
АРХИТЕКТУРА И РЕАЛИЗАЦИЯ СРЕДСТВ РЕПОРТИНГА В ДИНАМИЧЕСКИ НАСТРАИВАЕМЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ

Вячеслав Ланин

Аннотация: Статья посвящена описанию архитектуры и реализации подсистемы работы с запросами и отчетами в адаптируемых динамически расширяемых информационных системах. Разработанный метод характеризуется универсальностью применения, ориентацией на пользователя и возможностью интеграции с внешними информационными системами. Реализация программных средств основана на использовании многоуровневых метаданных.

Keywords: CASE-технология, адаптируемая информационная система, электронный документ, построитель запросов, генерация отчетов.

ACM Classification Keywords: D.2 Software Engineering: D.2.2 Design Tools and Techniques – Computer-aided software engineering (CASE); H.2: Database Management:: H.2.3 Languages – Report writers; H.3.3 Information Search and Retrieval – Query formulation.

Введение

В настоящее время под термином «business intelligence» понимаются инструментальные средства для анализа данных, построения отчетов и запросов, помогающие бизнес-пользователям обрабатывать большие объемы данных для того, чтобы извлечь из них и синтезировать значимую информацию. Получение результатов обработки данных, хранящихся в базе данных (БД), хранилище данных – одно из основных назначений любой программной системы. Данные вводятся пользователями и накапливаются для последующей их обработки (системы OLTP); выполнения анализа, прогнозирования и поддержки принятия обоснованных управленческих решений (OLAP и DSS) и т.п. Результаты, получаемые на основе данных, размещенных в БД, должны быть не только определенным образом обработаны, но и визуализированы, представлены в виде документов различных форматов в зависимости от текущих информационных потребностей пользователей.

Таким образом, при построении любой информационной системы возникают задачи реализации средств создания запросов, подготовки и генерации отчетов, по возможности, не требующих навыков программирования, доступных пользователям, умеющим работать в среде современных офисных продуктов. Средства «business intelligence» являются компонентом практически каждой информационной системы, эффективность их реализации во многом определяет эффективность всей системы.

Требования к подсистеме управления документами в CASE-системе METAS

Разработка информационных систем (ИС), допускающих возможность динамической настройки на меняющиеся условия и потребности пользователей, расширения функциональности в ходе эксплуатации системы предусматривает необходимость использования специальных инструментальных средств разработки.

CASE-технология METAS (METAdata System) – это основа для создания адаптируемых информационных систем, управляемых метаданными [1]. Данная технология предназначена для снижения трудоемкости разработки корпоративных информационных систем и повышения их гибкости, масштабируемости и адаптируемости непосредственно в процессе эксплуатации. Основное отличие рассматриваемой системы от многих существующих CASE-систем, генерирующих по некоторым спецификациям, описывающим

предметную область, код на каком-либо языке программирования, состоит в том, что METAS использует это описание *во время* своей работы, выполняя функции и отображая данные, определенные метаданными. Это дает возможность гибкой настройки приложения, реструктуризации описываемых объектов в процессе эксплуатации системы и создает хорошие предпосылки для создания интеллектуальной системы, которая может настраиваться на потребности пользователя. Технология METAS ориентированна на создание открытых распределенных приложений. Подсистема репортинга должна учитывать эти особенности технологии [2].

В подсистеме репортинга необходима реализация следующих функций:

- создание пользователем запросов к БД в терминах предметной области;
- генерация отчетных документов на основе разработанных пользователем шаблонов;
- передача шаблонов запросов и отчетов между узлами распределенной системы;
- интеграция с внешними системами.

Рассмотрим перечисленные выше требования более подробно с учетом специфики технологии METAS и предполагаемого характера применения ИС, построенных на основе данной технологии.

