

Шибанов Сергей Владимирович, Лысенко Эдуард Владимирович

КОНЦЕПТУАЛЬНЫЕ ОСОБЕННОСТИ СИСТЕМ УПРАВЛЕНИЯ АКТИВНЫМИ БАЗАМИ ДАННЫХ

Адрес статьи: www.gramota.net/materials/1/2010/11-2/35.html

Статья опубликована в авторской редакции и отражает точку зрения автора(ов) по рассматриваемому вопросу.

Источник

Альманах современной науки и образования

Тамбов: Грамота, 2010. № 11 (42): в 2-х ч. Ч. II. С. 106-114. ISSN 1993-5552.

Адрес журнала: www.gramota.net/editions/1.html

Содержание данного номера журнала: www.gramota.net/materials/1/2010/11-2/

© Издательство "Грамота"

Информация о возможности публикации статей в журнале размещена на Интернет сайте издательства: www.gramota.net

Вопросы, связанные с публикациями научных материалов, редакция просит направлять на адрес: almanac@gramota.net

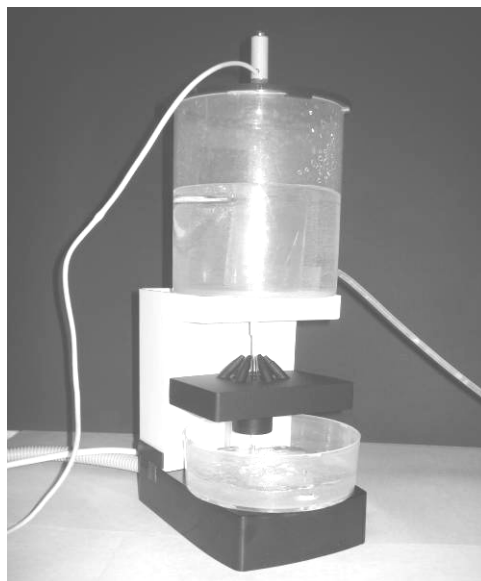


Рис. 4. Лабораторный макет бесконтактного мутномера

На Рис. 4 изображен действующий лабораторный макет описанного мутномера.

Список литературы

1. Андреев В. С., Попечителев Е. П. Лабораторные приборы для исследования жидких сред. Л.: Машиностроение, 1981.
2. Беляков В. Л. Автоматизация промышленной подготовки нефти и воды. М.: Недра, 1988.
3. Патент РФ № 2235991 на изобр. бесконтактный мутномер / В. С. Фетисов. Оpubл. 2004. Бюл. изобр. № 25.
4. Патент США № 5489977. НКИ 356/73. МПК G01B 21/00. Оpubл. 06.02.1996.
5. Фетисов В. С. Фотометрические полевые средства измерений концентрации жидких дисперсных систем / Уфимск. гос. авиац. техн. ун-т. Уфа: УГАТУ, 2005. 233 с.

УДК 681.3.01

*Сергей Владимирович Шибанов, Эдуард Владимирович Лысенко
Пензенский государственный университет*

КОНЦЕПТУАЛЬНЫЕ ОСОБЕННОСТИ СИСТЕМ УПРАВЛЕНИЯ АКТИВНЫМИ БАЗАМИ ДАННЫХ[©]

Введение. В общем понимании система управления базами данных (СУБД) - это система, которая позволяет хранить большой объем данных в соответствии с некоторой моделью данных, обращаться к ним и модифицировать их, соблюдая при этом целостность данных. Функционал СУБД предоставляется для разработчиков и пользователей через унифицированный интерфейс с использованием языка запросов - языка SQL для реляционных СУБД (РСУБД) или объектного языка запросов для объектно-ориентированных СУБД (ОСУБД).

Традиционные СУБД пассивны в том смысле, что действия, выполняемые СУБД, инициируются пользователями или программными приложениями. Данный подход в некоторых ситуациях оказывается неэффективным. Рассмотрим пример. Пусть в базе данных (БД) информационной системы (ИС) железнодорожной компании хранятся данные о поездах, расписаниях, местах, стоимости и пр., которые доступны через различные терминалы [21]. При некоторых обстоятельствах, например, в праздничные и предпраздничные дни, дни проведения культурных мероприятий необходимо добавить дополнительные вагоны, если число запасных мест на ближайший месяц ниже некоторого порогового значения. Для администратора пассивной БД, который должен обеспечить поддержку данного требования, есть два возможных варианта:

- добавить дополнительную контролирующую функциональность во все программы продажи билетов, чтобы вышеописанная ситуация проверялась при каждой продаже;
- реализовать опрашивающий механизм, который периодически проверяет количество свободных мест.

Применение первого подхода приводит к тому, что семантика задачи контроля будет распределена, дублирована и скрыта в различных клиентских приложениях. В отличие от первого подхода, в случае применения опрашивающего механизма, семантика приложения реализована в одном месте, но трудности появляются при определении наиболее подходящей частоты опроса. При слишком высокой частоте опроса страдает стоимость системы, при слишком низкой частоте опроса, реакция может быть запоздалой - например, вагон добавляется, но только после того, как нескольким покупателям было отказано из-за нехватки мест.

