

В. В. Миронов, Н. И. Юсупова, Г. Р. Шакирова

**СИТУАЦИОННО-ОРИЕНТИРОВАННЫЕ БАЗЫ ДАННЫХ:  
ВНЕШНИЕ ПРЕДСТАВЛЕНИЯ НА ОСНОВЕ XSL**

Представлены результаты распространения теории ситуационно-ориентированных баз данных (СОБД) на область организации внешних представлений. Обсуждается концепция внешних представлений, раскрывается их понятие, приводится классификация. Рассматривается обобщенная структурная модель внешних представлений на основе шаблонов XSLT с инструкциями извлекающей и форсирующей трансформаций. Рассматривается метод интерпретации внешних представлений при обработке данных из ситуационно-ориентированной базы. *Ситуационно-ориентированные базы данных; XSL-трансформация; XML-технологии; внешние представления*

В современном обществе роль информации столь велика, что оно по праву обрело статус информационного. Постоянно совершенствуются технологии хранения и обработки данных, усложняясь как концептуально, так и технически. Традиционно сочетание «хранение и обработка информации» относится к давно устоявшемуся термину «базы данных». Под базами данных принято понимать набор данных, организованный по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования, независимый от прикладных программ.

Основное назначение баз данных – хранить информацию, предоставляя пользователям и приложениям средства для ее извлечения и модификации. В этой связи выделяются два уровня представления данных:

- внутреннее представление: организация баз данных с точки зрения системы управления ими: структура объектов хранения данных, организации их взаимодействия и пр. Для большинства баз данных это то, как «видит» информацию СУБД;

- внешнее представление: различные комбинации элементов данных на основе их внутреннего представления. Фактически это то, как «выглядят» данные за пределами системы управления ими, как их «видят» пользователи и сторонние приложения.

На сегодняшний день известны различные классификации баз данных – от практической возможности их физической организации до концептуальной модели, лежащей в их основе. Последний классификационный признак является наиболее распространенным. В соответствии с ним базы данных делятся на иерархические, сетевые, реляционные, объектные и др.

В работе [1] предложен несколько иной вид баз данных, в основе которых лежит дискретная модель развития *ситуации*. Такие базы данных получили название ситуационно-ориентированных (СОБД). В той же работе дается обзорное описание принципов внутреннего представления СОБД – модели конечных состояний, с отдельными из которых ассоциированы определенные наборы данных.

Подход на основе встроенных моделей был исследован применительно к системам различного характера – от управления сложными объектами до электронных документов и интернет-приложений [6–15].

Данная статья посвящена вопросам организации внешнего представления СОБД, ориентированного на работу пользователей и сторонних приложений с сохранением ситуационных «акцентов» внутреннего представления баз данных.

**1. КОНЦЕПЦИЯ  
ВНЕШНИХ ПРЕДСТАВЛЕНИЙ СОБД**

Ситуационно-ориентированная база данных предназначена для ведения данных, характеризующих развитие некоторой ситуации предметной области [1]. Для этого в структуре СОБД предусмотрены встроенная динамическая модель (модель конечных состояний), описывающая эволюцию ситуации, и массивы данных, ассоциированных с отдельными состояниями модели.

Контактная информация: (347) 272-89-81

Работа выполнена в рамках научной школы УГАТУ «Теория и практика разработки информационных систем» и поддержана грантом РФФИ 10-07-00167-а «Электронные документы со встроенной динамической моделью», а также грантом Президента Российской Федерации НШ-65497.2010.9 для ведущих научных школ

В соответствии с известным подходом в СОБД могут быть выделены два уровня:

- внутреннее представление, задающее внутреннюю структуру СОБД, описывающее, как организованы структура модели конечных состояний, наборы данных, как задано соответствие между ними;

- внешнее представление – «проекции» СОБД на работу пользователей и сторонних приложений, то, как «выглядит» структура и содержимое базы данных «со стороны».

При этом специфика организации СОБД придает ситуационный оттенок этим уровням.

Внутреннее представление СОБД определяется двумя дополняющими друг друга базовыми компонентами:

- концептуальной моделью конечных состояний развития ситуации. В работе [1] она представлена в классе иерархических ситуационных моделей, задающих наборы состояний, объединенных в субмодели, содержащих погружения в другие субмодели и связанных друг с другом переходами. Каждому состоянию динамической модели ставится в соответствие свой набор данных, имеющий значение на определенном этапе развития ситуации;

- физической реализацией модели конечных состояний с ассоциированным набором данных. В работе [1] предлагается реализация концептуальной модели на платформе XML: каждому компоненту динамической модели ставится в соответствие свой XML-элемент, а наборы данных определяются иерархией XML-элементов и атрибутов.

Здесь принципиально важно определить, что именно подразумевается под внешним представлением СОБД. В действительности уже известен термин внешнего представления применительно к традиционным (главным образом, реляционным) базам данных. Попытаемся перенести его в СОБД. Под внешним представлением СОБД будем понимать пользовательское представление информации из СОБД, отражающее «срез» информации из базы данных применительно к потребностям конкретных пользователей и конечных приложений, скрывающее от них реальную внутреннюю структуру базы данных.

По аналогии с внутренним внешнее представление также может быть представлено двумя уровнями – концептуальной моделью, задающей правила построения представления, и ее физической реализацией.