Основным требованием к системе репортинга является обеспечение интерфейса пользователя, облегчающего пользователям-непрограммистам *подготовку отчетов и запросов в привычных для пользователя терминах предметной области*. Отчеты могут быть достаточно сложными, включать элементы дополнительной обработки, анализа и визуализации сложно организованных данных. Система должна предусматривать *возможность расширения*, то есть создания новых отчетных форм и запросов. При этом от пользователя не должно требоваться владение специальными навыками программирования или знание языка SQL. Пользователь при построении отчета *должен работать в знакомой ему среде*. Из перечисленных выше требований следует что, необходимым условием реализации данного компонента является использование *развитой системы метаданных для описания предметной области*. Эти метаданные в системах, построенных на технологии METAS, уже имеются: с помощью них описывается модель предметной области ИС. Таким образом, и предметная область, и создаваемые пользователями запросы и отчеты могут описываться одними и теми же терминами.

Система должна обеспечивать хранение построенных запросов и шаблонов отчетов с возможностью их *повторного применения* и использования в качестве источника данных результатов выполнения ранее созданных запросов. Данное требование приводит к необходимости включения в структуру метаданных METAS *модели репортинга*, описывающей запросы и отчеты. В распределенной ИС необходимы развитые средства обмена данными между различными подсистемами, то есть построенные запросы и шаблоны отчетов, метаданные, описывающие их, должны быть легко *переносимы*.

Построение запросов требует *формирования сложных условий и выражений*. Как правило, для этой цели применяется так называемый «редактор выражений» – компонент, позволяющий пользователю строить выражения из некоторых базовых конструкторов, обеспечивающий проверку правильности созданного выражения. Для увеличения мощности редактора выражений необходима реализация *встроенного языка* для создания формул.

Пользователю должны быть предоставлены *инструменты позиционирования элементов отчета* при разработке шаблона. Требования к точности позиционирования элементов отчета бывают достаточно высокими, особенно когда данный отчет служит целям внешней отчетности, допускает возможность автоматизированной обработки.

Для просмотра и редактирования документов непосредственно из системы должны быть доступны средства *интеграции с внешними программными продуктами*. Особенно необходима тесная интеграция с офисными программами, системами электронной почты, являющимися в организациях одним из основных средств передачи документов.

Еще одно требование – возможность настройки на *использование различных СУБД*.

Неотъемлемой частью предлагаемого решения является наличие *единого механизма обработки документов из различных источников, автоматизация процессов обмена данными с различными внешними информационными системами, обеспечение возможности импорта текстов и документов из файлов и баз данных разнообразных форматов*. Необходимо предусмотреть возможность применения средств OLAP и Data Mining для анализа данных.

Подходы к созданию отчетов в информационных системах

Все средства создания (генераторы) отчетов можно условно разделить на три категории:

1. *Встроенные средства* – генераторы отчетов, встроенные в средства разработки, электронные таблицы и настольные СУБД.
2. *Специализированные средства* – генераторы отчетов, выпущенные в виде отдельных приложений (как правило, они достаточно универсальны, предназначены для работы с различными БД и категориями пользователей).
3. *«Нетрадиционные» средства* – приложения, которые, с одной стороны, имеют высококачественные средства управления печатью документов или конвертирования их в различные форматы, а с другой стороны, являются серверами автоматизации, предоставляющими доступ к этим возможностям с помощью своих объектных моделей.

Рассмотрим каждую из категорий более подробно и приведем примеры программных продуктов, являющихся наиболее типичными представителями данных категорий.

Инструменты класса *встроенных средств* (Rave Reports, SQL Server 2005 Reporting Services, ReportBuilder) обычно обладают меньшими возможностями, нежели средства создания отчетов других классов. Встроенные средства создания отчетов предоставляют достаточно богатые возможности по представлению информации, но от пользователя в большинстве случаев требуется знание языка SQL для построения запросов, пользователь также должен разбираться в структуре таблиц БД, к которой строится запрос. Все это накладывает достаточно жесткие требования на уровень квалификации пользователя и ограничивает сферу применимости данных средств. Безусловно, мы можем заранее заложить в систему шаблоны стандартных отчетов, созданные с помощью встроенных средств, можно также предусмотреть определенные возможности по настройке данных шаблонов. Однако вряд ли таким образом можно предусмотреть все потребности, которые могут возникать у пользователей в ходе эксплуатации ИС.