Приведенный выше пример не является единственным и узкоспециальным, как может показаться на первый взгляд. Сходные проблемы возникают также в информационных системах электронной коммерции. Особенно остро они ощущаются, когда БД становится распределенной, и взаимодействовать компоненты ИС должны через Интернет [18]. Клиентов в таких информационных системах гораздо больше, чем в предыдущем примере, связь с ними часто менее надежна, значит и опрос - долгая, а иногда и требовательная по производительности процедура. С другой стороны, если функциональность содержится в клиентских приложениях, они не только становятся громоздкими, но их как-то необходимо обновлять, следя за тем, чтобы у всех была актуальная версия.

Аналогичные проблемы проявляются и в системах мониторинга, контроля и планирования. Речь идет о более узкоспециальных и разнотипных системах, таких как системы планирования и управления рабочим процессом, мониторинг физических процессов и так далее [24]. Эти задачи требуют своевременного обнаружения определенных, часто критических, ситуаций и своевременной реакции на них, а значит, необходима высокая адаптивность всей системы.

Всё большее распространение получают системы управления движущимися объектами (Moving Objects Database - MODB) [26]. Объекты изначально не статичны и обладают поведением, перемещаясь по различным маршрутам, меняя свои характеристики и влияя на характеристики маршрутов. К таким системам, в частности, можно отнести системы, позволяющие моделировать и прогнозировать пробки и аварии, проектировать транспортные пути и автоматизировать управление трафиком. Также можно выделить и однопользовательские транспортные задачи, например поиск оптимального маршрута с точки зрения расстояния и наличия пробок. В таких системах различных ситуаций возникает много, а на них нужно реагировать всё так же быстро, часто в реальном времени. Кроме того, появляется проблема описания поведения объектов. В терминах традиционных СУБД некоторые аспекты поведения движущихся объектов описать сложно или невозможно, например, не все источники событий могут быть описаны традиционными средствами.

Похожими на мониторинг трафика задачами являются моделирование сражений, контроль перемещения животных и птиц в экологии и, конечно же, средства мобильной коммуникации и другие.

Таким образом, задачи, которые ставятся перед современными СУБД, обслуживающими различные информационные системы, требуют, чтобы информационная система могла активно реагировать на события, происходящие как внутри нее, так и за ее пределами. Этому способствует и то, что коммерческие предприятия и большие транснациональные компании, которые являются основными потребителями разработок в области управления данными, все чаще переходят к новым структурам управления. Такие структуры управления отличаются тем, что создаются отдельные бизнес единицы, у которых возрастает степень свободы от бюрократического аппарата и тем самым уменьшается время принятия решений и ускоряется реагирование на изменения на рынке. Но компания не может изменить свою структуру, не внося соответствующих изменений в информационные системы. Одним из возможных решений в данном случае является применение активных баз данных [18; 24].

Цели и задачи. Активные базы данных (АБД) поддерживают механизмы, позволяющие автоматически реагировать на события, которые происходят как внутри, так и вне информационной системы. Семантика поведения (события, условия, реакции) централизована в самой АБД и обрабатывается (контролируется, исполняется) системой управления АБД (СУАБД). При этом СУАБД должна обеспечивать поддержку модели знаний - механизма описания и модели выполнения - механизма времени выполнения, чтобы реализовать поведение в виде реакций на те или иные ситуации [21].

Механизмы АБД могут быть использованы для того, чтобы описать и реализовать внутреннее поведение информационной системы без ссылок на внешние устройства или другие системы. Например, для описания восстановительных действий в моделируемой БД, для контроля продаж в базе данных, для распространения загрузочных вычислений в базе данных архитектурного дизайна или для прогнозирования активности рынка в базе данных управления ценными бумагами [24].

Механизмы АБД могут быть использованы в связке с контролируемыми устройствами, чтобы записывать и реагировать на ситуации вне информационной системы. Например, в приложениях управления и контроля меняющихся сценариев военных действий [5], в медицинских приложениях для предупреждения врачей о состоянии пациента [Ibidem], в транспортных приложениях для прогноза пробок и в контроле воздушных сообщений для обнаружения потенциально опасных маршрутов и предупреждении диспетчера о сближении самолетов [23].

Кроме того, механизмы АБД также могут быть использованы как средства реализации внутренних базовых задач СУБД. Например, для поддержания ограничений целостности [7; 25], материализованных представлений [25; 27], производных данных [15], координировании распределенных вычислений [8; 11], модели транзакций [16], и автоматическое обновление экрана в контексте изменения БД [13; 14; 23].

Обобщая разнообразие применения механизма активных БД, можно сформулировать типичную задачу, для решения которой желательно применение АБД и СУАБД. Имеется большая база элементарных фактов, описывающих состояние наблюдаемого объекта и изменяющаяся во времени, благодаря поступающим данным (в том числе, и извне). Информационная система должна обнаруживать некоторое, возможно, большое число задаваемых ситуаций и автоматически реагировать на их возникновение задаваемыми действиями.