Анализ принципов организации внутреннего представления СОБД позволяет сформулировать

основные требования к построению внешних представлений:

- внешнее представление должно предоставлять возможность отслеживания данных, ассоциированных с текущим состоянием (или иерархией текущих состояний) модели ситуации, описанной во внутреннем представлении СОБД;

- внешнее представление должно поддерживать различные виды операций по обработке данных из СОБД, не затрагивая ни их внутренней структуры, ни содержимого. Фактически такое представление сопоставимо с немодифицируемыми представлениями, используемыми с традиционными базами данных – информация доступна только для отображения в различных срезах, но не для редактирования. Внешние представления не задуманы как средство ведения данных;

- структура внешнего представления должна отражать ситуационную архитектуру внутреннего представления. Отдельные ситуации предполагают наличие ассоциированных с ними не только данных на уровне внутреннего представления, но и правил обработки на уровне внешнего представления;

- внешнее представление должно обеспечивать ситуационный дуализм:

- 1) ситуационно-зависимая обработка – возможность отслеживания данных, ассоциированных с текущим состоянием (или их иерархией), манипулирования ими в соответствии с заданным набором правил;

- 2) ситуационно-независимая обработка – возможность формирования внешних представлений в контексте рассмотрения набора данных в СОБД в отрыве от ее ситуационной составляющей;

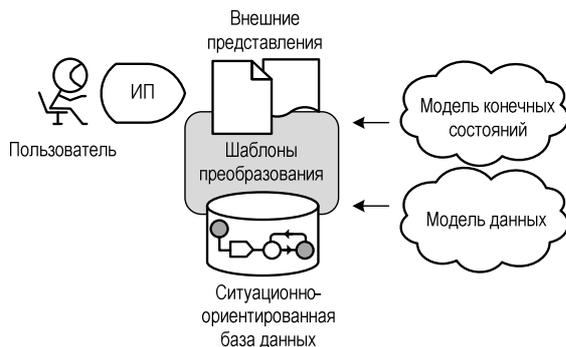
- внешнее представление должно предоставлять возможность наращиваемости, сборку единого представления из нескольких фрагментов, каждый из которых, в свою очередь, формируется в результате работы правил, ассоциированных с состояниями текущей иерархии;

- на физическом уровне внешнее представление должно обеспечивать платформенезависимость и расширяемость для последующего использования различными платформами. Такой подход обеспечит переносимость одного и того же набора правил на различные платформы, позволит указанным правилам обрабатываться с использованием разнородных аппаратных и программных технологий.

С учетом сформулированных требований уточним понятие внешнего представления СОБД. Под внешним представлением СОБД

будем понимать набор обобщенных и ситуационно-ориентированных правил обработки информации из базы данных, результатом действия которых является пользовательское представление данных в виде электронных документов и их фрагментов.

Работу с внешними представлениями можно условно представить как последовательную смену этапов: от разработки модели конечных состояний, сопоставления ее с моделью данных и ведения базы данных, до программирования правил преобразования и генерации внешних представлений данных (рис. 1).



**Рис. 1.** Общая схема работы внешних представлений СОБД

Основываясь на приведенных требованиях и специфике архитектуры СОБД, определим два типа внешних представлений:

- внешнее представление без привязки к текущему состоянию динамической модели. Внешнее представление формируется для всей встроенной модели в целом, независимо от того, какое ее состояние (или состояния) является на данный момент времени текущим. В представление вовлекаются данные, ассоциированные как с моделью в целом (с головным состоянием), так и с ее состояниями;

- внешнее представление в контексте текущего состояния. Такое представление напрямую зависит от того, на каком этапе в данный момент времени находится развитие ситуации. В процессе формирования внешнего представления используются те данные, которые ассоциированы с текущим состоянием динамической модели, а также те данные, которые соответствуют всем состояниям, родительским относительно текущего состояния (в соответствии с правилами организации иерархических ситуационных моделей такие состояния также являются текущими, имеет место иерархия текущих состояний).

## 2. СТРУКТУРНАЯ МОДЕЛЬ ВНЕШНЕГО ПРЕДСТАВЛЕНИЯ СОБД

В соответствии с приведенной выше концепцией, внешнее представление СОБД объединяет в себе две составляющие:

- ситуационную, отражающую динамическую модель внутреннего представления базы данных;
- прикладную, представленную набором правил, которые должны быть применены в той или иной ситуации.

Целесообразно представить взаимодействие этих составляющих в форме структурной модели, в обобщенном виде описывающей класс внешних представлений СОБД.

Важно отметить, что каждое конкретное внешнее представление будет определяться как спецификой СОБД, к которой оно применяется, так и собственным назначением. Однако в структуре любых внешних представлений однозначно присутствуют общие компоненты, которые и описываются структурной моделью.

Обсуждение структурной модели целесообразно предварить рассмотрением возможных технологий физической реализации внешних представлений. Очевидно, что при выборе способа реализации внешнего представления следует отталкиваться от того, каким образом устроено внутреннее представление базы данных.

В работе [1] внутренняя структура СОБД предложена в форме комбинированной XML / XSD-разметки как регламентирующей структуру данных, так и описывающей их содержимое. В этой связи целесообразно строить внешние представления в такой же XML-ориентированной манере.

На сегодняшний день сложно отыскать такую программную технологию или платформу, которая в своих решениях не поддерживала бы XML, это и языки программирования с библиотеками для работы с XML, и «родные» XML-технологии, являющиеся официальными дополнениями к исходной разметке.