Кроме встроенных средств на рынке программного обеспечения представлено несколько продуктов, относящихся к *специализированным средствам* создания отчетов для разных категорий пользователей (специалистов, занимающихся подготовкой отчетов, рядовых пользователей, разработчиков приложений, разработчиков Web-сайтов и др.). Нередко один и тот же продукт существует в нескольких редакциях, ориентированных на различные категории пользователей. Специализированные средства создания отчетов обычно характеризуются поддержкой различных механизмов доступа к данным и наличием мастеров и визуальных инструментов, ориентированных в первую очередь на пользователей-непрограммистов. Также следует отметить наличие встроенных средств деловой графики, интеграции с различными офисными приложениями и поддержки публикации данных в Internet. Для опытных пользователей и программистов предоставляются средства интеграции с наиболее популярными средствами разработки, встроенные языки для создания формул, возможности создания сложных аналитических отчетов. Безусловными лидерами на рынке специализированных средств создания отчетов являются продукт Seagate Crystal Reports фирмы Seagate Software (Crystal Decision) и комплекс продуктов фирмы BusinessObjects.

Нередко в качестве средств создания отчетов применяются приложения, которые имеют высококачественные средства управления печатью документов или конвертирования их в различные форматы. Наиболее часто в качестве таких генераторов отчетов используются приложения Microsoft Office или другие подобные продукты. В этом случае, как правило, при создании отчета возможность

спроектировать его макет с помощью визуальных средств отсутствует, поэтому чаще всего такой способ генерации печатных документов используется не рядовыми пользователями, а программистами, создающими решения на базе Microsoft Office. Приложения, являющиеся частью таких решений и «заставляющие» сервер автоматизации выполнять те или иные действия (в данном случае создавать или печатать документ требуемого формата, содержащий запрашиваемые данные), называются контроллерами автоматизации. К преимуществам создания отчетов с помощью Microsoft Office относится определенная гибкость при выборе механизма доступа к данным: можно применять как механизмы доступа к данным самого Office, так и механизмы доступа к данным, которые поддерживаются средством разработки, применяемым при создании такого приложения. Немаловажным преимуществом использования средств MS Office для генерации отчетов является также его распространенность и популярность в широких кругах пользователей.

Применение встроенных средств создания отчетов оправдано, в случае разработки системы с ограниченным количеством встроенных фиксированных отчетов. Однако эти средства требуют навыков программирования при необходимости создания новых отчетов или изменения шаблонов существующих. Это делает неприемлемым их применение в разрабатываемой системе.

Использование внешнего генератора отчетов, например BusinessObjects (BO), позволяет выполнить большинство требований. BusinessObjects имеет возможность настройки на предметную область за счет средств семантической прослойки (Юниверса), обеспечивая за счет этого работу пользователя в терминах предметной области. Основным недостатком данного продукта является его высокая стоимость, делающая невозможным его массовое применение. Кроме того, интерфейс BO достаточно сложен для понимания неподготовленного пользователя. При применении данного средства необходимо ведение двух систем метаданных: метаданных самой ИС и метаданных семантической прослойки BO, как следствие, необходимо также отслеживать непротиворечивость метаданных двух систем. Вышеперечисленное делает невозможным применение BusinessObjects в качестве средства репортинга разрабатываемой системы.

Другой известный продукт – Crystal Reports – не удовлетворяет требованиям, предъявляемым к подсистеме репортинга ИС, т.к. не обеспечивает возможность работы пользователя в терминах предметной области. В этом продукте имеется так называемый «Словарь» (Dictionary), однако он лишь позволяет задать псевдонимы для полей и таблиц БД, что недостаточно в нашем случае.

