ПЛАТФОРМА ДЛЯ РАЗРАБОТКИ ПРИЛОЖЕНИЙ АКТИВНЫХ БАЗ ДАННЫХ

С.В. Шибанов, А.А. Скоробогатько, А.Б. Зудов, П.В. Вишняков

Введение в активные базы данных. С развитием технологий баз данных (БД) и систем управления базами данных (СУБД) увеличивается круг областей применения и решаемых ими задач. Изначально базы данных рассматривались лишь как репозитории, которые хранят информацию, необходимую для приложений, и доступны прикладным программам или пользователям. Традиционные системы управления базами данных (СУБД) пассивны в том смысле, что действия, выполняемые СУБД, инициируются пользователями или программными приложениями.

Примерами систем, при разработке которых приходится преодолевать пассивность традиционных баз данных, являются корпоративные распределенные информационные системы и базы данных, системы удаленного управления и конфигурирования приложений, в том числе работающие в сети Интернет. Другими характерными примерами таких системам также являются технические системы – системы контроля и управления техническими объектами, системы управления движущимися объектами, а ткеж геоинформационные системы [1]. Как правило, в таких системах имеется большая, быстро изменяющаяся во времени база элементарных фактов, описывающая состояния объектов. Система должна обнаруживать некоторое, возможно большое, число заданных ситуаций и автоматически реагировать на их возникновение соответствующими действиями. Каждая из обнаруживаемых ситуаций задается своей системой правил. Данные, изменяющие базу элементарных фактов, могут поступать от различных датчиков, а также вводиться оператором с клавиатуры ЭВМ или других устройств ввода. Решение задач хранения и обработки информации в таких системах целесообразно возлагать на систему управления активной базой данных (СУАБД).

База данных называется активной базой данных (АБД), если система управления базой данных по отношению к ней выполняет не только те действия, которые явно указывает пользователь, но и дополнительные действия в соответствии с правилами, заложенными в саму БД [2]. При этом СУАБД постоянно отслеживает наступление определенных событий, реагируя на активность системы и пользователя, и отвечают на события путем вызова процедур, затрагивающих как саму базу данных, так и ее окружение. В данном случае АБД являются не просто пассивным набором данных, но и хранилищем знаний по обработке и управлению данными.

Применение активных баз данных в информационных системах позволяет перенести управляющую бизнес-логику в СУАБД, которая будет сама обнаруживать возникновение заранее описанных событий в объектах контроля и управления и реагировать на них в соответствии с заранее определенными действиями. Таким образом, СУАБД может стать не просто центром хранения и обработки информации от объектов контроля, но также центром принятия решений и выработки управляющих воздействий к объектам управления.

Чтобы обеспечивать реагирующее поведение СУАБД должна поддерживать соответствующие модели описания и выполнения правил. Классическая модель описания активных правил основывается на правилах, которые содержат три компонента: событие (event), условие (condition) и действие (action) и называются ECA-правилами (Event–Condition–Action) [3]. Компонент Событие (event) описывает какое-либо событие, которое может произойти в СУБД или вне нее. Компонент Условие (condition) проверяет контекст, при котором событие произошло. Действие (action) описывает процедуру, которая должна быть выполнена правилом, если соответствующее событие произошло и условие оказалось истинным. Действия могут менять структуру базы данных или набора правил, выполнять некоторый вызов поведения внутри базы данных или внешний вызов, информировать пользователя или системного администратора о какой-либо ситуации, прерывать транзакцию или принимать альтернативную линию поведения.

Классическая модель ECA-правил не учитывает, что в реальных приложениях поведение объекта часто зависит от его состояния (state). Состояние – это абстракция значений и связей объекта, которая определяет отклик объекта на получаемые события. В конкретном состоянии обрабатываются только те события, для которых явным образом описано поведение, любые другие события игнорируются. Это свойство состояний особенно важно – оно позволяет наделить объекты системы свойством изменчивости – способности изменять свое поведение в зависимости от возникающих событий.

Модель, учитывающую состояние объекта называется расширенной моделью ECA-правил, моделью SECA-правил или SECA-моделью (State–Event–Condition–Action). Добавление в классическую модель ECA-правил понятия состояния сокращает количество необходимых правил и упрощает процесс разработки активной системы [3]. Выбор модели активных правил зависит от особенностей предметной области и логики выполнения активных правил.