Набор правил, содержащихся во внешнем представлении, можно расценивать как набор программных инструкций, выполняющих обработку информации из базы данных. Важно обеспечить независимость этих правил от программы их «запуска» – изменение интерпретатора не должно влиять на внешнее представление. Этого сложно добиться с помощью традиционных языков программирования – обычно и правила обработки, и их интерпретатор совмещены. Намного эффективнее использовать XML-технологии, отделяя процедуру обработки

правил от их описания. Кроме того, использование традиционных языков программирования «потянет за собой» избыточную для внешнего представления функциональность.

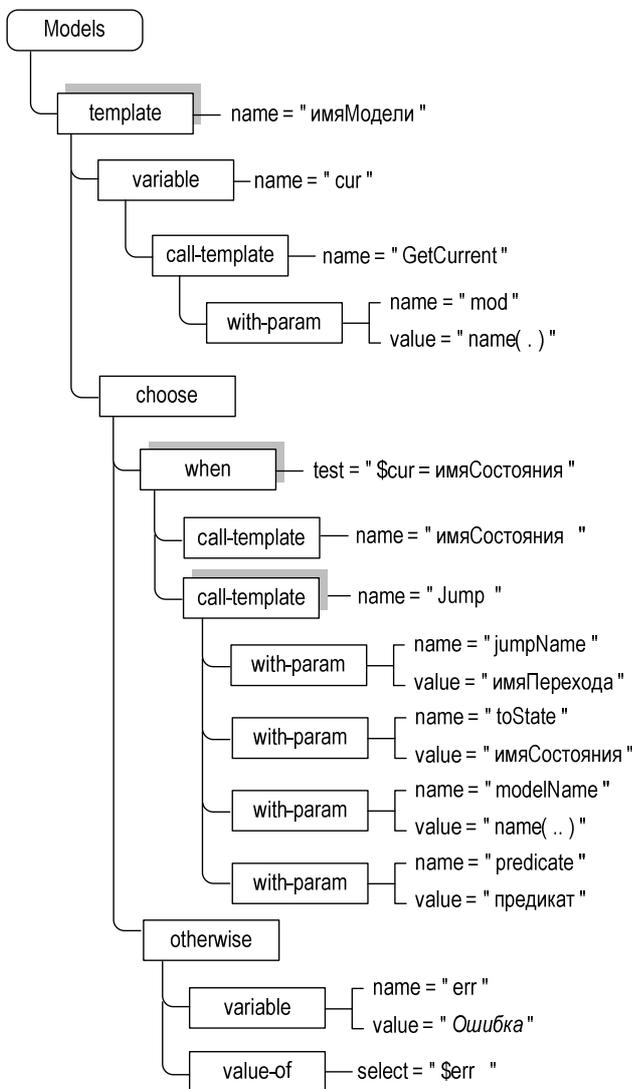
Обобщая опыт работы с XML-технологиями, целесообразно задачу генерации внешних представлений свести к задаче трансформации XML-данных СОБД. Очевидное решение такой задачи – применение технологии XSL-трансформации.

Технология XSLT (Extensible stylesheet language for transformations) – это технология, позволяющая преобразовать XML-документ в другой XML-документ, в HTML-документ для отображения веб-браузером, в документы иных форматов [2, 4, 5]. Базовым элементом XSLT-

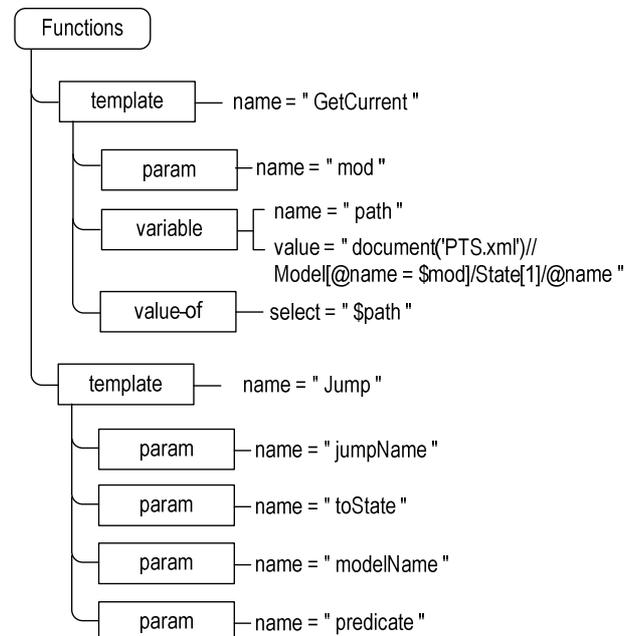
преобразования являются XSL-таблицы стилей, содержащие набор правил преобразования исходного документа, написанных на языке XSL и предназначенных для XSLT-процессора. При этом исходным XML-документом является СОБД, а под правилами преобразования подразумеваются правила внешнего представления СОБД.

Обсудим структурную модель внешнего представления СОБД в терминах XSLT-трансформации. Внешнее представление задается набором взаимовызываемых XSL-шаблонов трансформации, задающих правила обработки XML-данных из СОБД (рис. 2).