На данный момент активные базы данных является интенсивно развивающейся областью исследований в технологии баз данных и систем управления базами данных [18; 24]. Фирмы-разработчики современных систем управления базами данных часто позиционируют свои системы, либо как поддерживающие отдельные функции СУАБД, либо полностью относящиеся к классу СУАБД. При этом не всегда понятно, какими свойствами должна обладать СУБД, чтобы ее можно было бы назвать активной, как должна проявляться в процессе функционирования системы ее «активность».

Цель данной статьи - выделить основные особенности СУАБД, определить их отличия от «пассивных» СУБД. Кроме того, среди выделенных особенностей СУАБД определить какие из них являются обязательными для реализации, а какие желательными. Также делается попытка классифицировать системы управления АБД.

Модель знаний и модель выполнения АБД. Классическая модель знаний активной базы данных определяет принципы построения активных правил и базируется на трех элементах: событие, условие и действие (Event, Condition, Action).

Правила типа «событие - условие - действие» (ЕСА-правила) включают в себя информацию о внешнем или внутреннем событии, условии для дальнейшего продолжения транзакции и конечном действии при выполнении условия. На основе набора ЕСА-правил происходит мониторинг внешних и внутренних событий. Если система мониторинга обнаруживает событие, которое описано в правиле, то запускается правило, соответствующее данному событию. Если событие описано в нескольких правилах, то запускаются все правила для этого события.

Классическая ЕСА-модель описывает реакцию объекта на внешние события, но не учитывает, что в реальных приложениях поведение объекта часто зависит от его состояния. Например, учетная запись пользователя может быть «активна» или «заблокирована», и в зависимости от этого различна реакция на попытку входа пользователя в систему. В системе документооборота документ может находиться в состояниях «свободен» и «редактируется». Если документ свободен, то любой пользователь может начать его редактирование, при этом документ переводится в состояние «редактируется». После этого остальные пользователи не имеют доступа к документу, пока не завершится его редактирование, и он не вернется в состояние «свободен».

Расширенная модель активной базы данных строится с учетом состояния объектов. В такой модели правила задаются четверкой элементов состояние объекта, событие, условие и действие (State, Event, Condition, Action). Состояние объекта - это абстракция значений и связей объекта. Множество значений и связей группируются в состояние в соответствии с массовым поведением объекта. Состояние описывает отклик объекта на получаемые события таким образом, что в конкретном состоянии игнорируются любые события, за исключением тех, поведение при получении которых описано явным образом. Введение в модель состояний позволяет наделить объекты свойством изменчивости - способностью изменять свое поведение в зависимости от возникающих событий. Таким образом, расширенная SECA-модель описывает реакцию объекта на внешние события с учетом его внутреннего состояния.

Кроме полных правил, что означает, что в правиле описано событие, условие и действие, существует несколько вариантов их модификации:

- безусловное правило (ЕА-правило);
- правило с неявным событием (СА-правило).

Безусловное правило (ЕА-правило) - правило с опущенным условием, т. е., такое правило, которое содержит в себе только событие и действие. Данный тип правил поддерживает события, реакция на которые обязательна к исполнению. Поэтому нет необходимости осуществлять дополнительную операцию по проверке условия, а можно сразу же перейти к непосредственному действию, которое соответствует данному событию. Возможен вариант ЕА-правила, когда условие является частью действия.

Правило с неявным событием (СА-правило) - правило, в котором описано только условие и действие. Может возникнуть ситуация, когда сама СУАБД или ее компилятор определяют событие. Например, изменение данных, когда определены только ограничения, которые накладываются на изменение данных и метод их восстановления, при некорректном вводе. В этом случае система использует внутренние сигналы, сообщающие, что данные были изменены, и необходимо их проверить на соответствующие ограничения и, при необходимости, восстановить.

В таких случаях говорится, что событие определенно неявно - пользователь определяет лишь условие и действие на события такого рода, а СУАБД определяет событие автоматически. При этом СУАБД должна предоставлять возможность пользователю определять такого рода правил через соответствующий интерфейс.

По аналогии с языком описания данных в традиционных системах управления базами данных СУАБД должна предоставлять язык для определения ЕСА-правил или SECA-правил в зависимости от поддерживаемой модели, который должен включать в себя конструкторы для определения состояний, событий, условий и действий.

Также в СУАБД должен присутствовать модуль выполнения правил, поддерживающий исполняющую модель, которая определяет когда и как должно выполняться правило [19]. Модель выполнения описывает то, как набор правил обрабатывается во время выполнения, и определяет режим связывания условий и действий, масштаб транзакций, политику сетевого эффекта, цикловую политику, приоритеты, планирование активности, управление ошибками и т.д.

Концептуальные особенности СУАБД. В данной части речь пойдет о характерных особенностях СУАБД, которые позволяют с одной стороны отнести ту или иную систему управления базами данных к классу СУАБД, а с другой, оценить возможности конкретной СУАБД. Концептуальные особенности СУАБД разобьем на три группы, в соответствии с которыми и будет происходить их рассмотрение:

- 1) возможности СУАБД по определению правил;
- 2) возможности СУАБД по выполнению правил;
- 3) особенности реализации и применения СУАБД.