Таким образом, для реализации поставленной задачи не подходят решения, представленные в первых двух категориях. Необходимо решение из третьей категории, включающие в себя преимущества первых двух. Разработка собственного генератора отчетов – наиболее гибкий вариант в отношении удовлетворения требований. При таком подходе имеется возможность тесной интеграции с приложениями пакета Microsoft Office и остальными компонентами системы.

Данное решение обладает следующими преимуществами:

- Поскольку разрабатывается собственное средство репортинга, есть возможность *учесть все требования, предъявляемые к подсистеме репортинга*. В частности, мы можем использовать метаданные системы для обеспечения возможности работы в терминах предметной области.
- Использование Microsoft Word и Microsoft Excel в качестве графической оболочки для разработки макета отчета *позволяет исключить затраты на разработку собственного средства представления документов*. Офисные продукты Microsoft имеют развитые средства оформления документов и точного позиционирования элементов, возможность применения средств анализа, деловой графики и т.п.
- Документы данного типа являются стандартом де-факто электронного документа в России. Большинство пользователей знакомы с данной программой, что позволит *снизить затраты на обучение пользователей и обеспечит их комфортную работу*. Microsoft Office инсталлирован на

большинстве пользовательских компьютеров, следовательно не будет необходимости закупки данного продукта.

Следует сделать одно замечание относительно создания сложных *аналитических отчетов*. Можно выделить два противоположных подхода к их реализации. Первый подход, применяющийся в большинстве существующих систем, состоит в том, что для добавления нового отчета требуется написание программного кода и создание запросов на языке SQL. Очевидно, что в этом случае ИС требует сопровождения профессиональным программистом, хорошо разбирающимся в предметной области данной системы, обработка данных таким подходе вынесена в программный код. Второй подход заключается в том, что при необходимости создания сложных отчетов сначала формируются с помощью реализованных средств простые отчеты, а затем используются программные продукты сторонних производителей, такие как Microsoft Excel или Word и т.д., которые позволяют реализовать дополнительную обработку полученных результатов (анализ данных, визуализация в виде диаграмм и пр.). Конечно, первым способом можно построить отчет любой степени сложности, но в силу того, что одна из основных концепций системы METAS – исключение сопровождения ИС программистами на всех стадиях существования системы, данный подход неприменим. Таким образом, необходима реализация подхода, с одной стороны, достаточно простого в использовании, не требующего в процессе эксплуатации навыков программирования, а с другой стороны, содержащего потенциальные механизмы создания сложных аналитических отчетов.

Реализация подсистемы репортинга CASE-системы METAS

Как уже было сказано выше, одно из основных требований к подсистемам создания запросов и отчетов – это возможность их разработки пользователями-непрограммистами. Такое требование может быть выполнено только за счет введения дополнительного семантического слоя, основой которого могут быть метаданные, уже присутствующие в системе. Это даст пользователю возможность работы с данными в соответствии с терминологией, принятой в конкретной предметной области, позволит абстрагироваться от физической нормализованной структуры таблиц. В предлагаемом подходе система репортинга состоит из двух компонентов: *построителя запросов* и *генератора отчетов*.

За основу инструмента создания запроса был взят аналогичный инструмент Microsoft Access, знакомый большинству опытных пользователей. Согласно предложенной концепции пользователь выбирает сущности, участвующие в запросе, и необходимые связи между ними. Затем включает в запрос интересующие его атрибуты сущностей и другие параметры, влияющие на сортировку и группировку данных. В результате учета требований пользователя и интерпретации метаданных построитель запросов автоматически генерирует SQL-запрос к БД ИС.

Для создания отчетов разработан специальный инструмент «Менеджер отчетов». В качестве шаблонов для отчетов могут быть использованы документы Word и рабочие книги Excel. Для обеспечения обмена с другими узлами распределенной ИС построитель имеет функции экспорта и импорта шаблонов отчетов и документов.