**Model.xslt**



**Functions.xslt**



**States.xslt**

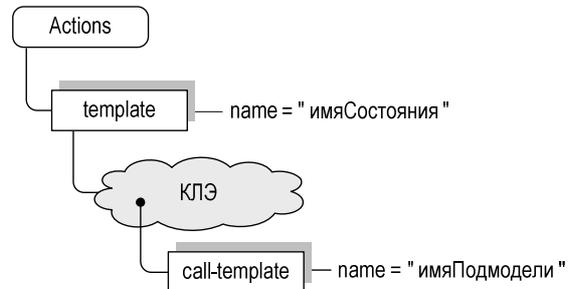


Рис. 2. Концептуальная модель внешних представлений СОБД на основе XSL-шаблонов

Принципиально важным является различие XSL-описания ситуационно-зависимых и ситуационно-независимых внешних представлений. С точки зрения XSL-трансформации, простейшим является внешнее представление, не зависящее от текущей ситуации. В таком случае внешнее представление – это стандартный XSL-файл со стандартным набором правил извлекающей трансформации, применяемый к корневому XML-элементу (головному состоянию модели СОБД).

Несколько сложнее обстоит дело с ситуационно-зависимыми представлениями. Их особенность заключается в ситуационно-ориентированной архитектуре, отражающей структуру внутреннего представления СОБД. Это значит, что в модели внешнего представления должны быть описаны субмодели и состояния динамической модели ситуации.

Для наглядности структурную модель ситуационно-зависимого представления СОБД можно представить в виде двух взаимосвязанных компонент. Первая из них предназначена для обработки субмоделей, а вторая – для работы с состояниями.

Множество правил обработки субмоделей размещается в XSL-таблице стилей *Models.xslt*. Для поддержания корректности XML-документа (каковым является таблица стилей) в представлении задан корневой элемент *Models*, родительский по отношению ко всем шаблонам-правилам.

Соотнесение субмоделей во внутреннем и внешнем представлениях СОБД осуществляется с помощью атрибута *name* в таблице стилей *Models*. Значением этого атрибута является название соответствующей субмодели во внутреннем представлении (предполагается глобальная уникальность имен субмоделей и состояний).

Каждый шаблон обработки субмоделей содержит ссылку на то состояние, которое в данный момент времени является в нем текущим (конечно, если сама субмодель является текущей). Для этого в структуре модели предусмотрена переменная *cur*, значением которой является название текущего в данной субмодели состояния. Фактически, здесь имеет место косвенная XPath-адресация: точкой отсчета в выражении выступает узел, соответствующий текущей субмодели.

Ситуационная структура субмодели во внешнем представлении имитируется с помощью вызовов шаблонов, соответствующих дочерним состояниям. Во избежание многократного выполнения правил, соответствующих всем со-

стояниям субмодели, вводится конструкция, выбирающая только текущее состояние. В XSL эта задача решена с помощью конструкции *choose*, расширяющей традиционную условную инструкцию *if..else*. Для каждого дочернего состояния субмодели вводится XSL-инструкция *when*, истинность булева параметра которой определяется соответствием названию текущего состояния из переменной *cur*. Невозможность выполнения ни одного булева условия *when* свидетельствует об ошибке интерпретации, что подтверждается соответствующими действиями, вводимыми инструкцией *otherwise*.

В соответствии с синтаксисом иерархических ситуационных моделей, одноуровневое взаимодействие между состояниями субмодели выполняется с помощью переходов. В модели обработки субмоделей предусмотрены конструкции, имитирующие такие переходы. Они представлены XSL-инструкцией вызова суррогатного (искусственного) шаблона *jump*, универсального для всех переходов в модели. Это пустой шаблон, параметры которого задают название, цель и условия выполнения перехода. Такая конструкция не несет никакой практической нагрузки, а предназначена исключительно для «прорисовки» ситуационного «рельефа» модели.

Для правил обработки состояний субмоделей модели СОБД используется таблица стилей *States.xslt*. Концептуально эта структура намного проще, чем описание правил работы с состояниями. Однако именно этот на первый взгляд простой элемент и несет в себе основную смысловую нагрузку внешнего представления СОБД. Здесь определяются не просто правила идентификации текущих состояний (хотя и не без этого), но задаются правила обработки данных, ассоциированных с такими состояниями. Такие инструкции задаются с помощью специальных элементов XSL-трансформации – конечных литеральных элементов, описывающих «внешний вид» результирующего документа.

По правилам построения иерархических ситуационных моделей [3], многоуровневое взаимодействие между состояниями выполняется с помощью погружений в дочерние субмодели. В модели внешнего представления такой эффект достигается вызовом соответствующих шаблонов из таблицы стилей *Models.xslt*, описывающей правила обработки субмоделей. Название субмодели, в которую выполняется погружение, отражается в названии вызываемого шаблона трансформации. Важно отметить, что здесь не используется никаких условных конструкций, поскольку предполагается, что по

умолчанию выполняются все погружения из текущего состояния.

Дополнительный компонент внешнего представления `Functions.xslt` не имеет никакого отношения к ситуационной архитектуре. В нем объединены обе служебные функции, используемые в стилях трансформации `Model.xslt` и `States.xslt`:

- `GetCurrent` – поиск текущего состояния в заданной субмодели на основании данных из памяти текущего состояния СОБД;

- `jump`: искусственно введенный шаблон, имитирующий переходы в ситуационной структуре ситуационно-ориентированного внешнего представления.

