников данных. В отличие от традиционных СОД, где пользователь видит только то, что ему разрешено, здесь пользователю будет полезно знать, какие еще данные есть в системе, кто является их собственником и какие шаги он должен предпринять, чтобы получить к ним доступ.

- Статистические оценки запросов. Еще до выполнения запроса пользователю желательно иметь хотя бы приблизительную оценку времени, которое потребуется для получения ответа, и представлять, каков будет объем этого ответа.

Пользователь аналитической системы должен иметь возможность видеть, из каких именно внешних источников была загружена информация в ХД. Можно изобразить этот вид взаимодействия следующим образом:

Пользователь ↔ ХД ↔ Источники данных

Для того чтобы перевести требования, предъявляемые пользователем к АС, на язык, понятный разработчику этой системы, и чтобы реализованные разработчиками механизмы были понятны пользователю в терминах его предметной области, предлагается следующая трехуровневая архитектура метаданных.

- Уровень приложения (внешних источников данных).

Описывает структуру данных в операционных БД и других источниках данных. Обычно этот уровень достаточно сложен для понимания неподготовленного пользователя и является приложение-ориентированным. Этот уровень будет использоваться разработчиками ХД.

- Уровень ядра Хранилища Данных

Описывает логическую и физическую структуру и взаимосвязи данных в ХД.

- Уровень конечного пользователя

Описывает структуры данных в ХД в терминах предметной области конечного пользователя.

Предложенное многоуровневое описание метаданных в настоящее время используется при разработке АС для коммерческих банков.