Обобщенная архитектура платформы. Предлагается платформа для разработки приложений активных баз данных на основе существующих реляционных и объектно-ориентированных СУБД. При этом функциональность активных баз данных реализована в виде переносимых компонентов, выполненных

средствами традиционных СУБД. Компоненты имеют единую архитектуру и набор метаданных на стороне сервера баз данных и различаются только нюансами реализации для конкретных СУБД.

Платформа для разработки приложений активных баз данных реализована на основе клиент-серверной архитектуры и включает в себя сервер активных баз данных (сервер АБД), активный сервер приложений, активное клиентское приложение, а также набор клиентских приложений для управления сервером активных баз данных и непосредственно АБД (рисунок 1).

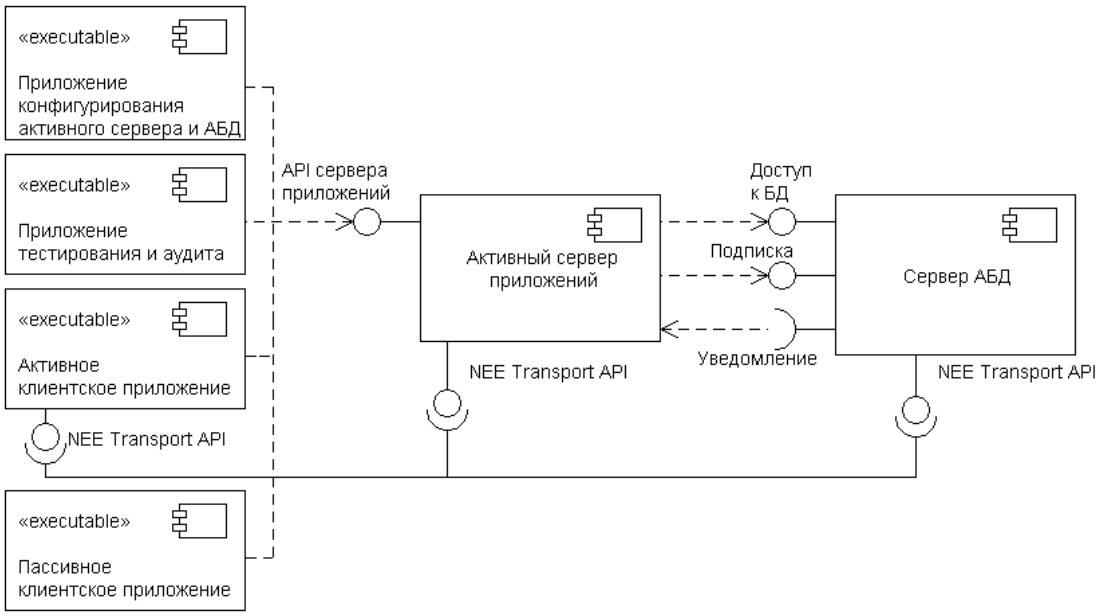

Рис. 1 Обобщенная архитектура платформы для разработки приложений активных баз данных

Роль СУАБД в данной платформе выполняет сервер АБД. Можно выделить несколько возможных вариантов архитектур для построения сервера АБД: монолитная архитектура, сервисная архитектура и архитектура в виде надстройки над традиционной СУБД [4].

Построение СУАБД в виде монолитной системы предполагает реализацию активных компонент как неотъемлемую часть СУБД. Такая реализация делает активные компоненты сильно связанными с конкретной СУБД. В результате СУАБД становится частью большого программного продукта, который сложно внедрить и осуществлять техническую поддержку. Кроме того, применение монолитной СУАБД может «утяжелить» информационную систему.

В СУАБД, выполненных в виде отдельного сервиса, слой активности находится «поверх» стандартной СУБД. Активные компоненты в этом случае менее связаны и ограничены. Однако такой подход проигрывает в производительности и увеличивает количество программных слоев системы.

Реализация СУАБД в виде надстройки над традиционной СУБД позволяет использовать все преимущества традиционных СУБД, дополнив их механизмами активных баз данных. Кроме того, данный подход делает возможным интеграцию активной функциональности в существующие приложения [4].

Сервер активных баз данных. Центральной частью платформы является сервер активных баз данных – сервер баз данных с возможностью хранения и управления активными базами данных. Архитектура сервера АБД изображена на рисунке 2.