В соответствии с предлагаемым подходом последовательность действий пользователя для *создания нового отчета* выглядит следующим образом:

- 1) подготовка необходимых запросов с помощью «Менеджера запросов»;
- 2) подготовка шаблона офисного документа (включение в шаблон статической информации: элементов оформления, формул и диаграмм и т.д.) и его разметка (включении в шаблон информации о диапазонах, куда будут помещаться данные при генерации документа на базе данного шаблона);
- 3) связывание запросов и соответствующих диапазонов документа, в которые должны быть помещены результаты их выполнения;

4) сохранение полученного шаблона в базе метаданных системы для последующего использования.

Предложенный подход обладает рядом преимуществ. Во-первых, для создания нового отчета не требуется программирование и написание запросов на языке SQL. При необходимости аналитическая обработка информации может быть произведена в Microsoft Excel. Во-вторых, хранение шаблона отчета в базе метаданных (БМД) делает этот отчет частью метаданных. При таком подходе отчет совместно с запросами, на базе которых он получен, может тиражироваться между узлами распределенной информационной системы. При поступлении нового типа отчета в узлах информационной системы не потребуются обновления программного обеспечения, в отличие от традиционных систем.

Последовательность *генерации отчетов* является следующей:

- 1) из БМД извлекаются шаблоны отчета, на их основе создаются документы;
- 2) выполняются запросы к БД ИС, связанные с каждым из шаблонов;
- 3) происходит вставка результатов выполнения запросов в диапазоны отчета, при необходимости в нем происходят дополнительные вычисления;
- 4) созданный отчет сохраняется в БД как документ, он может быть распечатан или передан по сети.

Благодаря реализованной возможности хранения электронных документов в БД, сгенерированные отчеты становятся частью данных ИС. Анализ первичных отчетов за различные временные промежутки позволяет создавать новые – сводные – отчеты.

Описанные выше подходы к разработке средств репортинга могут быть реализованы в системе, имеющей архитектуру, показанную на рис. 1.

