

УДК 004.91

В. В. МИРОНОВ, Н. И. ЮСУПОВА, Г. Р. ШАКИРОВА

**СИТУАЦИОННО-ОРИЕНТИРОВАННЫЕ БАЗЫ ДАННЫХ:  
КОНЦЕПЦИЯ, АРХИТЕКТУРА, XML-РЕАЛИЗАЦИЯ**

Рассматривается ситуационно-ориентированный подход к базам данных в рамках нового класса – ситуационно-ориентированных баз данных. Формулируются основные принципы организации ситуационно-ориентированных баз данных, обсуждаются их архитектурные и структурные особенности в контексте концептуальной диаграммы и графовой модели. Иллюстрируется возможность использования ситуационно-ориентированных баз данных при решении практических задач. *Базы данных; ситуационно-ориентированные базы данных; встроенные динамические модели; XML-технологии*

**ВВЕДЕНИЕ**

В настоящее время любая область деятельности, поддерживаемая информационными технологиями, немислима без баз данных. Их роль важнее, чем просто накопление и хранение информации – все чаще базы данных используются для поддержки принятия решений, выступают как сложные интеллектуальные системы. В них отражаются любые нововведения в области аппаратного и программного обеспечения. Новые стандарты, новые технологии, новые форматы – все неизбежно затрагивает сферу баз данных. Это во многом обуславливает многообразие как классов баз данных, так и признаков их классификации.

Одним из ключевых признаков, по которым принято классифицировать базы данных, является вид модели данных, в соответствии с которой осуществляется управление данными. В этом плане различают иерархические, сетевые, реляционные, объектные базы данных, основанные, соответственно, на иерархической, сетевой, реляционной и объектной моделях данных. Каждый из этих классов имеет свои недостатки и достоинства, которые позволяют выбрать тот из них, который наиболее подходит разработчику в тех или иных условиях.

Во многих случаях моделируемые в базе данных бизнес-процессы носят ситуационный характер, т. е. могут быть интерпретированы как смена этапов некоторой ситуации. Напри-

мер, процесс проектирования технического изделия представляет собой совокупность последовательно / параллельно выполняемых этапов: техническое задание, эскизный проект, рабочий проект. Часть информации должна быть внесена на одном этапе, другая – на следующем этапе и т. д. В этих условиях возникает необходимость создания и ведения данных, характеризующих соответствующие этапы развития ситуации выполнения проекта.

В настоящее время на уровне систем управления базами данных (СУБД) неизвестны механизмы управления данными в зависимости от текущей ситуации. Традиционный подход к организации баз данных не ориентирован на *ситуационную* «фильтрацию» информации пользователей. Эту задачу приходится решать на уровне приложений.

В этом контексте предлагается применить к базам данных технологию встраивания динамических моделей. Этот подход успешно применяется в системах разного рода – от систем управления техническими объектами до электронных документов [1, 4]. В процессе разработки и исследования динамических документов [4], т. е. таких документов, в которые встраиваются динамические модели, отражающие существенные этапы их жизненного цикла (ситуации использования), был осознан тот факт, что в этом случае требуется особый класс баз данных, в основе которых лежит динамическая (ситуационная) модель.

Данная статья посвящена изложению концепции ситуационно-ориентированных баз данных, управляемых встроенными динамическими моделями, и обсуждению ее возможной реализации на платформе XML.

---

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 встроенными динамическими моделями»

## 1. КОНЦЕПЦИЯ СИТУАЦИОННО-ОРИЕНТИРОВАННЫХ БАЗ ДАННЫХ

**Определение.** Ситуационно-ориентированная база данных<sup>1</sup> (СОБД) предназначена для ведения данных, характеризующих развитие некоторой ситуации предметной области, на двух уровнях:

- *макроуровне* – в виде укрупненных (макро-) состояний;

- *микроуровне* – в виде детальных (микро-) состояний, ассоциированных с макросостояниями,

и *отличается* тем, что с целью предоставления приложениям более удобного доступа к данным в ней предусмотрены:

- встроенная динамическая модель макросостояний, отражающая развитие ситуации на макроуровне;

- возможность активного поведения в ответ на внешние события для отслеживания текущих макросостояний;

- предоставление доступа к микросостояниям в контексте текущих макросостояний.

Это определение содержит ряд ключевых понятий: двухуровневая модель ситуации; динамическая модель макросостояний; активное поведение ситуационно-ориентированной базы данных; доступ к данным в ситуационном контексте. Эти и другие вопросы обсуждаются ниже.

**Двухуровневая модель ситуации.** Обсуждение идеи ситуационно-ориентированной организации баз данных важно предварить раскрытием ее базового термина – «ситуация». Под ситуацией принято понимать совокупность условий и обстоятельств, создающих определенную обстановку, положение [4].

Предполагается, что моделируемая в базе данных предметная область может рассматриваться как развитие некоторой ситуации, т. е. как смена ее текущих состояний. В этом плане модель ситуации можно рассматривать на двух уровнях [1]. На *верхнем* – макроуровне, описывающая качественное развитие ситуации в виде конечного множества

состояний, переходы между которыми определяются некоторым набором правил. На *нижнем* – микроуровне – модель детальных состояний, задающих ситуацию с большей степенью детализации: каждое макросостояние ситуации описывается набором своих микросостояний.

Например, рассматривая бизнес-процесс проектирования некоторого изделия как процесс развития ситуации, на макроуровне мы можем рассматривать в качестве макросостояний стандартные этапы проектирования: техническое задание, рабочий проект, технический проект и т. д., а также отдельные подэтапы этих этапов. Тогда процесс развития ситуации – это переход от одного текущего макросостояния (этапа) к другому, от одного подэтапа к другому. В качестве микросостояний ситуации здесь выступают наборы тех или иных свойств, характеризующих то или иное макросостояние: даты принятия определенных решений, лица, ответственные за исполнение определенных работ, техническая документация, подготовленная на определенных этапах / подэтапах, и т. д. Понятно, что хранение этой информации можно организовать в традиционных базах данных, например, реляционных. Но в этом случае задачу отслеживания текущих макросостояний и организацию доступа к соответствующим микросостояниям придется решать на уровне приложений, использующих базу данных. Мы же хотим переложить эти задачи, в значительной степени, на СУБД, управляющую данными.