Рассмотрим концептуальные особенности СУАБД в пределах каждой группы.

1. Определение правил. Основные возможности СУАБД по определению правил сведены в Табл. 1.

1.1. Функциональные возможности, свойственные пассивным СУБД. СУАБД должны иметь в себе все функциональные возможности, которые предоставляют «пассивные» СУБД, например, хранение, восстановление, модификацию данных. Это означает, что, если пользователь не использует функциональные возможности СУАБД, то система используется в качестве «пассивной» СУБД.

1.2. Поддержка модели правил. Отличительной особенностью СУАБД является то, что система такого рода поддерживает возможность обратной реакции на внешние и внутренние события. То, как система должно отреагировать на то или иное событие, должно быть определено пользователем. Стандартные для «пассивных» систем интерфейсы для определения типа данных расширяются за счет дополнительных возможностей по определению правил. Использование такого рода правил предполагает, что СУАБД должна предоставлять инструменты для определения типов события, методы для определения условий и действий.

Определение типов событий. Тип события (описание события, образец), описывает ситуации, на которые должна реагировать АБД. События могут быть как простыми, так и сложными. К простым событиям относятся базовые события, которые возникают при работе с базой данных, например, обращение к данным, изменения кортежей данных и пр. Сложное событие состоит из двух и более простых или сложных событий.

Табл. 1. Группа особенностей СУАБД «Определение правил»

№ п/п	Особенность СУАБД		Описание
1.1	СУАБД есть просто СУБД		Наличие функциональных возможностей, характерных для «пассивных» СУБД
1.2	Поддержка модели правил	Определение типов событий	Наличие инструментов для определения различных типов событий, как простых, так и сложных (состоящих из нескольких простых или сложных событий), отслеживания возникновения событий и определения дополнительных параметров для событий
		Определение условий	Наличие инструментов для определения условия выполнения правил
		Определение действия	Под действие понимается реакция на событие. Наличие инструментов для определения для определения действий, как возможных реакций на наступление событий
1.3	Наличие средств управления базами правил и их эволюции	Управление базами правил	База правил, содержащая определенный набор правил, должна быть представлена в виде метаинформации и должна быть доступна пользователям и приложениям
		Поддержка эволюции баз правил	Возможность изменения событий, условий, действий в существующих правилах
		Возможность активировать или деактивировать правила	Одно или несколько правил могут быть деактивированы, и не будут учитываться при возникновении событий. И наоборот, одно или несколько деактивированных правил могут быть активированы, после чего они станут доступны при обработке событий

Для сложных событий метка времени определяется на основании меток времени всех входящих в него событий, например, это может быть момент времени, когда выполнилось последнее входящее в него событие. Также для сложных событий должны быть сформулированы ограничения, чтобы можно было сформировать корректную композицию простых событий. Одним из способов определения таких ограничений является предикат на параметрах события (например, метка времени), но возможны и более общие предикаты. Например, можно требовать для сложного события, чтобы все его элементарные события произошли в пределах той же самой транзакции, или все операции по обращению к данным обращались к одной области данных.

В АБД могут присутствовать различные типы событий. Например, события «до» и события «после». Событие «до» характеризуется тем, что сигнал о нем подается до, того как само событие начнет выполняться. Событие «после» подразумевает, что сигнал о нем подается после того, как оно выполнилось.

Также возможны дополнительные параметры для события, например, идентификатор пользователя или процесса, инициировавшего событие.

Определение условий. Условие определяет, в каком состоянии должны находиться критические части база данных, чтобы выполнить действие из соответствующего правила. Условие может быть простым предикатом SQL-запросов, например, «where» или же запросом к базе данных, возвращающим пустое или непустое множество записей. Условие выполняется, если принимает значение истины или выбранное множество записей является непустым.

Определение действий. Под действием понимается реакция на событие, которая выполняется после того, как выполнится правило, вызвавшее данное действие. Действием может быть изменение данных, получения данных, а также прекращение выполнения транзакции и пр.

1.3. Средства для управления базами правил и их эволюции. СУАБД должна обеспечивать управление базой правил как метainформацией, поддержку эволюции правил, возможность активации и деактивации правил.

Управления базой правил как метainформацией. Набор правил, определенный в АБД, хранится в базе правил. Базой правил должна управлять СУАБД. Другими словами, определение правил есть часть системы управления базами данных, метainформацией и самой базы данных. СУАБД должна хранить информацию о том, какие в настоящее время существуют правила и как они определены. База правил должна быть доступна пользователям и приложениям с учетом их прав.

Предпочтительно, чтобы СУАБД представляла метainформацию относительно правил в рамках основной модели данных как часть системного каталога. В этом случае, пользователи и приложения могут получить информацию о правилах из каталога таким же образом, как может быть получена информация о «пассивной» схеме - отношениях, классах, и т.д.

Поддержка эволюции базы правил. База правил должна предоставлять возможность для поддержки эволюции правил в течение времени. При этом недостаточно поддерживать только набор ограниченных правил, и, тем более, не уместно внедрять правила в код системы управления базами данных. СУАБД должна позволять добавлять новые правила и удалять старые, изменять состояния объектов, для которых определены правила, события, условия, или действия уже существующих правил.