Таким образом, внешнее представление задается набором из трех XSL-таблиц стилей, описывающих правила обработки субмоделей и состояний модели конечных состояний, а также ассоциированных с ними данных из СОБД.

### 3. МЕТОД ИНТЕРПРЕТАЦИИ ВНЕШНИХ ПРЕДСТАВЛЕНИЙ СОБД

Метод интерпретации внешних представлений определяется их теоретическими и практическими аспектами. Определяющим фактором при этом является двойственная природа внешних представлений СОБД: с одной стороны, это ситуационно-ориентированная структура, а с другой – обычные таблицы стилей, используемые в XSL-трансформации.

Обработка внешнего представления СОБД выполняется в два этапа. На начальной стадии интерпретации анализируется память текущего состояния на предмет определения текущих состояний и субмоделей модели ситуации. На следующем этапе управление передается правилам обработки, размещенным во внешнем представлении: к XML-данным, соответствующим текущим состояниям, применяются XSL-инструкции из шаблонов трансформации.

Важно отметить, что предварительная обработка памяти текущего состояния касается только ситуационно-зависимых внешних представлений. Если внешнее представление относится к категории ситуационно-независимых, то в процесс вовлекается только XSL-процессор, который выполняет стандартную процедуру XSL-трансформации на основе XSL-таблиц стилей.

В обработке ситуационно-зависимых внешних представлений СОБД задействовано два программных механизма – собственно интерпретатор внешних представлений, выполняющий манипуляции с ситуационной структурой

СОБД, и XSL-процессор, отвечающий за преобразование XML-данных СОБД в заданную структуру в соответствии с определенными в представлении правилами.

Принципиальное значение имеет тот факт, что внешние представления ориентированы на форсирующую обработку и представляют собой «библиотеку» именованных шаблонов. Для XSL-процессора такой информации недостаточно – для выполнения трансформации необходима «точка отсчета», т. е. тот шаблон, с которого должен запускаться процесс.

Для указания «стартового» шаблона вводится дополнительная таблица стилей `Transform.xslt` (рис. 3). Он подключает шаблоны внешнего представления, содержащие правила обработки субмоделей и состояний, а также шаблоны, упрощающие эту обработку. На уровне XSL такие включения выполняются с помощью инструкции `include`, вводимой в XSL-таблицу стилей до объявления всех остальных конструкций. Помимо подключения стилей трансформации таблица `Transform.xslt` содержит единственный шаблон, который применяется к корневому элементу обрабатываемого XML-документа (к памяти текущего состояния). Этот шаблон будет вызываться всегда, независимо от содержимого документа, к которому он применяется. В составе объявленного шаблона обработки корневого XML-элемента содержится единственная инструкция, вызывающая шаблон той субмодели, которая содержит текущее на данный момент состояние, или само текущее состояние. Структура модели конечных состояний, определяющая архитектуру СОБД, предполагает наличие головного состояния, которое всегда является текущим, служит корнем иерархии текущих состояний. Поэтому в таблице стилей `Transform.xslt` по умолчанию вызывается головное состояние из динамической модели СОБД.

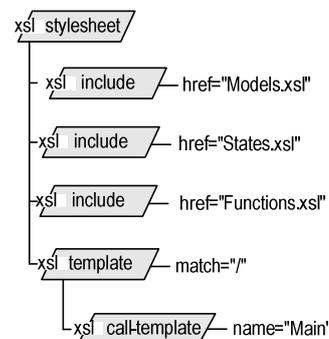


Рис. 3. «Стартовый» шаблон трансформации внешних представлений

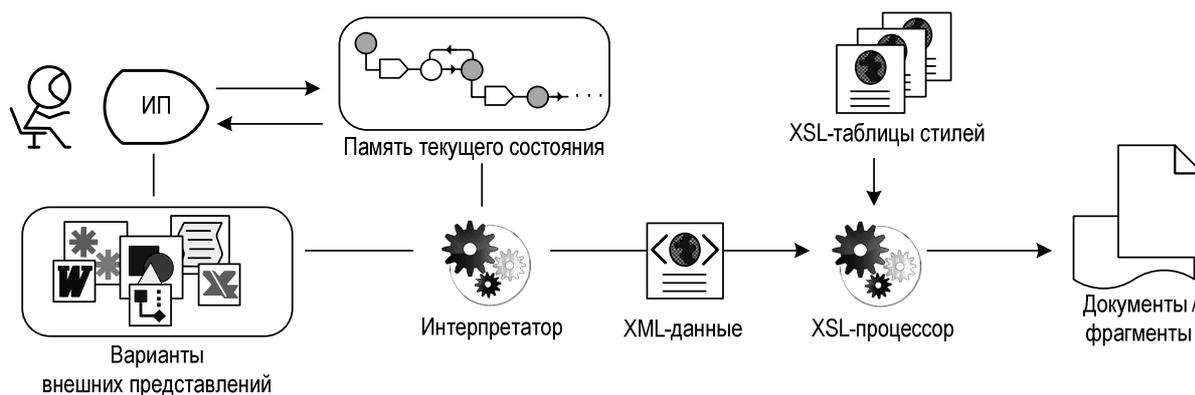


Рис. 4. Схема интерпретации внешних представлений СОБД

С учетом описанных компонентов интерпретации, процесс обработки внешних представлений СОБД можно представить следующим образом (рис. 4).