**Динамическая модель макросостояний.** Чтобы СОБД могла учитывать текущие ситуации, необходимо располагать моделью состояний ситуации (моделью развития ситуации). Мы предполагаем, что развитие ситуации на верхнем уровне может быть задано в виде модели конечных состояний (МКС, Finite State Model – FSM) в той или иной форме (граф состояний, сеть Петри и т. д.), т. е. в виде конечного множества состояний и множества правил, описывающих условия перехода из одного состояния в другое в зависимости от внешних данных. Поэтому базу данных, управляемую этой моделью (FSM-driven database, FSM-DB), можно рассматривать как ситуационно-ориентированную. В данной работе МКС рассматривается в классе так называемых иерархических ситуационных моделей (ИСМ) [1, 2, 4].

В этом случае хранимые состояния – это сведения о том, какие состояния МКС являются текущими (а какие были в прошлом), а хранимые данные ассоциированы с состояниями МКС. Хранимые состояния и хранимые данные

<sup>1</sup> При выборе названия для нового класса баз данных был соблазн назвать их «динамическими», имея в виду, что в их основе лежит встроенная динамическая модель данных. Однако, как оказалось, термин «динамическая база данных» уже занят, хотя и не имеет широкого распространения. В ряде источников под динамической понимается база данных, информация в которой отражает изменяющуюся предметную область. Такую трактовку нельзя считать безупречной, поскольку неясно, управляется база данных моделью (model-driven), данными (data-driven) или чем-то еще.

задаются в памяти текущего состояния (ПТС) – разделе СОБД, в котором фиксируется смена текущих состояний ситуации.

Что касается микроуровня, то мы предполагаем, что каждому макросостоянию ситуации можно поставить в соответствие определенный набор данных, характеризующий микросостояния данных макросостояния. Предполагается также, что для этих данных можно заранее задать схему (т. е. модель данных), так что изменение микросостояний в рамках своего макросостояния выражается в изменениях значений данных в рамках схемы.

Например, определяя макроуровень ситуации проектирования изделия, в простейшем случае мы создаем граф состояний, вершины которого соответствуют макросостояниям (этапам проектирования), а дуги – переходам между этими состояниями. Каждой вершине поставлен в соответствие набор свойств, характеризующих этот этап.

Таким образом, если на макроуровне СОБД ее поведение (изменение микросостояний) достаточно жестко ограничено встроенной динамической моделью, то на микроуровнях динамические ограничения, вообще говоря, не предусмотрены: значения микросостояний ограничиваются статическими ограничениями, накладываемыми соответствующей схемой.

При переходе в новое макросостояние инициализируются ассоциированные микросостояния в соответствии со схемой<sup>2</sup>, после чего открывается доступ приложений к микросостояниям для создания, выборки, изменения и обновления соответствующих значений.

В этой связи важно выделить *темпоральные* СОБД, предполагающие хранение не только текущих, но и прошлых состояний, т. е. отслеживание предыстории состояний ситуации.

**Память текущего состояния.** Как и любая другая современная база данных СОБД, должна удовлетворять принципу самоописания, т. е. должна помимо собственных данных содержать их описание – метаданные, опираясь на которые СУБД осуществляет эффективное управление. В рассматриваемом классе СОБД роль метаданных на макроуровне играет встроенная динамическая модель, а на микроуровне – схемы данных микросостояний. При этом собственно данные – ПТС – на макроуровне представлены текущими макросостояниями,

а на микроуровне – значениями данных, ассоциированных с этими состояниями. Фактически речь идет о двух уровнях ПТС:

- ПТС макроуровня фиксирует смену текущих макросостояний;
- ПТС микроуровня отражает процесс «наращивания» значений данных в ходе смены текущих макросостояний.

#### **Правило «событие-условие-действие».**

Смысл правила заключается в том, что при наступлении заданного события проверяется заданное в правиле условие; в случае, если оно выполняется, то выполняется и заданное действие.

Специфика организации СОБД придает свое значение событиям и условиям. В качестве *событий* должны выступать смена текущих состояний МКС, операции создания и ведения ассоциированных данных и пр. В качестве *действий* – процессы отражения смены текущего состояния в МКС, принудительная смена текущего состояния (внутренние предикаты МКС), «перевод» значений ассоциированных данных от одного состояния к другому и т. д.

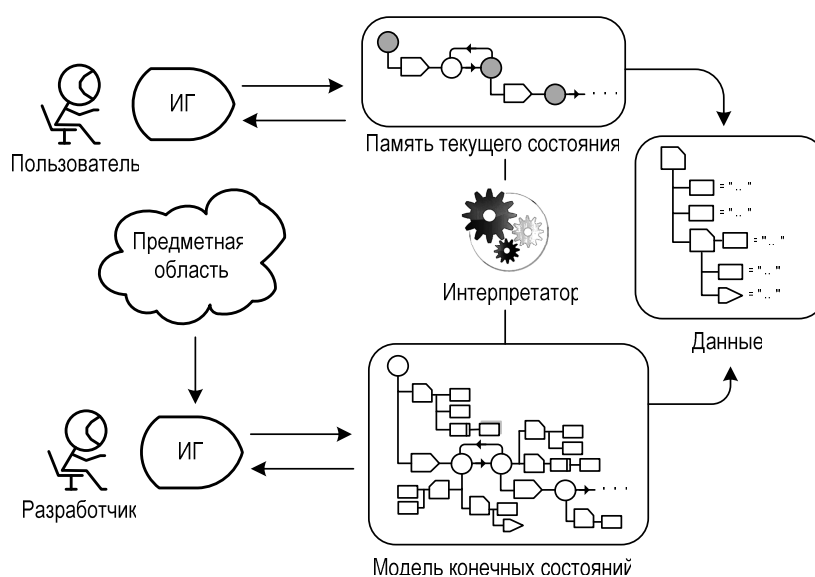
Характерным примером ЕСА-правила является ведение ПТС в МКС. Исходным событием для этого правила является операция «запуска» перехода между состояниями, условием – возможность выполнения такого перехода (внутренние и внешние предикаты), а действием – создание нового состояния в ПТС.