Возможность активировать или деактивировать правила. Эта особенность отражает то, что в СУАБД правила могут быть активированы и деактивированы. Это означает, что одно или несколько ранее определенных правил могут быть деактивированы, и не будут учитываться в дальнейшем при возникновении событий. И наоборот, активировать одно или несколько правил означает, что эти, до этого момента деактивированные правила, снова становятся активными и используются при обработке возникающих событий.

2. Выполнение правил. Основные возможности СУАБД по выполнению правил сведены в Табл. 2.

2.1. СУАБД должна включать в себя исполняющий модуль, обеспечивающий обнаружение возникновения событий, поддержку режима связывания правил и данных, проверку выполнения условий, выполнение соответствующих действий.

Обнаружение возникновения событий. В идеальном случае от СУАБД ожидают, что система сама автоматически будет определять возникновение всех видов событий. Возможно, что некоторые виды событий не определяются СУАБД автоматически, и сигналы об их возникновении передаются непосредственно из приложения или от пользователя.

Если в СУАБД пользователь или программист отвечают за передачу корректных сигналов о всех видах возникающих событий, то такая система является синтаксическим вариантом пассивной системы управления базами данных. Тем не менее, в концепцию СУАБД вписывается, что у пользователя есть возможность непосредственно сигнализировать системе о возникновении события.

Виды связывания событий и данных. Часто, возникшее событие связано с какой-либо частью данных, по отношению к которым оно и произошло. Вид связывания определяет детализацию событий, а также части данных, к которым может обратиться СУАБД при проверке условия и выполнения действия, инициированных событием.

Существует несколько видов связывания событий и данных [22]:

- *связывание, ориентированное на исключение.* Применение данного вида связывания означает, что имеются отдельные исключения, и кортежи данных или какой-либо объект, для которых произошло событие, могут быть переданы для дальнейшей проверки условия и выполнения действия. Если же событие является сложным и содержит в себе несколько событий, то связанное правило выполняется для каждого события отдельно;

– *связывание, ориентированное на общность*. Данный вид связывания подразумевает, что возникновение события ассоциируется с набором исключений, для которых это самое событие и возникло. В таком случае ассоциативное правило выполниться только один раз для всех исключений, с которыми связано данное событие. При этом условие и действие могут обратиться ко всем исключениям, находящимся в наборе.

В обоих случаях, если связывание является предшествующим, то условие может также обратиться к состоянию исключений, в котором они находились до возникновения данного события.

Табл. 2. Группа особенностей СУАБД «Выполнение правил»

№ п/п	Особенность СУАБД		Описание
2.1	Наличие исполняющего модуля	Обнаружение возникновения события	Автоматически отслеживается возникновение всех поддерживаемых видов событий, сигналы о возникновении определенных событий передаются от приложений или пользователей
		Поддержка режима связывания	Возникшее событие может быть связано с какой-либо частью данных, которые хранятся в АБД, и по отношению к которым событие произошло
		Механизм проверки условий	Возможность проверки условия, которое следует за тем, как обнаружено событие
		Механизм выполнения действий	Возможность выполнить действия после того, как было обнаружено событие и выполнено соответствующее условие, а также передать информацию от события и условия к непосредственному действию
2.2	Методы сцепления транзакций		Поддержка базовых типов сцепления транзакций: немедленное, отсроченное, расцепленное
2.3	Способы выявления сложных событий		Поддержка сложных событий требует наличия механизма выявления таких событий
2.4	Управление историей событий		История события хранит сведения о всех возникших событиях заданных типов и требуется для анализа сложных событий, которые произошли во время различных сессий работы приложений или транзакций
2.5	Механизмы разрешения коллизий		При возникновении многочисленного количества событий, которые должны быть выполнены в один момент времени, СУАБД должна определить будут ли транзакции, вызванные такими действиями, выполнены последовательно друг за другом или же эти транзакции могут быть выполнены параллельно

Механизм проверки условия. СУАБД должна предоставлять возможность проверки условия, которое следует за тем, как обнаружено событие. В практическом смысле это означает, что необходимо передать информацию от события к модулю проверки условия, для последующего вычисления и проверки на выполнение условия. Если событие произошло для отдельного объекта или ряда кортежей в отношении, то должно иметься возможность обращения к этой информации в условии.

Механизм выполнения действий. СУАБД должна быть в состоянии выполнить действия после того, как было обнаружено событие и выполнено соответствующее условие. Также должна иметься возможность передачи информации от события и условия к непосредственному действию, например, информация относительно объекта, для которого условие соблюдается.

Дополнительно к этому должна быть предоставлена возможность выполнения действия, как части выполняемой транзакции, и как только она выполнится, совершенное действие необходимо проверить на соответствие ограничениям, наложенным на данные, и при необходимости осуществить восстановление предыдущего состояния данных.