Интерпретатор внешнего представления на начальном этапе обработки анализирует память текущего состояния и строит иерархию состояний, являющихся текущими на данный момент времени. Далее все тем же интерпретатором выполняется поиск того внешнего представления, которое можно применить в данной ситуации. Окончательный выбор используемого представления остается за пользователем, работающим с СОБД. После выбора представления интерпретатор загружает соответствующую таблицу стилей Transform.xslt. На этом этапе в процесс обработки вступает XSL-процессор. Он запускает таблицу стилей Transform.xslt и подключает соответствующие таблицы стилей обработки субмоделей и состояний. На следующем шаге XSL-процессор работает и с XSL-шаблонами трансформации, и с XML-данными памяти текущего состояния. На основе обрабатываемого XML-файла производится последовательный вызов шаблонов, соответствующих текущим состояниям и их субмоделям. Результатом XSL-преобразования является конечный документ, который и является внешним представлением заданных данных из СОБД.

#### 4. ПРИМЕР

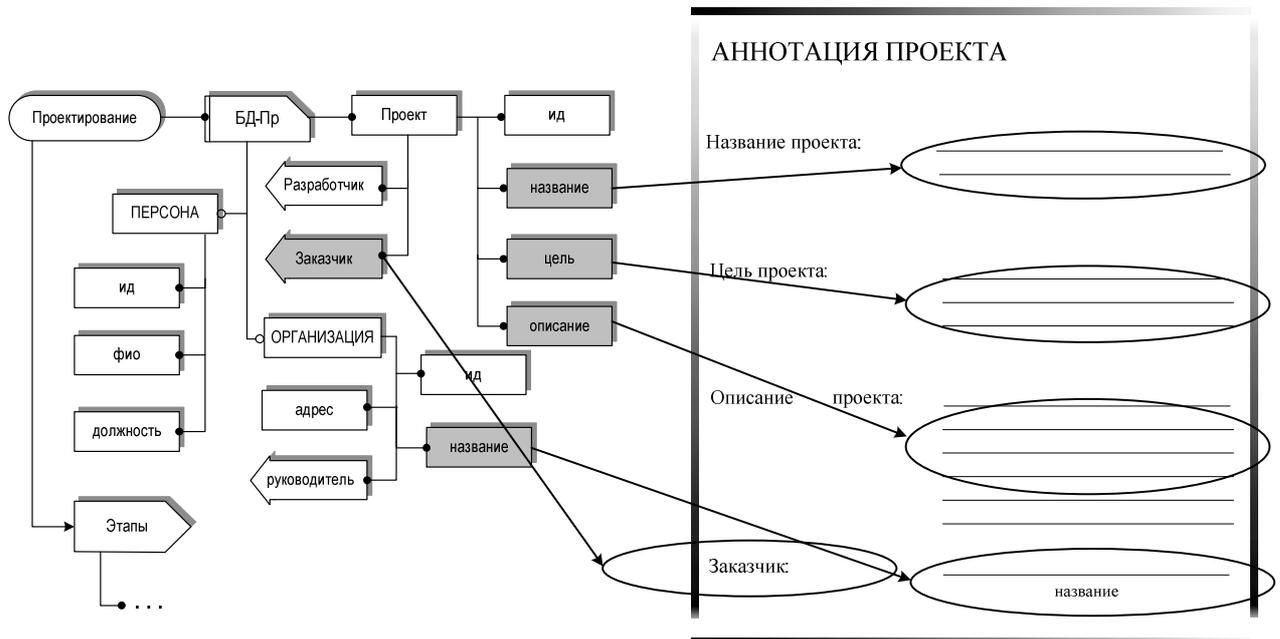
В работе [1] СОБД рассматривается на примере развития процесса проектирования технического изделия, проходящего целый ряд этапов – от сбора требований и подготовки технического задания до формирования технического проекта. Продолжим рассмотрение того же примера для демонстрации концепции внешних

представлений, их модели и метода интерпретации.