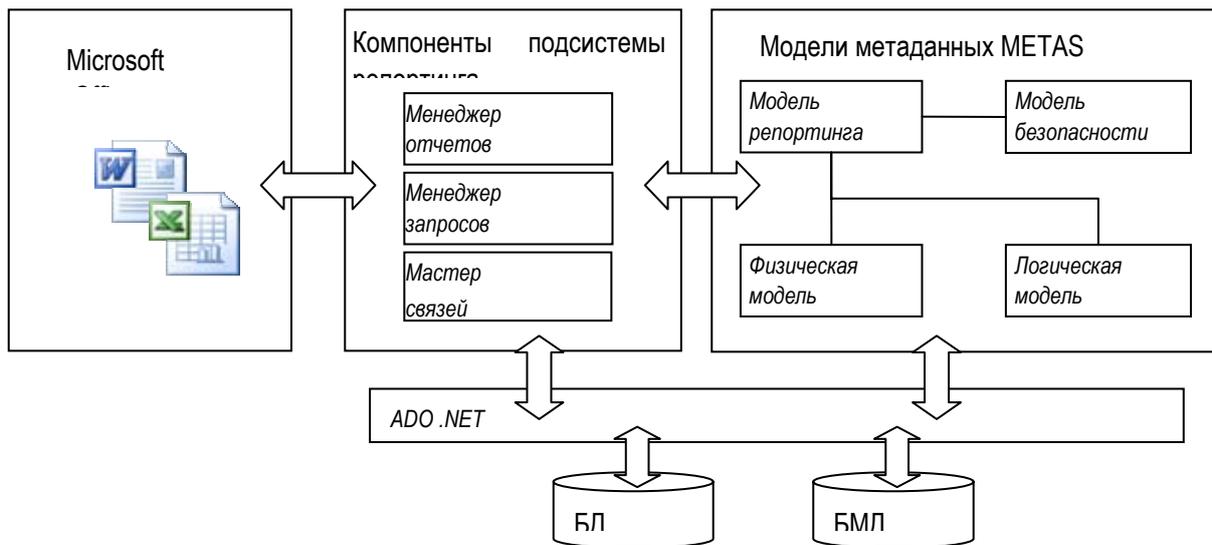


Рис.1. Архитектура подсистемы репортинга

Была разработана *модель репортинга*, которая включает в себя две подмодели: подмодель запроса и подмодель отчета. Модели тесно интегрированы в общую концепцию метаданных системы, они опираются на функциональность физической и логической моделей, а также на модель безопасности. Программный интерфейс модели репортинга служит для:

- выполнения, добавления, удаления и редактирования запросов к данным ИС, формулируемым в терминах логической модели ядра метаданных, т.е. в терминах сущностей и их атрибутов;
- управления шаблонами отчетов: генерации отчета по заданному шаблону, добавления, удаления и изменения шаблона отчета.

Основное предназначение модели запроса – хранение информации о созданных пользователем запросах к данным ИС, редактирования ранее созданных запросов, возможность экспорта и импорта запросов.

Модель отчета хранит информацию об отчетах, доступных в системе, позволяет добавлять новые, изменять и удалять уже существующие отчеты. Особенностью модели является возможность непосредственного хранения шаблона отчета в метаданных.

Менеджер запросов используется для управления пользовательскими запросами. С помощью данного компонента возможны создание, редактирование, удаление и выполнение пользовательских запросов к базе данных информационной системы. «Менеджер запросов» представляет собой интерфейс для работы с моделью отчета. Компонент «Мастер связей» предназначен для связи диапазонов шаблонов и запросов в визуальном режиме. При генерации отчета в диапазоны шаблонов отчета помещается результат выполнения связанных с ними запросов.

Алгоритм генерации запроса

Как было описано выше, составленный пользователем запрос хранится в модели репортинга БМД. Данные ИС хранятся в ее реляционной БД, поэтому возникает задача трансляции запроса в терминах модели на язык запросов SQL.

Пользователь формирует свой запрос в терминах сущностей, ограничения и выражения для полей запроса он задает с помощью выражений, в которых используются термины логической модели (сущности, атрибуты и т.д.). Так как конечным результатом процедуры генерации запроса должен быть SQL-запрос к БД ИС, очевидно, что применение терминов сущностей и атрибутов недопустимо. Таким образом, необходима трансляция терминов логической модели в термины таблиц БД. Но решение такой задачи является очень сложным: требуется мощный анализатор выражений либо необходимо ограничить пользователя в сложности выражений.

Предлагается другое решение, не ограничивающее пользователя в сложности условий и не требующее сложного программирования. Источником данных будут не таблицы БД, а предварительно приведенные к первой нормальной форме представления сущностей. Логическая модель обладает средствами генерации запросов для таких представлений. Благодаря такому подходу, мы получим одинаковое представление источников данных и налагаемых на них ограничений.

С учетом модели запроса и особенностей функционирования ядра исполнительской среды CASE-системы METAS, необходимо привести запрос к следующему виду:

```
SELECT поля_запроса
FROM ( (запрос_для_Сущности1) Псевдоним_для_Сущности1
INNER|LEFT|RIGHT JOIN (запрос_для_Сущности2) Псевдоним_для_Сущности2
    ON условие_соединения_Сущности1_и_Сущности2, ...
INNER|LEFT|RIGHT JOIN (запрос_для_СущностиN) Псевдоним_для_СущностиN
    ON условие_соединения_СущностиN_и_СущностиM),
    Таблицы_Соединения_Сущностей_M:M, (...)Псевдоним_для_Сущности(N+1),...,
    (...)Псевдоним_для_Сущности(N+n),
WHERE Условия_Соединения_Сущностей_M:M, Условия_Пользователя
GROUP BY group_by_expression
ORDER BY order_expression [ ASC | DESC ]
```

Ключевыми этапами алгоритма являются следующие шаги:

1. Формирование списка полей, выводимых в качестве результата запроса.
2. Разбиение множества используемых в запросе сущностей на три подмножества: связанные с помощью внутреннего, левого или правого объединения (*JoinLinked*); связанные отношением

«многие ко многим» и не входящие в первое множество (*MultiLinked*); к последнему множеству относятся несвязанные сущности (*NotLinked*).