**Активное поведение ситуационно-ориентированной базы данных.** Развитием ЕСА-правил СОБД является ее активное поведение. По определению, база данных является *активной* (Active Database), если в ней имеются некоторые заранее определенные процессы, изменяющие состояние базы данных, которые запускаются на выполнение не по инициативе пользователей (или приложений), а системными механизмами самой базы данных в ответ на возникновение некоторых заранее определенных условий (событий).

Активное поведение должно быть отражено на обоих уровнях СОБД:

- на макроуровне – в виде отслеживания смены текущего состояния МКС и ее фиксирования в ПТС;
- на микроуровне – в форме «наращивания» значений ассоциированных данных при смене текущих состояний.

<sup>2</sup> Возможно, что начальные значения некоторых макросостояний наследуют конечные значения определенных микросостояний предшествующих макросостояний. Такое поведение должно быть задано схемой.



**Рис. 1.** Взаимодействие различных категорий пользователей с ситуационно-ориентированной базой данных

**Доступ к данным в ситуационном контексте.** В предельных случаях СОБД должна обеспечивать традиционную функциональность, т. е. доступ к хранимым данным независимо от текущего состояния ситуации. В этом плане можно различать:

- ситуационно-зависимые данные, ассоциированные с текущими состояниями ситуации и требующие контроля текущего состояния для доступа к ним;

- ситуационно-независимые данные, соответствующие любым состояниям ситуации и не требующие знания текущего состояния для доступа к ним.

В первом случае СОБД дает пользователю определенные преимущества, выражающиеся в том, что СОБД берет на себя контроль текущих состояний ситуации, а также обеспечивает доступ к данным в контексте текущего состояния.

В случае ситуационно-независимой манеры организации данных информация может быть как введена на начальном этапе некоторой ситуации, так и дополнена в процессе ее развития.

Возможен также дуализм доступа: одни и те же данные могут быть извлечены как с помощью ситуационно-зависимых запросов, учитывающих текущие состояния, так и с помощью традиционных ситуационно-независимых запросов, не учитывающих текущие состояния модели.

**Создание и ведение СОБД.** Работу с СОБД можно представить в виде двух этапов (рис. 1). На начальном этапе разработчик базы данных

на основе анализа предметной области задает структуру МКС, описывает состав данных и определяет соответствие между данными и состояниями модели. На следующем этапе формируется экземпляр базы данных – ПТС, где фиксируется перемещение между состояниями модели и задаются значения ассоциированных данных.

**Принципы организации СОБД.** По аналогии с правилами построения моделей данных можно сформулировать основные принципы, которым должна соответствовать СОБД, восходящие к известным правилам Кодда, сформулированными для реляционной модели [6].

В общем виде СОБД должна соответствовать следующим принципам.

1. Доступ к данным в МКС – к накапливаемым и хранимым в базе данных (текущие состояния и ассоциированные с ними данные) должны быть применимы те же операции, что и к метаданным (структура динамической модели и структура ассоциированных данных).

2. Физическая независимость данных – организация работы с данными не должна зависеть от способов хранения данных на носителях, от аппаратного обеспечения компьютеров, на которых находится база данных.

3. Гарантированный доступ к данным – доступ к данным должен быть однозначным. К каждому элементу данных должен быть гарантирован доступ с помощью комбинации текущего состояния модели, номера его экземпляра или имени самого элемента данных.

4. Полнота подмножества языка – система управления СОБД должна поддерживать хотя

бы один язык для работы с данными в ситуационно-ориентированной форме, который обеспечивает операции определения данных, определения представлений, манипулирования данными, ограничители целостности.

5. Наличие высокоуровневых операций управления данными – операции вставки, модификации и удаления данных должны поддерживаться по отношению к любому множеству элементов данных.

**Платформа реализации СОБД.** Принципы, которым должна соответствовать СОБД, определяют выбор платформы для ее реализации (и построения системы управления такой базой данных). В этой связи целесообразно построение СОБД на платформе XML ввиду двух основных ее преимуществ:

- разнообразия XML-технологий и широкого спектра их возможностей по описанию и ведению любого множества данных;
- аппаратной и программной независимости XML-данных.

В современных условиях традиционные реляционные СУБД поддерживают обработку XML посредством одноименного типа данных. Поэтому XML-реализация СОБД возможна и средствами таких СУБД.

## 2. АРХИТЕКТУРНЫЕ И СТРУКТУРНЫЕ ОСОБЕННОСТИ ОРГАНИЗАЦИИ СИТУАЦИОННО-ОРИЕНТИРОВАННЫХ БАЗ ДАННЫХ

Типичная архитектура базы данных представлена набором из двух обязательных компонентов – метаданных и собственно данных.

Данные – это информация, которая накапливается и хранится в базе данных. По одному из определений [3], данные представляют собой заданные факты, из которых можно логически получить другие данные. Иными словами, это зафиксированная в базе данных информация, на основе запросов к которой можно получить какую-то новую информацию.

Метаданные – это данные более высокого уровня, которые являются описанием структуры данных, хранящихся в базе, «данные о данных». К ним относятся вспомогательные, справочно-реферативные сведения, информация о происхождении данных. Способ представления как данных, так и метаданных определяется используемой базой моделью данных. Так, к примеру, в реляционной базе данных метаданные имеют реляционную структуру и содержат описание таблиц, столбцов, индексов, ограничений и других компонент, из которых состоит база данных. Сами данные представле-

ны также в форме реляционной структуры – совокупности связанных двумерных таблиц.

Организация данных и метаданных определяет средства описания и обработки информации в базе данных – языки определения и манипулирования данными. В системах управления базами данных должны быть предусмотрены процессоры для обработки запросов на этих языках – процессоры языка обработки данных и языка манипулирования данными.