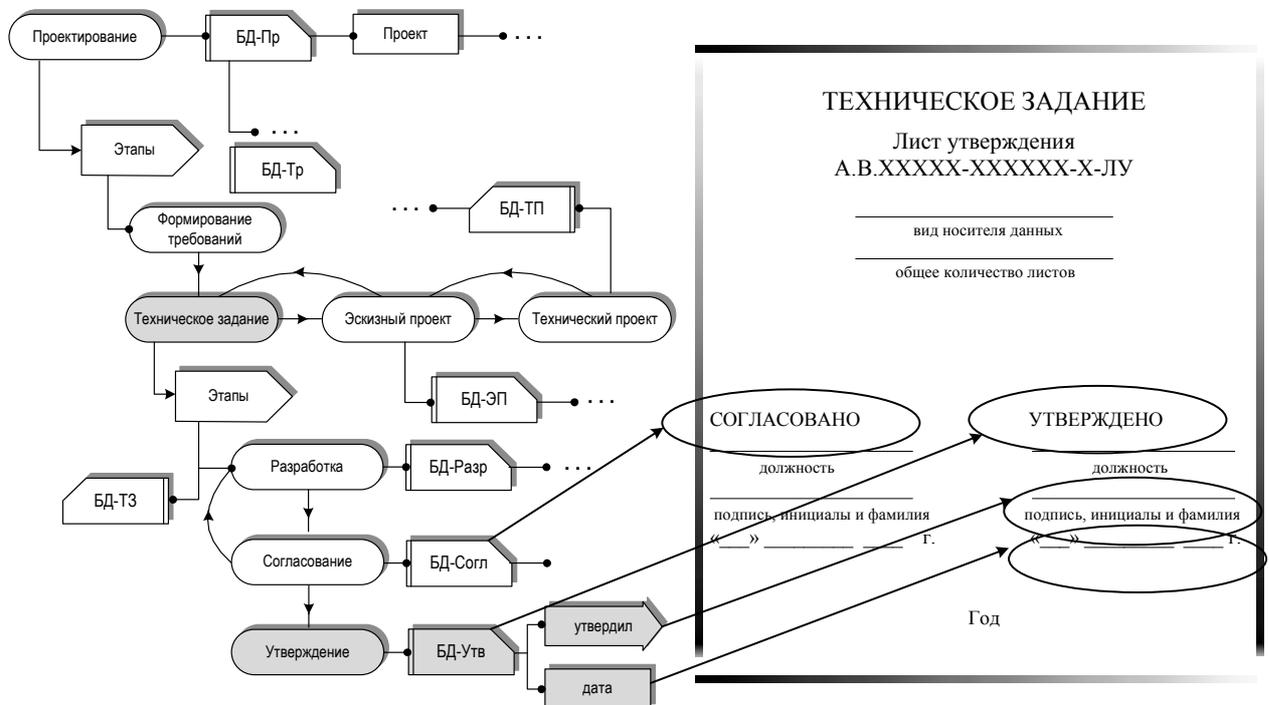
Проектирование технического изделия традиционно сопровождается многочисленными итерациями обработки данных, ассоциированных с отдельными этапами этого процесса. К ним относятся как документы, сопровождающие этапы проектирования, так и результаты разнообразных запросов к данным. Такую обработку данных можно эффективно сочетать с внешними представлениями соответствующей СОБД.

В качестве примеров рассмотрим два вида внешних представлений – ситуационно-независимые, применяемые ко всему набору данных модели процесса проектирования, и ситуационно-зависимые, определяемые текущим этапом проектирования (рис. 5).

Ярким примером внешнего представления первого вида является документ под названием «Аннотация проекта». Этот документ предоставляет наиболее общую информацию о выполняемом проекте и доступен на любом этапе проектирования. В аннотации должны быть представлены название проекта, его цель и описание, а также информация о его заказчиках и исполнителях. В соответствии с моделью процесса проектирования такая информация связана с головным состоянием «Проектирование» динамической модели СОБД и представлена в наборе данных «БД-Пр». Так, название, цель и описание проекта хранятся в группе данных «Проект», а данные о заказчике можно получить через одноименный элемент, ссылающийся на группу данных «Организация». Все указанные данные никак не связаны с ситуационной архитектурой СОБД, развитие процесса проектирования не влияет на доступность этой информации, а следовательно, и на возможность формирования внешнего представления.



а



б

Рис. 5. Примеры ситуационно-независимого (а) и ситуационно-зависимого (б) внешних представлений

На рис. 5, *а* показано сопоставление между моделью конечных состояний СОБД с ассоциированными данными и аннотацией проекта. Серой заливкой отмечены те элементы данных СОБД, которые используются в заявленном внешнем представлении. Овалами отмечены фрагменты документа, содержимое которых определяется информацией из СОБД.

Другой пример внешнего представления относится к категории ситуационно-зависимых. В качестве примера рассмотрим один из документов, сопровождающих этап подготовки технического задания на проектирование. В соответствии с динамической моделью проектирования формирование технического задания включает в себя три этапа: непосредственно разработку, согласование проекта и его утверждение. На последнем из перечисленных этапов используется документ «Титульный лист» технического задания.

Это сравнительно простой по своей структуре документ с двумя группами данных – соответственно об утверждении и согласовании документа.

Источником данных об утверждении технического задания является группа данных «БД-Утв», ассоциированных с состоянием «Утверждено» подмодели «Этапы» состояния «Техническое задание». Группа данных содержит два элемента – «дата» и «утвердил», которые используются на титульном листе, т. е. в соответствующем внешнем представлении. Так, к примеру, информация о дате утверждения передается внешнему представлению из одноименного элемента данных, параметры утвердившего документа (должность, фамилия и инициалы) определяются элементом «Утвердил», ссылающимся на элемент «Персона».

Важно отметить еще одну группу данных во внешнем представлении – информацию о согласовании. Хотя эта информация и не относится к текущему состоянию, однако она соответствует предшествующему ему состоянию. Такое решение укладывается в концепцию внешних представлений СОБД – для обработки доступны не только данные, с которыми пользователь работает на данном этапе, но и та информация, с которой он уже работал на пройденных этапах.

Очевидна ситуационная природа внешнего представления – ведется работа с данными, ассоциированными с текущим состоянием («Техническое задание»), с предшествующими ему состояниями, а также с обобщенными данными, связанными с головным состоянием («Проектирование»), которое также является текущим.

На рис. 5, *б* показано сопоставление между моделью конечных состояний СОБД с ассоциированными данными и титульным листом технического задания. Серой заливкой отмечены те элементы данных СОБД, которые используются во внешнем представлении, а также текущие состояния модели. Овалами отмечены фрагменты документа, содержимое которых формируется обработкой информации из СОБД.