2.2. Методы сцепления транзакций. Отношения между порожденной транзакцией и порождающей транзакцией в СУАБД обычно называют *методом сцепления транзакций*. Первоначально были предложены следующие виды сцепления [17]:

– *немедленный* - порожденная транзакция выполняется непосредственно после того, как поступил сигнал о событии;

- *отсроченный* - порожденная транзакция выполняется в конце порождающей транзакции, но прежде, чем та завершится;
- *расцепленный* - порожденная транзакция выполняется как отдельная транзакция.

Для первых двух видов сцепления (немедленные и отсроченный) порожденные транзакции по существу являются вложенными транзакциями (подтранзакциями) порождающей транзакции. Например, аварийное завершение работы относится к немедленному виду сцепления транзакций и не может существовать в расцепленном виде.

Для третьего вида сцепления (расцепленный) порожденная транзакция может быть начата непосредственно после передачи сигнала о событии и не зависит от порождающей транзакции. Расцепленная транзакция может завершиться или прерваться независимо от порождающей ее транзакции.

В последующих исследованиях были предложены другие методы сцепления транзакций. Например, в [25], расцепленный метод разделен на два подвида: *причинно зависимое сцепление* и *причинно независимое сцепление* транзакций.

В [6] были предложены еще два новых подвида причинно зависимого сцепления. *Последовательный причинно зависимый способ сцепления* предполагает, что вызванная транзакция может только начаться после того, как транзакция вызова завершилась. В *исключительном причинно-зависимом способе* вызванная транзакция может начаться, только если транзакция вызова потерпела неудачу.

2.3. Методы выявления сложных событий. Поддержка сложных событий, которые состоят из двух и более простых событий, предполагает наличие в СУАБД механизма выявления такого рода событий.

Механизм выявления сложных событий определяет, какие составляющие события относятся к сложному событию, в которое оно входит, и как параметры сложного события вычисляются из параметров, входящих в него простых событий. Для различных классов приложений могут потребоваться разные методы выявления сложных событий, например, такие, как «недавний», «хроника», «непрерывный» и «совокупный» [23]. В СУАБД может быть жестко зафиксирована стратегия выявления сложных событий, либо выбор должен быть предоставлен пользователю.

2.4. Управление историей событий. История события включает сведения о возникновении всех событий заданных типов, включая компоненты сложных событий. Такая история событий должна поддерживаться в любой активной системе баз данных. История событий начинается в пункте времени, когда первое событие обнаружено. История может продлиться свыше одной сессии и более чем нескольких транзакций. Таким образом, постоянная история события требуется всякий раз, когда возникает необходимость сигнализировать о сложных событиях, основанных на событиях, что произошли во время различных сессий приложений или транзакций. Задача минимум состоит в том, что история события хранит только те события, которые могут понадобиться для анализа сложного события.

СУАБД может иметь в своем составе специальные средства конструирования правил, поддерживающие нескольких основных типов событий и позволяющие определить, как долго сведения о возникновении событий должны сохраняться. В этом случае, история событий может использоваться позднее и для других задач, таких как контроль и трассировка правил.

2.5. Механизмы разрешения коллизий. Может возникнуть такая ситуация, что большое число событий могут быть выявлены в один и тот же момент времени. При возникновении такой ситуации СУАБД должна определить будут ли транзакции, вызванные реакцией на возникшие события, выполнены последовательно друг за другом или же эти транзакции могут быть выполнены параллельно друг с другом.

Разрешение коллизий, зачастую, зависит от семантики правил, которая известна только пользователю. Это подразумевает, что у правила должен быть определен спецификатор, который дает возможность определить путь разрешения коллизий, например, посредством приоритетов. Если пользователем не был определен такого рода спецификатор, то СУАБД должна иметь собственный механизм для определения в какой последовательности будут выполнены действия, связанные с наступившими событиями, а реализация механизма приоритетов - это всего лишь один из вариантов решения данной проблемы [26].

3. Особенности реализации и применения СУАБД. СУАБД должна предоставлять разработчикам и пользователям соответствующий набор средств и инструментов для практического использования. Особенности реализации и применение СУАБД сведены в Табл. 3.

3.1. Программная окружающая среда. Само собой разумеется, что СУАБД должна быть пригодной к использованию. В практике простота и удобство в использовании означает, что система должна иметь формальный язык определения правил.

Для дополнительного удобства использования разработчиками и пользователями предполагается также дополнительный набор инструментов:

- браузер правил;
- конструктор правил;
- анализатор базы правил;
- отладчик правил;
- средство трассировки правил;
- сервисные средства.

Это может быть специальный набор инструментов, предназначенный для использования в СУАБД, или же расширение средств существующих традиционных систем управления базами данных.

3.2. Настройка СУАБД. Для обеспечения гибкости проектирования, разработки и функционирования СУАБД предлагается трехуровневая модель [25], аналогичная трехуровневой модели проектирования традиционных баз данных. На внешнем уровне каждый пользователь или приложение определяют собственные правила, которые необходимы им для работы с системой. Концептуальный уровень содержит в себе все правила, которые были определены всеми пользователями и приложениями системы. На внутреннем уровне находятся конкретные детали внутреннего представления и выполнения правил. Также на внутреннем уровне должны определяться способы оптимизации выполнения правил.

Данный подход обеспечивает полную методологию проектирования, разработки и применения активных правил в СУАБД.