Рассмотрим основные компоненты сервера АБД.

СУБД. Сервер АБД строится на базе существующей системы управления базами данных. СУБД осуществляет управление активными и пассивными базами данных, а также активными метаданными и шаблонами. Сервер АБД управляет как активными базами данных, так и традиционными пассивными базами данных.

Активная база данных — это база данных, содержащая помимо собственно данных предметной области список активных правил. Эти правила описывают ситуации, возникающие в базе данных, и требуемую реакцию СУБД на эти ситуации. Функционирование активных правил обеспечивается стандартными механизмами СУБД.

Пассивная база данных — обычная база данных, содержащая только данные предметной области и не содержащая активных правил.

На стороне сервера АБД также размещаются база метаданных АБД и база шаблонов АБД.

База метаданных АБД содержит служебную информацию сервера АБД: настройки сервера, список активных баз данных на сервере, список подключений активных клиентов и т.п. В зависимости от

конфигурации сервера, база активных метаданных может также являться централизованным хранилищем активных правил сервера.

База шаблонов АБД – база данных, содержащая шаблоны триггеров и хранимых процедур, применяемые для автоматизации построения активных баз данных. На основе шаблонов формируется набор программных средств (ПС) управления АБД в процессе создания.

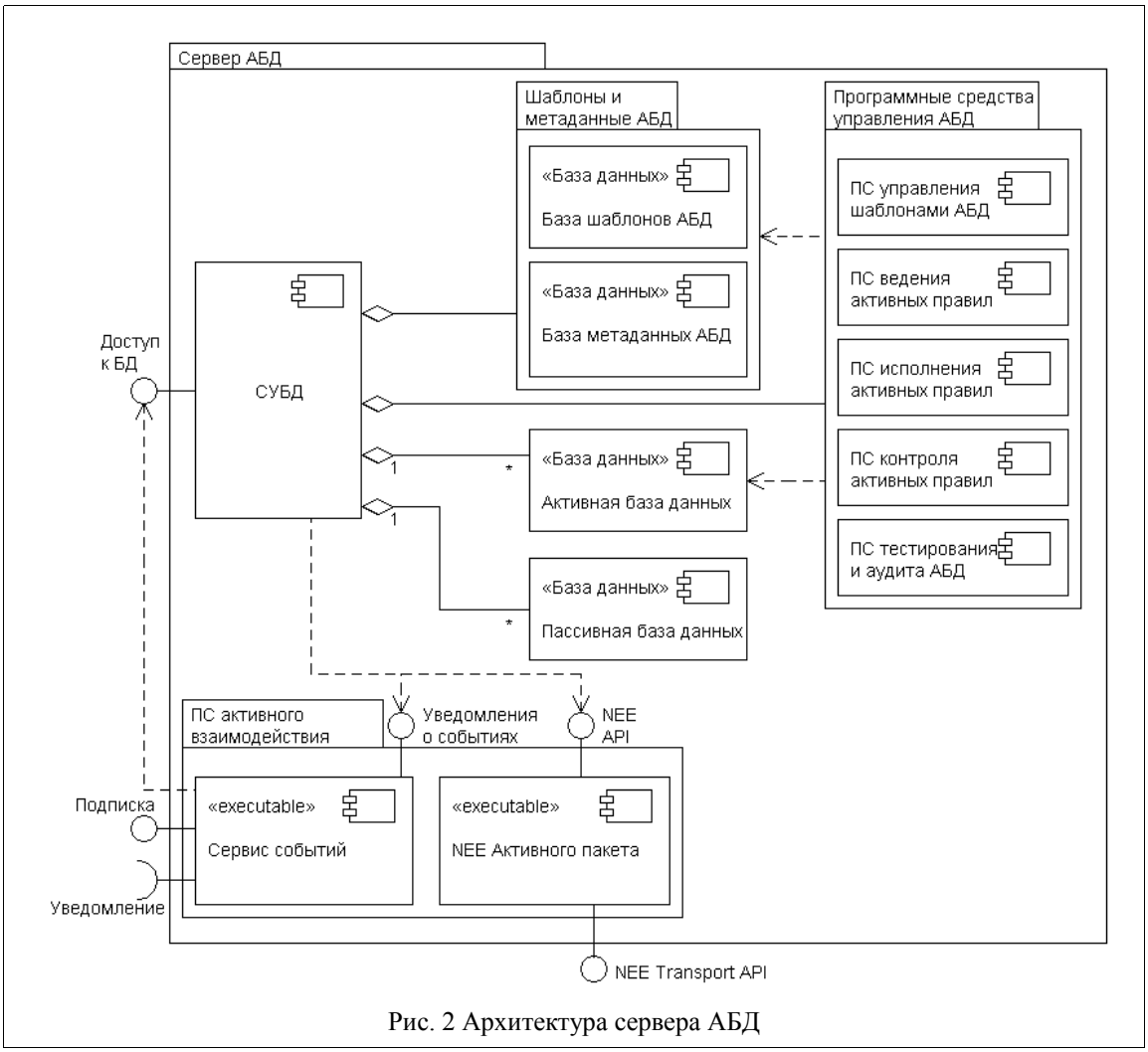

Рис. 2 Архитектура сервера АБД

Программные средства управления АБД включают в себя следующие компоненты: ПС управления шаблонами АБД, ПС ведения активных правил, ПС исполнения активных правил, ПС контроля активных правил, ПС тестирования и аудита АБД.

Программные средства управления шаблонами обеспечивают создание и редактирование шаблонов, а также генерацию на их основе программного кода для управления АБД (триггеров, хранимых процедур, функций обратного вызова и т.п.).

Программные средства ведения правил реализуют программный интерфейс для создания и модификации активных правил. Ведение правил включает выполнение следующих операций: создание/удаление/редактирование правил, включение/выключение правил, анализ связей между правилами, тестовый запуск (без сохранения изменений), назначение приоритета обработки, настройка параллельного выполнения правил, протоколирование правил [5].

Программные средства исполнения правил – совокупность программных средств, отвечающих за обнаружение событий, поиск и выполнение активных правил в процессе функционирования активной базы данных. Для управления активным поведением реализованы специальные программные средства на основе механизмов, предоставляемых целевой СУБД. В реляционных СУБД такими механизмами являются триггеры, хранимые процедуры и функции, в объектно-ориентированных СУБД – методы классов и функции