## ВЫВОДЫ

- Внешнее представление ситуационно-ориентированной базы данных представляет собой набор обобщенных и ситуационно-ориентированных правил обработки информации из базы данных, результатом действия которых является пользовательское представление данных в виде электронных документов и их фрагментов.

- В зависимости от назначения внешние представления могут быть ситуационно-зависимыми (обработка данных выполняется в контексте определенного состояния) и ситуационно-независимыми (работа ведется независимо от текущего этапа развития ситуации).

- Применение XSL-трансформации является эффективным способом построения внешних представлений ситуационно-ориентированных баз данных ввиду XML-организации их внутреннего представления.

- Концептуальная модель внешних представлений определяется их XSL-архитектурой – для субмоделей и состояний динамической модели СОБД предусматриваются XSL-шаблоны трансформации, инструкции в которых задают правила преобразования данных СОБД.

- Интерпретация внешних представлений СОБД также определяется их XSL-структурой. В этой связи обработка ведется в два этапа: на первом этапе интерпретатор внешних представлений анализирует память текущего состояния и определяет доступные правила преобразования, на втором этапе XSL-процессор выполняет применение XSL-инструкций к XML-данным из СОБД.

- Применение внешних представлений предоставляет широкие возможности по манипулированию ситуационно-ориентированными базами данных, позволяя по-разному обрабатывать одни и те же данные в зависимости от того, на каком этапе находится развитие ситуации, описываемой в базе данных.

## СПИСОК ЛИТЕРАТУРЫ

1. **Миронов В. В., Юсупова Н. И., Шакирова Г. Р.** Ситуационно-ориентированные базы данных: концепция, архитектура, XML-реализация // Вестник УГАТУ. 2010. Т. 14, № 2(37). С. 233–244.
2. **Миронов В. В., Юсупова Н. И.** XML-технологии в базах данных. Введение: Учеб. пособие. Уфа: Уфимск. гос. авиац. техн. ун-т, 2004. 182 с.
3. **Юсупова Н. И.** Критические ситуации и принятие решений при управлении в условиях помех: Монография. Уфа: Гилем, 1997. 112 с.
4. **Кэй М.** XSLT. Справочник программиста. СПб.: Символ-Плюс, 2001. 408 с.
5. **Валиков А. Н.** Технология XSLT. СПб.: БХВ-Петербург, 2002. 544 с.
6. **Миронов В. В., Шакирова Г. Р.** Концепция динамических XML-документов // Вестник УГАТУ. 2006. Т. 8. № 5. С. 58–63.
7. **Миронов В. В., Ахметшин Р. Ф.** Асинхронная децентрализованная интерпретация иерархических ситуационных моделей // Там же. 2003. Т. 4. № 1. С. 108.
8. **Миронов В. В., Шакирова Г. Р.** Интерпретация XML-документов со встроенной динамической моделью // Там же. 2007. Т. 9. № 2. С. 88–97.
9. **Миронов В. В., Шакирова Г. Р., Яфаев В. Э.** Информационная технология персонализации электронных документов Microsoft Office в web-среде на основе XML // Там же. 2008. Т. 10. № 2. С. 112–122.
10. **Миронов В. В., Шакирова Г. Р.** Программно-инструментальное средство для создания и ведения динамических XML-документов // Там же. 2007. Т. 9. № 5. С. 54–63.
11. **Миронов В. В., Шакирова Г. Р., Яфаев В. Э.** Иерархическая модель персонализированных документов и ее XML-реализация // Там же. 2008. Т. 11. № 1. С. 164–174.
12. **Миронов В. В., Шакирова Г. Р.** Обработка XML-документов со встроенной моделью // Обозрение прикладной и промышленной математики. 2008. Т. 15. № 2. С. 336–336.
13. **Миронов В. В., Шакирова Г. Р.** Электронные документы со встроенной динамической моделью на основе XML: Монография. 2009. 179 с.
14. **Миронов В. В., Маликова К. Э.** Интернет-приложения на основе встроенных динамических моделей: идея, концепция, безопасность // Вестник УГАТУ. 2009. Т. 13, № 2. С. 167–179.
15. **Миронов В. В., Маликова К. Э.** Интернет-приложения на основе встроенных динамических моделей: архитектура, структура данных, интерпретация // Там же. 2010. Т. 14, № 1(36). С. 154–163.

## ОБ АВТОРАХ



**Миронов Валерий Викторович**, проф. каф. автоматизир. систем упр-я. Дипл. радиофизик (Воронежск. гос. ун-т, 1975). Д-р техн. наук по упр. в техн. сист. (УГАТУ, 1995). Иссл. в обл. иерархич. моделей и ситуац. управления.



**Юсупова Нафиса Исламовна**, проф., зав. каф. выч. мат. и киб., декан ФИРТ. Дипл. радиофизик (Воронежск. гос. ун-т, 1975). Д-р техн. наук по упр-ю в техн. сист. (УГАТУ, 1998). Иссл. в обл. критич. сит. упр-я, информатики.



**Шакирова Гульнара Равилевна**, доц. каф. автоматизир. систем упр-я. Дипл. инженер по АСОИУ (УГАТУ, 2005). Канд. техн. наук по матем. и прогр. обеспечению вычисл. машин, комплексов и комп. сетей (УГАТУ, 2008). Иссл. в обл. XML-технологий.