Классификация СУАБД. Различные особенности, о которых речь шла выше, оставляют некоторый выбор разработчикам СУАБД и, следовательно, некоторую неопределенность в том, как объединить выше упомянутые особенности в СУАБД. Другими словами, есть различные пути решения того, как «связать» их функциональные возможности в рабочем варианте СУАБД.

Табл. 3. *Группа особенностей СУАБД «Особенности реализации и применения»*

№ п/п	Особенность СУАБД	Описание
3.1	Поддержка программной окружающей среды	Наличие формального языка определения правил, браузера правил, проектировщика правил, анализатора базы правил, отладчика, инструментов обслуживания, средств трассировки
3.2	Настройка СУАБД	Для обеспечения гибкости проектирования, разработки и функционирования СУАБД предлагается трехуровневая модель, аналогичная трехуровневой модели проектирования традиционных баз данных

Были попытки классифицировать некоторые стандартные связки из выделенных особенностей СУАБД. Но как показывает практика единственным возможным способом сделать это - рассматривать классы приложений и, исходя из этого, определять их требования к функционалу системы, и затем идентифицировать классы СУАБД, которые удовлетворяют особенностям характерным для таких типов приложений. Таким образом, можно определить необходимую и достаточную реализацию из особенностей, описанных выше [22].

Классифицировать СУАБД можно, например, по ее месту и роли в информационной системе. Для такого рода классификации СУАБД можно использовать следующие критерии:

- роль СУАБД в информационной системе;
- степень интеграции с информационной системой.

В соответствие с ролью СУАБД в информационной системе можно определить назначение СУАБД - *мониторинг* или *контроль*. Мониторинг подразумевает, что СУАБД проверяет запросы к базе данных на соответствие их ограничениям, накладываемым на данные для поддержания их целостности. И после этого выполняет простые действия, например, уведомление, прекращение транзакций, распространение обновлений и т.д.

СУАБД, которая управляет информационной системой, т.е. осуществляет контроль, способна также вызывать внешние функции, например, прикладные программы. В этом случае, СУАБД в состоянии управлять поведением всей прикладной окружающей среды, а не только состоянием базы данных.

По степени интеграции с информационной системой СУАБД подразделяются на *гомогенные* и *гетерогенные*. Информационная система в данном контексте является гомогенной, если все ее компоненты являются приложениями СУАБД, то есть, они разделяют общую схему и общие базы данных. В противном случае информационная система является гетерогенной. В принципе, СУАБД может иметь контролируемую систему, которая реализована на других платформах.

Заключение. Задачи, которые ставятся перед современными СУБД, обслуживающими различные информационные системы, требуют, чтобы информационная система могла активно реагировать на события, происходящие как внутри нее, так и за ее пределами. Одним из возможных решений состоит в применении активных баз данных и систем управления активными базами данных, которые поддерживают механизмы, позволяющие автоматически реагировать на события, которые происходят как внутри, так и вне информационной системы.

Модель знаний активной базы данных определяет принципы построения активных правил и базируется на таких элементах, как: состояние, событие, условие, действие (State, Event, Condition, Action). Модель выполнения описывает обработку правил во время выполнения, и определяет режим связывания условий и действий, масштаб транзакций, политику сетевого эффекта, цикловая политика, приоритеты, планирование активности, управление ошибками и т.д.

Основные особенности СУАБД, выделенные в данной статье, касаются моделей АБД, функционала и набора инструментов систем управления активными базами данных. На основе проведенного анализа можно классифицировать системы база данных на предмет их соответствия концепции СУАБД и определить требования к разработке программного обеспечения СУАБД.