обратного вызова. При этом события, связанные с модификацией (добавлением, удалением и обновлением) данных, отслеживаются и обрабатываются явно. События, которые не могут быть сведены к операциям модификации данных, фиксируются и обрабатываются в специальной очереди событий. Одной из особенностей активных правил является сложность их взаимодействия друг с другом. Поэтому отдельной задачей средств исполнения активных правил является исключение различного рода ошибочных ситуаций. При выполнении группы правил

выполняется оптимизация, основанная на использовании принципа результирующего эффекта и группировки событий.

Программные средства контроля активных правил обеспечивают обнаружение возможных некорректных и конфликтных ситуаций при добавлении активного правила и во время исполнения. В каждом из этих случаев применяются методы статического и динамического анализа, соответственно.

Основной задачей статического анализа является обнаружение потенциального зацикливания и неконфлюентности с последующим выводом соответствующего предупреждения. Кроме того, с помощью статического анализа можно заранее находить слишком длинные цепочки инициирующих друг друга правил, чтобы впоследствии оптимизировать их [6].

Динамический анализ используется для протоколирования выполненных правил, оптимизации, а также для отката транзакции в случае зацикливания правил. Динамический анализ позволяет разделить выполнение правила по отдельным транзакциям, назначить приоритеты правилам при одновременном вызове, задать условия параллельной обработки одновременно вызываемых правил, группировать события правил, протоколировать выполнение правил.

Программные средства тестирования и аудита АБД предназначены для тестирования производительности активных баз данных в процессе разработки и оперативного аудита АБД во время работы системы. ПС тестирования и аудита реализуется в виде модуля управления эталонным тестированием и модуля сбора статистики [7].

Эталонное тестирование выполняется аналогично эталонным тестам TPC-C/S, разработанных международным советом по производительности обработки транзакций (Transaction Performance Council, TPC) [8].

Управление тестированием полностью перенесено на сервер АБД в двухуровневой архитектуре или сервер приложений в трехуровневой архитектуре. Исходные данные для тестирования и алгоритм их итерационного изменения определяются профилем нагрузки, хранящимся в базе метаданных АБД, что позволяет оперативно оценивать эффективность различные типы правил в зависимости от заданного профиля нагрузки. Возможно использование как ранее подготовленных исторических тестовых данных, так и их генерация перед началом тестирования согласно профилю нагрузки. Промежуточные и итоговые результаты тестирования сохраняются в базе метаданных АБД для дальнейшего анализа.