Языки определения данных (ЯОД) предназначены для описания информации в виде исходной концептуальной схемы и преобразования этого описания в соответствующую объектную форму. Обработка данных – выборка, изменение или удаление данных в базе, или добавление в нее новых данных. Иными словами, это манипулирование данными, хранящимися в базе, с помощью языка манипулирования данными (ЯМД).

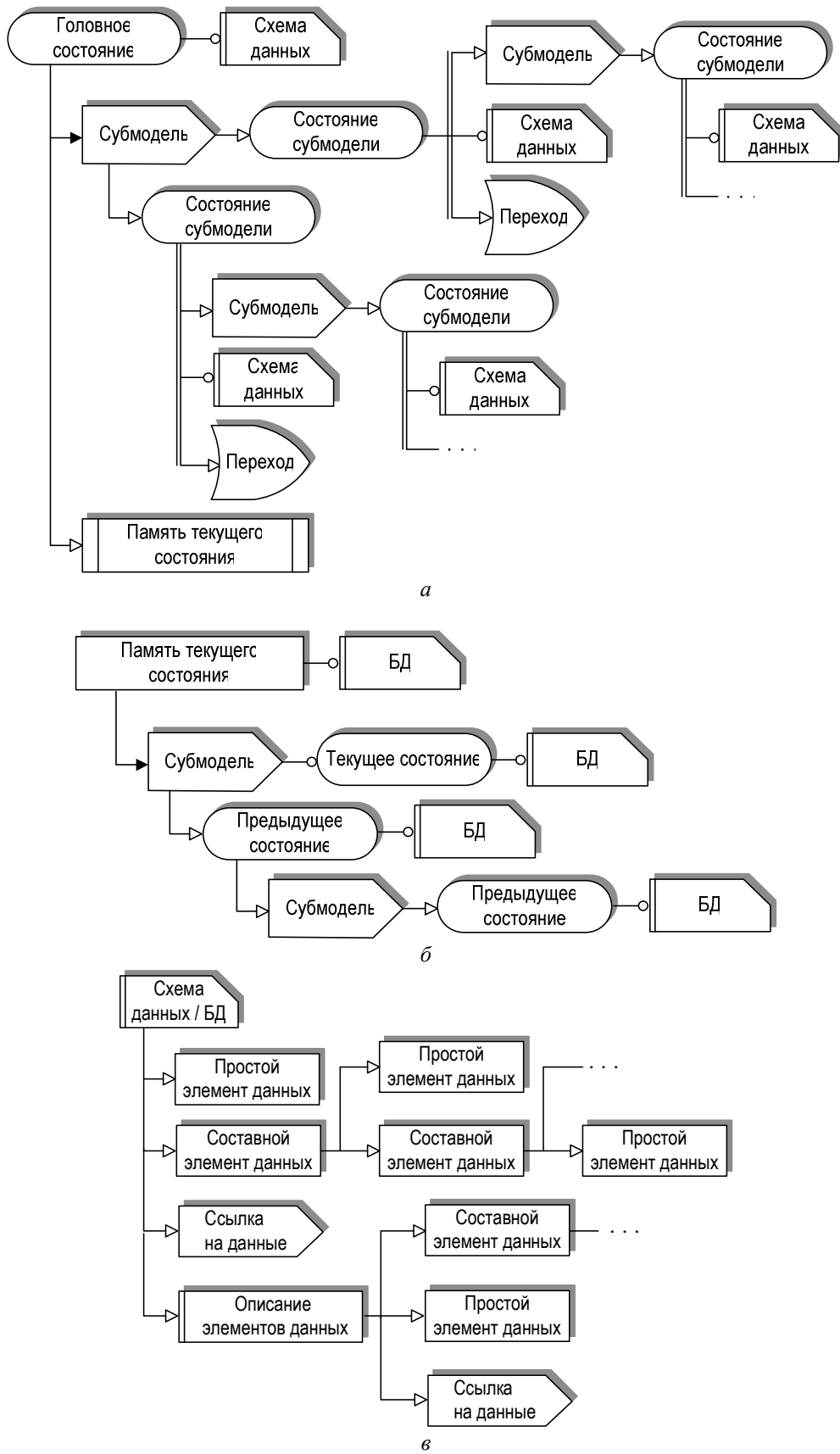
К архитектуре СОБД должны быть предъявлены те же требования, что и к любому другому классу баз данных. В качестве метаданных выступает МКС, которая задает последовательность доступных ситуаций и описывает структуру и состав данных, привязанных к отдельным состояниям. Данные вводятся пользователями при перемещении по состояниям модели и фиксируются в ПТС.

Языки определения и манипулирования данными СОБД выполняют две группы операций – построение МКС и ведение ПТС. Синтаксис этих операций определяется форматом представления СОБД.

Модель класса СОБД рассматривается как совокупность трех моделей (рис. 2):

- исходной динамической модели (МКС), задающей ситуационную составляющую базы данных;
- схемы ассоциированных данных – ее информационной составляющей;
- памяти текущего состояния – пользовательского экземпляра ситуационной модели со значениями данных.

В соответствии с концепцией XML-СОБД каждая из этих составляющих должна иметь объектную XML-структуру. Технологии платформы XML предоставляют широкий спектр возможностей по описанию и манипуляции содержанием и структурой данных, которые могут быть эффективно использованы для работы с СОБД [2].



**Рис. 2.** Архитектура динамической базы данных:  
 а – динамическая модель (МКС); б – память текущего состояния; в – словарь данных / база данных

**Модель конечных состояний.** Модель конечных состояний как ситуационная составляющая базы данных представляет собой совокупность состояний, объединенных в субмодели, связанных друг с другом переходами и содержащих погружения в другие субмодели.

МКС рассматривается как множество экземпляров трех типов объектов – субмоделей, состояний и переходов (рис. 2). Одно из состояний является головным. Оно содержит погружения в субмодели, заданные множеством состояний (состояний субмодели). При этом структура субмодели не зависит от того, к какому типу состояний оно относится – она идентична как для головного, так и для отличных от него состояний. Дополнительными компонентами МКС (которые могут рассматриваться и как самостоятельные структуры) являются словарь данных и память текущего состояния.