3. Обращение к логической модели для получения текста SQL-запроса для каждой из используемых сущностей. Логическая модель сгенерирует запрос, возвращающий таблицу экземпляров сущности в первой нормальной форме.
4. Формирование раздела запроса «FROM»:
 - Объединение запросов, являющихся результатом выполнения шага 3 алгоритма, для сущностей из множества *JoinLinked* с помощью указанного типа объединения (LEFT JOIN, RIGHT JOIN, INNER JOIN), присваивание псевдонима результату подзапроса.
 - Перечисление запросов для сущностей из множества *MultiLinked*, в ходе которого присваиваются псевдонимы результатам подзапросов и добавляется имя вспомогательной таблицы, организующей связь «многие со многими».
 - Перечисление запросов для сущностей из множества *NotLinked*, с присваиванием псевдонимов результатам подзапросов.
5. Формирование раздела запроса «WHERE»:
 - Для каждой сущности из множества *MultiLinked*, добавляются условие связи двух сущностей.
 - Добавляются ограничения, наложенные пользователем на атрибуты сущностей.
6. Формирование раздела запроса «ORDER BY» путем объединения элементов сортировки.
7. Формирование раздела запроса «GROUP BY» путем объединения элементов группировки.

Описанная последовательность действий приведет к генерации SQL-запроса в терминах таблиц и полей БД ИС. Данный запрос будет произведен непосредственно к базе данных без обращения к моделям ядра метаданных системы.

Очевидно, что сформированный запрос не будет являться оптимальным по своей структуре, в силу противоречивости предъявленных к нему требований. Но, учитывая, что все современные СУБД имеют встроенные внутренние средства оптимизации запросов, и то, что скорость отклика системы не критична, предложенное решение удовлетворяет основным требованиям.

Следует заметить, что источники оптимизации запроса касаются логической модели. Сейчас логическая модель в качестве запроса для сущности возвращает ее первую нормальную форму, в которую входят все ее атрибуты, совершенно необязательные для выполнения запроса. Таким образом, объем текста запроса и скорость его выполнения можно уменьшить за счет добавления дополнительной функциональности в логическую модель.

Заключение

Предложенный подход к построению системы репортинга CASE-системы METAS позволил создать гибкую, ориентированную на пользователя систему работы с отчетами и запросами. Применение метаданных позволяет уйти от использования SQL при формировании запроса к реляционной базе данных. Реализованная подсистема репортинга стала частью комплексного подхода к управлению электронными документами в CASE-системе METAS [3].

Библиографический список

- [1] Лядова Л.Н., Рыжков С.А. CASE-технология METAS // Математика программных систем: Межвуз. сб. науч. трудов / Перм. ун-т. Пермь, 2003. С. 4-18.
- [2] Бакланов Д.М., Варламов А.А., Ланин В.В., Лядова Л.Н. Подсистема репортинга программного комплекса MDK METAS // Математика программных систем: Межвуз. сб. научных трудов / Перм. ун-т. Пермь, 2003. С. 19-34.
- [3] Ланин В.В. Подсистема управления документами CASE-системы METAS // Математика программных систем: Межвуз. сб. науч. трудов / Перм. ун-т. Пермь, 2006. С. 135-146.

Сведения об авторе

Вячеслав Ланин – Пермский государственный университет, аспирант кафедры математического обеспечения вычислительных систем; Россия, г. Пермь, 614990, ул. Букирева, д. 15; e-mail: lanin@psu.ru