Для оперативного аудита АБД используется модуль сбора статистики. Этот модуль позволяет фиксировать время работы правил и сохранять его в базу метаданных АБД для дальнейшего использования (например, для сравнения эффективности различных реализаций правил).

Взаимодействие клиентских приложений и/или специализированных серверов приложений с СУАБД реализуется с помощью сервиса сообщений, среды исполнения активного пакета, а также традиционным для приложений баз данных способом – на основе языков запросов.

Сервис событий — специализированная служба для доставки приложениям информации о событиях, возникающих на сервере АБД и в самих АБД. Сервис предоставляет возможность подписки на события АБД и рассылает клиентам уведомления о возникших событиях. Интерфейс «Уведомление о событиях» позволяет АБД уведомлять сервис о возникающих событиях. Интерфейсы «Подписка» и «Уведомление» служат для взаимодействия с клиентскими приложениями.

NEE Активного пакета — среда исполнения активного пакета, отвечает за распространение активного пакета по сети передачи данных и за выполнение инструкций активного пакета. Активный пакет является дополнительным средством распространения активности в системе. С его помощью можно передать клиентскому приложению не только информацию о возникшем событии, но и набор инструкций, которые нужно выполнить при возникновении события.

Структура метаданных АБД. С целью унификации программных средств платформы на стороне различных целевых СУБД и упрощения реализации платформы для новых СУБД предлагается обобщенная структура метаданных активной базы данных [9], представленная на рисунке 3.

Рассмотрим основные элементы метаданных АБД. Класс Object описывает объекты активной базы данных. Объект обладает именем, уникальным в пределах активной базы данных, и может находиться в одном из нескольких состояний. В каждый момент времени объект может находиться только в одном из своих состояний. Для определения текущего состояния служит метод getCurrentState. Реализация метода getCurrentState различается для разных объектов — она хранится в атрибуте getCurrentStateCode в виде строки и выполняется при вызове getCurrentState. Это позволяет связывать объекты класса Object с любыми объектами предметной области (в т.ч. Составными).

Класс State описывает состояние объекта активной базы данных. Имена состояний должны быть уникальными в пределах соответствующего им объекта. Связь объекта предметной области с соответствующим состоянием реализуется через метод enterState. Данный метод получает в качестве параметра идентификатор объекта предметной области и изменяет его состояние.

Объекты класса Event содержат информацию о событиях активной базы данных. Процедура обработки события инициируется обработчиком события. Обработчик события — это программный код, выполняемый СУБД при возникновении события. Вызов обработчика может выполняться из триггера, хранимой процедуры,

клиентского приложения и т.п. Обработчик состоит из двух частей: формирование контекста события и поиск и выполнение правил, соответствующих событию.

Контекст события — это информация о возникшем событии. Состав этой информации зависит от конкретного события. Контекст события может использоваться при проверке условия и выполнении действия.

Класс Event содержит методы для установки (setupHandler) и удаления (removeHandler) обработчика события. Эти методы выполняют всю необходимую работу: создание/удаление триггера, добавление обертки к хранимой процедуре и т.п.

Объекты класса Action описывают действия активной базы данных. Действие — это последовательность операций над объектами активной базы данных. При выполнении действию доступна информация, содержащаяся в базе данных, а также данные из контекста события. Последовательность операции действия хранится в атрибуте actionCode. Эти операции представлены в виде запросов к базе данных