Объектная структура языка XML позволяет каждому из типов объектов МКС поставить в соответствие свой XML-элемент, а их свойства представить в виде XML-атрибутов.

Диаграмма XML-элементов МКС приведена на рис. 2, а. На схеме использована нотация, применяемая для концептуального описания моделей баз данных [2] и XML-структур [5].

Элементы МКС являются элементами закрытого типа – их экземпляры не могут содержать атрибуты и вложенные элементы, кроме тех, которые объявлены в модели. На диаграмме свойство закрытости элементов помечено темной тенью за контуром графического представления типа объекта.

На диаграмме показаны три основных свойства каждого типа объекта модели – многозначность, обязательность и порядок следования.

Многозначность элементов МКС показывает допустимое количество их экземпляров. Однозначные элементы, допускающие единственный свой экземпляр в модели, отмечены кружком на конце выносной линии, соединяющей их с родительским элементом, многозначные – стрелкой-треугольником. Так, к примеру, в субмодели все элементы (за исключением словаря данных) являются многозначными.

Среди элементов МКС только субмодели головного состояния являются обязательными, что показано затемненным символом на конце выносной линии, соединяющей графическое представление субмодели с головным состоянием. Необязательность остальных элементов показана с помощью светлых символов на концах выносных линий, соединяющих эти элементы с символом родительского элемента.

Жесткий порядок следования элементов характерен также только для головного состояния. Несущественный порядок следования других элементов указан с помощью параллельных отрезков, группирующих эти элементы в составе их родительского элемента.

**Схема ассоциированных данных.** Схема данных в составе динамической модели задает описание структуры и состава данных, ассоциированных с состояниями МКС.

В схеме данных представлены четыре типа объектов: простые и составные элементы, элементы-ссылки и описания элементов данных. Простыми элементами (по аналогии с простыми индексами в теории баз данных) являются элементы, которые могут содержать только однозначные значения. Составные элементы значений не содержат, но зато включают в себя множество простых элементов данных. Описания элементов данных можно рассматривать как своеобразные справочники – они содержат простые и составные элементы, описывающие некоторый класс объектов заданной предметной области. Один из простых элементов такого справочника играет роль идентификатора – значения, уникальным образом характеризующего каждый его экземпляр. Элементы-ссылки представляют собой значения, ссылающиеся на такие идентификаторы.

Декларативные функции схемы данных дополняются детальным описанием отдельных его элементов. Каждый элемент данных – нечто большее, чем просто название: он может иметь сложную структуру с заданной системой ограничений. Ограничения могут быть как стандартными (многозначность, обязательность, открытость и пр.), так и специфическими. К последним относятся тип данных (числовой, строковый, булевый и пр.), значения по умолчанию, домены и т. д.

Диаграмма элементов схемы данных приведена на рис. 2, б. Здесь использована та же нотация, что и при описании структуры МКС.

XML-реализация схемы данных возможна с помощью технологии XML-схем (XML Schema). Это позволит как описать структуру отдельных элементов, так и задать ограничения на их значения.

Современный уровень развития XML-технологий предоставляет широкий спектр задания схем данных – DTD, XDR, XSD и пр. Любой из этих вариантов применим для описания словаря данных, например, XSD как стандарт консорциума W3C.

Поскольку описание данных входит в состав динамической модели, то имеет место ин-

теграция двух видов разметки – XML и XSD. Фактически динамическая модель уже не может расцениваться как традиционный XML-документ – это XML-документ с «вкраплениями» XSD-кода.

**Память текущего состояния.** Память текущего состояния – единственный компонент СОБД, который несет в себе отличную от системной информацию. Это значения данных, которые пользователь вносит и использует на различных стадиях работы с исходной динамической моделью.

Работа пользователя с СОБД в конечном итоге сводится к ведению экземпляра ПТС, где фиксируются пройденные состояния и хранимые данные, ассоциированные с ними.

ПТС СОБД рассматривается на двух уровнях:

- на макроуровне – отражение смены макросостояний МКС;
- на микроуровне – отражение изменения ассоциированных данных при смене макросостояний.

Структура ПТС представлена тремя типами объектов: корневым элементом, соответствующим головному состоянию МКС; субмоделями и состояниями.

Для каждого состояния (в том числе и для корневого элемента) может быть определен объект типа «база данных» (БД). По своей структуре этот элемент подобен схеме данных – в нем также определены простые и составные элементы, ссылки и описания данных. Разница лишь в том, что объект данного типа – не просто структура, а набор значений, заданных пользователем.

Архитектура ПТС поддерживает свойство темпоральности модели – каждое новое текущее состояние не заменяет предыдущие, что позволяет как отследить последовательность пройденных пользователем шагов, так и выполнить откат для возврата к любому уже пройденному состоянию.

На рис. 2, в приведена диаграмма объектов ПТС, в основе которой лежит описанная выше нотация.

Диаграмма показывает принцип работы ПТС – при обращении пользователя к СОБД формируется корневой элемент с экземплярами субмоделей – непосредственных потомков головного состояния. Корневому элементу ставится в соответствие свой объект типа БД со значениями из заданного набора данных. Первым потомком каждого экземпляра субмодели является элемент, соответствующий текущему состоянию. Кроме того, имя текущего состоя-

ния субмодели задается в атрибуте *current*. Все остальные потомки того же экземпляра – предыдущие текущие состояния, размещенные в обратном порядке добавления. Всем состояниям (и текущим, и нетекущим) ставится в соответствие свой набор значений данных. Это позволяет пользователям работать не только с данными текущего состояния, но и с информацией, введенной на предыдущих шагах. Помимо набора данных состояния содержат погружения в экземпляры субмоделей, по структуре аналогичные одноименным объектам корневого элемента.

С точки зрения XML-реализации очевидно, что ПТС должна быть построена в форме иерархии XML-элементов и атрибутов. Иными словами, для каждого пользователя, обращающегося к XML-СОБД, создается свой XML-документ, в котором описываются пройденные им состояния и введенные данные.