Список литературы

1. **Бекич З.** Активные базы данных: аналитический обзор // Программирование. 1990. № 5. С. 63-72.
2. **Коголовский М. Р.** Энциклопедия технологий баз данных. М.: Финансы и статистика, 2005. 800 с.
3. **Саймон А. Р.** Стратегические технологии баз данных [Электронный ресурс] / пер. с англ. URL: ilpubs.stanford.edu:8090/24/1/1993-17.pdf
4. **Agrawal R., Cochrane R., Lindsay B.** On maintaining priorities in a production rule system [Electronic Resource] // Proc. 17th VLDB. Barcelona, Spain, September 1991. URL: citeseerx.ist.psu.edu/viewdoc/download?doi=10.1
5. **Blue A., Brown B., Gray W.** An implementation of alerters for health district management // Proceedings of the Sixth British national conference on databases (BNCOD) / ed. by W. Gray. 1988. P. 125-140.
6. **Branding H., Buchmann A., Kudrass T., Zimmermann J.** Rules in an open system: the REACH rule system [Electronic Resource] // Proc. rules in database systems: workshops in computing. 1994. URL: portal.acm.org/citation.cfm?id=253262.253287
7. **Ceri S., Gottlob G., Tanca L.** Logic programming and databases [Electronic Resource]. Berlin: Springer-Verlag, 1990. URL: elib.tu-darmstadt.de/tocs/2089653.pdf
8. **Ceri S., Widom J.** Managing semantic heterogeneity with production rules and persistent queries [Electronic Resource] // Proceedings of the Nineteenth international conference on very large data bases / ed. by R. Agrawal, S. Baker, and D. Bell. Morgan-Kaufmann, San Mateo, CA, 1993. P. 108-119. URL: ilpubs.stanford.edu:8090/24/1/1993-17.pdf
9. **Chakravarthy S., Mishra D.** An event specification language (Snoop) for active databases and its detection [Electronic Resource]: technical report UF-CIS TR-91-23 / CIS Department. University of Florida. September 1991. URL: www.dbis.ethz.ch/education/ws0708/adv_top_infsyst/papers/snoop_dke94.pdf
10. **Dayal U., Hsu M., Ladin R.** A transactional model for long-running activities [Electronic Resource] // Proc. 17th VLDB. Barcelona, Spain, September 1991. URL: citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.68.7445&rep=rep1&type=pdf
11. **Dayal U., Hsu M., Ladin R.** Organising long-running activities with triggers and transactions // Proceedings of the SIGMOD conference. New York: ACM, 1990. P. 204-214.
12. **Diaz O.** Deriving rules for constraint maintenance in an object-oriented database // Proceedings of the International conference on databases and expert systems DEXA, I. R. A. M. / ed. by Tjoa. Springer-Verlag, 1992. P. 332-337.
13. **Diaz O., Jaime A., Paton N.** A debugger for active rules in an object-oriented context // Proceedings of the First international workshop on rules in database systems / ed. by N. Paton and M. Williams. Springer-Verlag, 1994. P. 180-193.
14. **Diaz O., Jaime A., Paton N., Al Qaimari G.** Supporting dynamic displays using active rules [Electronic Resource] // SIGMOD. 1994. Rec. 23. № 1. P. 21-26. URL: portal.acm.org/citation.cfm?id=181555
15. **Etzion O.** PARDES - a data-driven oriented active database model [Electronic Resource] // Ibidem. 1993. Rec. 22. № 1. P. 7-14. URL: portal.acm.org/citation.cfm?id=156884
16. **Geppert A., Ditrich K.** Rule-based implementation of transaction model specifications [Electronic Resource] // Proceedings of the First international workshop on rules in database systems / ed. by N. Paton and M. Williams. Springer-Verlag, 1994. P. 127-142. URL: citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.9648
17. **Hsu M., Ladin R., McCarthy D.** An execution model for active DBMS [Electronic Resource] // Proc. 3 Intl. conf. on data and knowledge bases. Jerusalem, Israel. June 1988. URL: books.google.com
18. **Ishikawa H., Ohta Manabu.** An active web-distributed database system for E-commerce [Electronic Resource] // ACM computing survey. URL: Acm.com
19. **Moss J. E. B.** Nested transactions: an approach to reliable distributed computing [Electronic Resource] // MIT Press. 1985. URL: publications.csail.mit.edu/lcs/pubs/pdf/MIT-LCS-TR-260.pdf
20. **Naqvi W., Ibrahim M.** Rule and knowledge management in an active database system // Proceedings of the First international workshop on rules in database systems / ed. by N. Paton and M. Williams. Springer-Verlag, 1994. P. 58-69.
21. **Paton N. W., Diaz O.** Active database systems [Electronic Resource] // ACM computing surveys. 1999. March. Vol. 31. № 1. URL: www.cs.man.ac.uk/~norm/papers/surveys.pdf
22. **Paton N., Diaz O., Williams M., Campin J., Dinn A., Jaime A.** Dimensions of active behaviour [Electronic Resource] // Proc. rules in database systems: workshops in computing. Springer-Verlag, 1994. URL: citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.51.7388&rep=rep1&type=pdf
23. **Paton N., Doan D., Diaz O., Jaime A.** Exploitation of object-oriented and active constructs in database interface development [Electronic Resource] // Proceedings of the Third international workshop on interfaces to database systems / ed. by J. Kennedy and P. Barclay. Springer-Verlag, 1996. URL: citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.54.813&rep=rep1&type=pdf
24. **Picouet P., Vianu V.** Semantics and expressiveness issues in active databases [Electronic Resource] // ACM computing surveys. URL: Acm.com
25. **Stonebraker M., Jhingran A., Goh J., Potamianos S.** On rules, procedures, caching and views in database systems [Electronic Resource] // Proceedings of ACM SIGMOD. 1990. P. 281-290. URL: db.cs.berkeley.edu/papers/ERL-M90-36.pdf
26. **Trajcevski G., Scheuermann P.** Reactive maintenance of continuous queries [Electronic Resource] // Mobile computing and communications review. Vol. 8. № 3. URL: elib.tu-darmstadt.de/tocs/2089653.pdf
27. **Widom J., Cochrane R., Lindsay B.** Implementing set-oriented production rules as an extension to starburst [Electronic Resource] // Proceedings of the Seventeenth international conference on very large data bases (Barcelona), R. C. G. / ed. by M. Lohman and A. Sernadas. Morgan-Kaufmann, San Mateo, CA, 1991. P. 275-286. URL: www.vldb.org/conf/1991/P275.PDF