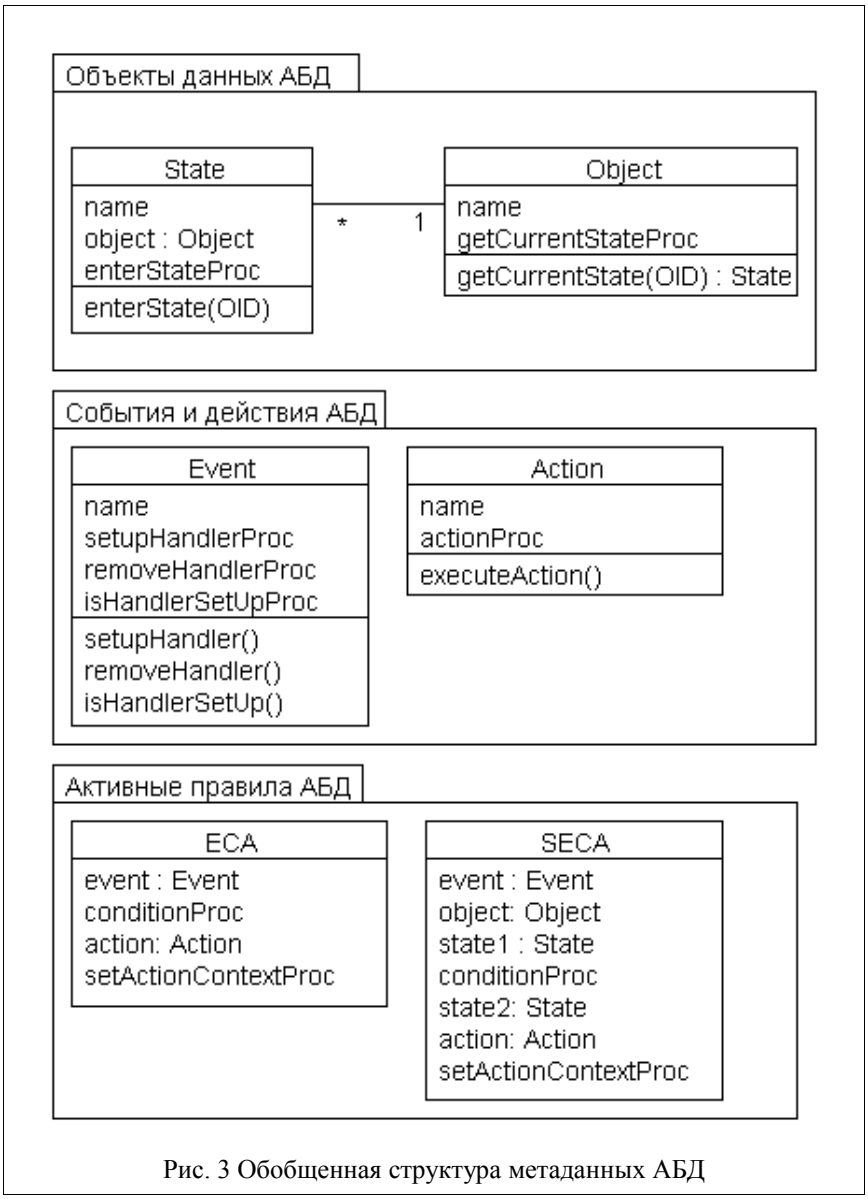

Рис. 3 Обобщенная структура метаданных АБД

на языке СУБД и выполняются при вызове метода executeAction.

Классы ECA и SECA содержат информацию соответственно о ECA- и SECA-правилах активной базы данных. Атрибуты классов соответствуют элементам ECA- и SECA-правила.

Применение абстрактной модели активной базы позволяет использовать единую концепцию управления АБД вне зависимости от средств реализации и используемой СУБД. В дальнейшем эта модель отображается на объектно-ориентированную или реляционную модель данных и реализуется средствами конкретной СУБД, например, в виде классов и методов для объектно-ориентированных СУБД или в виде таблиц, хранимых процедур и триггеров для реляционных СУБД. Благодаря этому программные средства платформы на стороне различных целевых СУБД имеют общую структуру и механизм работы.

Активный сервер приложений. В рамках предлагаемой платформы разработан активный сервер приложений [10], на котором разворачивается дополнительный набор компонентов для взаимодействия с сервером АБД (рисунок 4).

Драйвер АБД – это библиотека для взаимодействия компонент активного сервера приложений с сервером АБД, предоставляющая базовые операции управления сервером АБД и активными базами данных.

Набор драйверов целевой СУБД, в данном случае драйверы Cache', Oracle, SQL Server, обеспечивает непосредственное взаимодействие активного сервера приложений с целевой СУБД. райвер АБД осуществляет управление своими реализациями и выбирает нужную в зависимости от того, к какой СУБД производится подключение.

Набор программных интерфейсов (API) обеспечивает доступ клиентским приложениям к функциональности сервера АБД и активного сервера приложений.

API конфигурирования активного сервера предоставляет программный интерфейс для настройки сервера АБД, создания и администрирования активных баз данных, ведения активных правил и т.п.