**Адресация.** Важным аспектом структурной модели является адресация элементов данных в пользовательской ПТС.

Для каждого элемента данных в иерархии ПТС может быть задан его «адрес» – выражение, позволяющее обратиться к элементу данных. Ключевым аспектом адресации является «точка отсчета» – узел, от которого отсчитывается выражение адреса. От того, насколько удален искомый узел от контекстного узла, зависит скорость доступа к элементу данных.

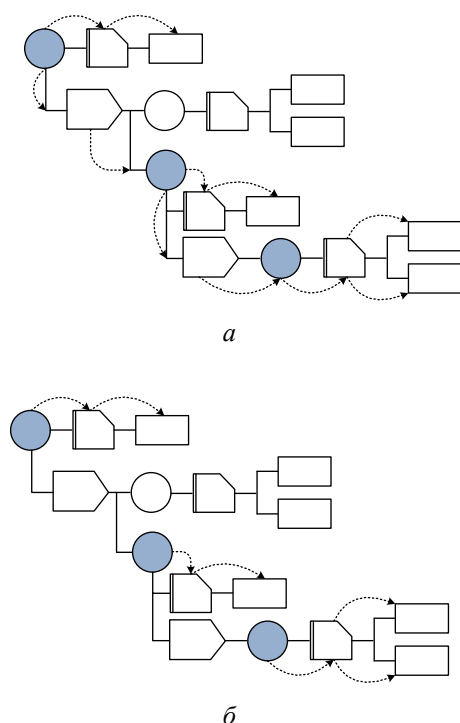
Адресация в составе XML-СОБД предусматривает два варианта. Разница между ними состоит в определении контекстного узла (рис. 3).

Во-первых, XML-база данных как XML-документ допускает традиционную абсолютную XPath-адресацию. Такой адрес включает в себя полный путь к состоянию, с которым этот элемент данных ассоциирован, имя самого элемента и номер его экземпляра в массиве таких же значений. При этом полный путь к состоянию представляет собой последовательное перечисление всех его родительских элементов – от головного элемента ПТС до субмодели, в составе которой определено состояние. Главным недостатком такого способа является его громоздкость – если элемент данных не является дочерним по отношению к потомку первого уровня головного состояния, то доступ к нему будет чрезмерно усложнен (рис. 3, а).

Во-вторых, адресацию можно сделать ситуационно-ориентированной (как и саму базу данных). Это значит, что отслеживание адреса ведется не от самого корня иерархии, а непосредственно от текущих состояний. Для этого



XPath-выражение дополняется логикой отслеживания предикатов – в модели «ищутся» суб-модели с актуальным значением атрибута *siggent* и определяются их текущие состояния. Преимуществом такого подхода по сравнению с предыдущим является простота пути адресации. В качестве его недостатка (пусть и не слишком существенного) можно отметить трудоемкость отслеживания предикатов в выражениях, хотя возможности XPath весьма упрощают этот процесс (рис. 3, б).



**Рис. 3.** Пути адресации в ситуационно-ориентированных базах данных:  
*а* – абсолютная XPath-адресация (контекстный узел – головное состояние);  
*б* – относительная ситуационная XPath-адресация (контекстный узел – текущее состояние)

Кроме того, структура имени элемента определяется его типом. Если элемент данных является непосредственным потомком состояния, то его имя – это просто название элемента. В противном случае, если он входит в состав сложного элемента данных, то его имя – это комбинация из имени этого элемента и всех его родителей.

**Ограничения.** В соответствии с концепцией динамических документов [5] в базе данных со встроенной моделью конечных состояний могут быть выделены следующие группы ограничений:

- структурности – в виде ограничений на форму структуры, к примеру, определенный

тип данных, строгая последовательность дочерних элементов, формат (маска) значений данных и пр.;

- обязательности – в виде условия наличия значений данных;

- ссылочности – в виде наличия для одних данных других с соответствующими значениями. Такие ограничения ориентированы на взаимодействие элементов-ссылок и описаний данных;

- баланса – в виде ограничений на совместные значения элементов данных.

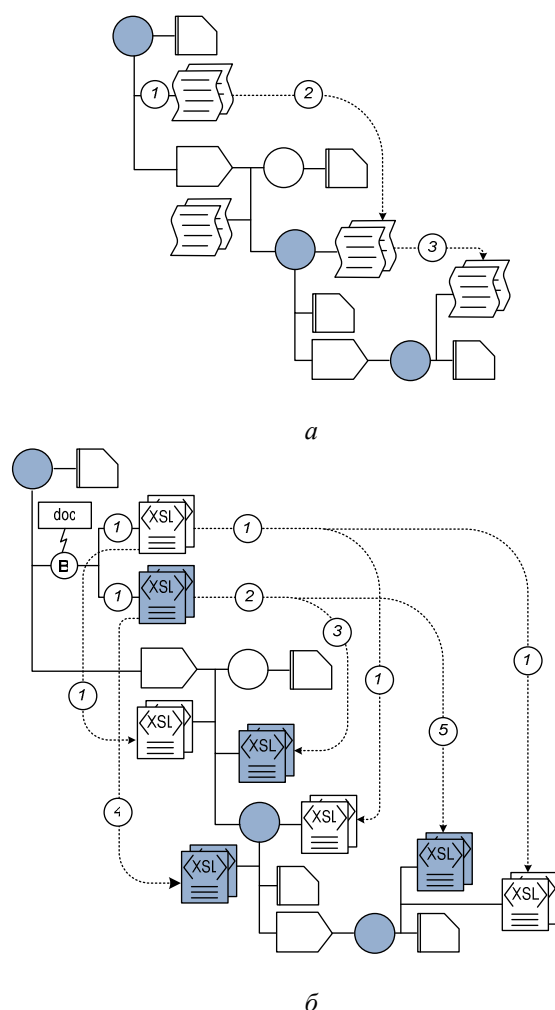
Реализация XML-СОБД ставит целью максимально эффективно использовать XML-технологии в том числе и для ограничений целостности. Первые три группы ограничений могут быть реализованы посредством XML-схем, которые располагают механизмами контроля структуры и обязательности элементов и атрибутов, а также позволяют контролировать ссылочную целостность XML-данных.

Вместе с тем возможностями XML-схем недостаточно для формирования ограничений баланса. Для их реализации должны быть построены некоторые внешние механизмы уровня интерпретатора базы данных. Они могут быть заданы с помощью таких XML-средств, как DOM-объекты, запросы XQuery, XSL-таблицы стилей и пр.

**Модели поведения.** Работа с базой данных не сводится просто к накоплению и хранению информации – различные категории действий пользователей должны быть отражены в виде соответствующей реакции базы данных. Как и в случае с адресацией, возможны два варианта (рис. 4). Их принципиальное различие состоит в том, какие механизмы реализуют поведение базы данных.

Первый подход предполагает использование программных сценариев на языках высокого уровня (рис. 4, *а*). Его основой являются положения, представленные в работах [7, 8] применительно к встраиванию динамических моделей в интернет-приложения. Суть данного подхода заключается в следующем. С состояниями динамической модели ассоциируются фрагменты программного кода. Интерпретатор последовательно обходит все текущие состояния встроенной модели, собирает фрагменты в единый сценарий и запускает его на выполнение. Особенностью подхода является то, что для заданной комбинации текущих состояний предусмотрен единственный вариант сценария, или, иначе, за встроенной моделью закреплен один сценарий (один документ). На рис. 4, *а* приведен пример формирования сценария.

Фрагменты сценариев всех текущих состояний (на рис. они отмечены серой заливкой) последовательно (шаги 1–3) собираются в сценарий.



**Рис. 4.** Поведенческие модели ситуационно-ориентированных баз данных:  
а – сценарии поведения;  
б – спецификации трансформации

Второй подход ориентирован на использование технологий платформы XML, а именно – технологии XSL-трансформации XML-данных.

Суть подхода в следующем. С состояниями субмоделей ассоциированы XSL-шаблоны. С корневым состоянием связано несколько альтернативных шаблонов. Каждый из них является «точкой входа» для генерации разных документов и вызывает шаблоны, связанные с состояниями, которые уже были или в данный момент являются текущими. Интерпретатор «собирает» шаблоны в спецификацию трансформации и запускает преобразование.

Особенностью данного подхода является возможность генерации различных документов

из одного набора данных – пользователь указывает тип документа (параметр doc на рис. 4, б), а интерпретатор при втором проходе вызывает соответствующий XSL-шаблон головного состояния.

Важно отметить, что второй вариант поведения базы данных предусматривает два вида спецификаций трансформации. Первый – ориентированный только на текущие состояния встроенной модели (отмечено серой заливкой на рис. 4, б). Корневой шаблон головного состояния «собирает» только те XSL-шаблоны, которые связаны с текущими состояниями (1–5 на рисунке). Второй вариант определяется тем, что преобразование выполняется применительно ко всей базе данных, независимо от состава ее текущих состояний (отмечено белой заливкой на рис. 4, б). Интерпретатор собирает шаблоны такого типа за один шаг (1' на рис. 4, б).

#### ПРИМЕР СИТУАЦИОННО-ОРИЕНТИРОВАННОЙ БАЗЫ ДАННЫХ

Возможность практического применения концепции СОБД иллюстрируется на примере процесса проектирования технического изделия. Это яркий пример процесса, ориентированного на ситуации. Проектирование можно рассматривать как четкую последовательность этапов – от формулировки требований до разработки технического проекта изделия.

Каждому этапу проектирования может быть поставлен в соответствие свой набор данных – информация, актуальная в той или иной конкретной ситуации. В этих условиях целесообразно организовать представление данных в ситуационно-ориентированной манере, что позволит повысить эффективность взаимодействия с ними. На уровне ситуационно-ориентированных баз данных такая организация сводится к построению исходной динамической модели, отражающей этапы проектирования технического изделия, и нагрузке ее состояний наборами данных. Фрагмент базы данных приведен на рис. 5.

Головное состояние «Проектирование» является наиболее общим – оно отражает сам процесс проектирования технического изделия. Поэтому с этим состоянием ассоциированы такие данные, которые либо незначительно изменяются от этапа к этапу (к примеру, информация об организации-заказчике), либо «известны» проектировщику заранее (название проекта, его описание и т. д.).

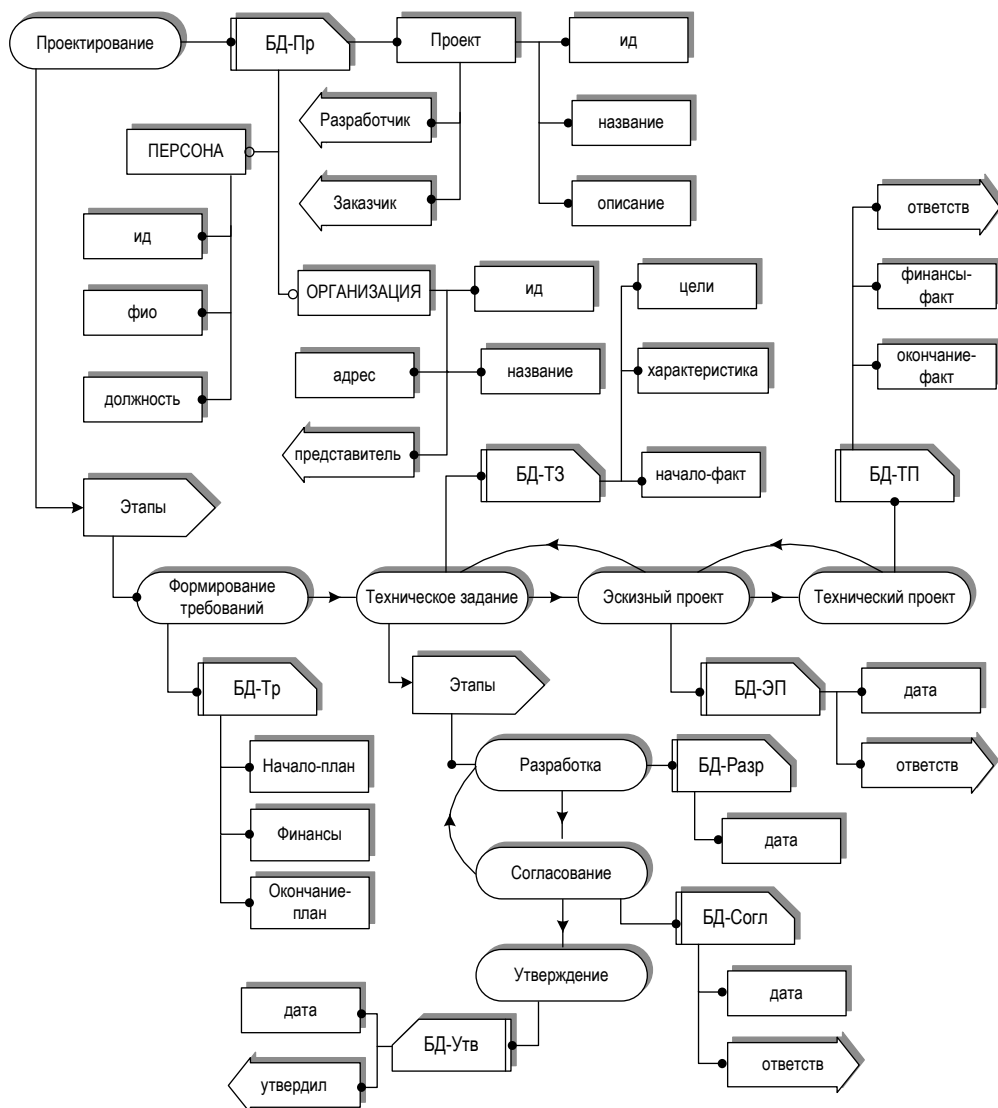


Рис. 5. Фрагмент ситуационно-ориентированной базы данных диссертационного дела соискателя

Этапы проектирования перечислены в одноименной субмодели. Она объединяет четыре состояния, с каждым из которых ассоциирован свой набор данных. Ряд описанных в модели ситуаций могут быть, в свою очередь, также представлены как совокупность некоторых состояний. Так, к примеру, стадия подготовки технического задания может быть задана совокупностью состояний разработки задания, его согласования и утверждения.

Значительная часть модели отводится именно описанию данных, в то время как состояний, с которыми они ассоциированы, существенно меньше. Все наборы данных по своей семантике объединены в группы. Так, группа «Персоны» предназначена для описания информации обо всех лицах, задействованных в подготовке проекта изделия, – от лиц, входящих в команду разработчиков, до лиц, которые

утверждают проект на разных стадиях разработки.

Важно отметить, что в модели представлены элементы-ссылки. Они обозначены стрелками с названием элемента данных внутри. Это позволяет упростить структуру модели и избежать дублирования данных – повторяющиеся данные описываются однократно.

Для повышения наглядности модели различные виды данных представлены по-разному. В названии простых элементов данных первая буква задана в нижнем регистре, а сложных – в верхнем. При этом если сложный элемент данных является необязательным, то весь текст его названия представлен в верхнем регистре. Так, к примеру, название элемента данных «цели» свидетельствует о том, что он простой, а название «ОРГАНИЗАЦИИ» – о том, что он сложный и необязательный.

Вообще говоря, с головным состоянием ассоциировано несколько элементов-описаний. Это справочники, данные из которых в дальнейшем используются в других субмоделях и состояниях исходной модели. Использование подобных справочников позволяет минимизировать избыточность информации: при многократном использовании одних и тех же данных не нужно их дублировать – достаточно просто описать их в справочнике, а потом ссылаться на них.

Использование СОБД при проектировании сводится к манипулированию МКС для формирования экземпляра ПТС. Здесь фиксируются этапы проектирования, а также та информация, которая имеет значение на этих этапах.

### ЗАКЛЮЧЕНИЕ

- Ситуационно-ориентированные базы данных представляют собой базы данных со встроенными динамическими моделями.

- Ключевым преимуществом ситуационно-ориентированных баз данных является возможность взаимодействия с данными в контексте ситуации, развитие которой описывается посредством модели конечных состояний.

- Архитектура ситуационно-ориентированной базы данных представлена тремя компонентами – исходной динамической моделью, задающей последовательность возможных ситуаций в формате модели конечных состояний, схемой данных, регламентирующей структуру данных, ассоциированных с состояниями модели, и памятью текущего состояния, определяющей пользовательский экземпляр динамической базы данных.

- XML-реализация архитектуры ситуационно-ориентированной базы данных направлена на представление описательных данных в форме XML-разметки, а регламентирующих – в виде XML-схем. В этой связи исходная динамическая модель и память текущего состояния – документы XML, а словарь данных – XML-схема (например, XSD).

### СПИСОК ЛИТЕРАТУРЫ

1. Юсупова Н. И. Критические ситуации и принятие решений при управлении в условиях помех: монография. Уфа: Гилем, 1997. 112 с.
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. Дейт К. Дж. Введение в системы баз данных. М.: Вильямс, 2001. 1072 с.
4. Миронов В. В., Шакирова Г. Р. Электронные документы со встроенной динамической моде-

лью на основе XML: монография. Уфа: УГАТУ, 2009. 179 с.

5. Миронов В. В., Юсупова Н. И., Шакирова Г. Р. XML-технологии в электронных документах. Документы Word: учеб. пособие. Уфа: УГАТУ, 2009. 208 с.

6. 12 правил Кодда [Электронный ресурс] ([http://ru.wikipedia.org/wiki/12\\_правил\\_Кодда](http://ru.wikipedia.org/wiki/12_правил_Кодда)).

7. 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.

8. Миронов В. В., Маликова К. Э. Интернет-приложения на основе встроенных динамических моделей: архитектура, структура данных, интерпретация // Вестник УГАТУ. 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**, ст. преп. 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.