API тестирования и аудита активного сервера предоставляет программный интерфейс для доступа к статистической информации о работе АБД и всей системы в целом.

API взаимодействия с АБД позволяет активному клиентскому приложению, осуществить подписку на события и получение уведомлений от сервера АБД, а также взаимодействие с средой исполнения (NEE) активного пакета.

На стороне активного сервера приложений также присутствует среда исполнения активного пакета (NEE Активного пакета).

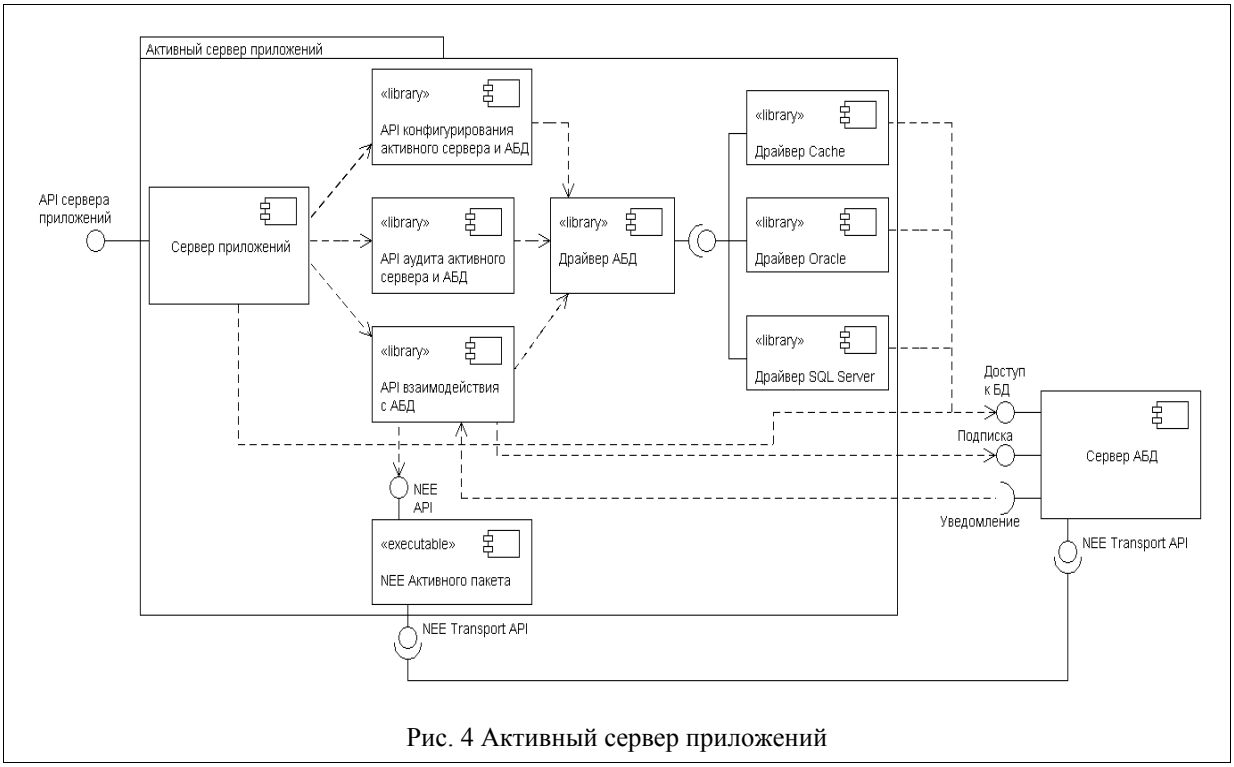

Рис. 4 Активный сервер приложений

Кроме того, сервер приложений может взаимодействовать с сервером АБД через интерфейс запросов, минуя драйвер АБД.

Клиентские приложения. Набор клиентских приложений платформы включает в себя следующие приложения: rлиентское приложение конфигурирования активного сервера и АБД, клиентское приложение тестирования и аудита, активное клиентское приложение.

Кроме того, на базе предлагаемой платформы могут функционировать и традиционные (пассивные) клиентские приложения. Взаимодействие клиентских приложений с сервером АБД в двухуровневой архитектуре показано на рисунке 5. При этом используются те же интерфейсы, что и в активном сервере приложений.

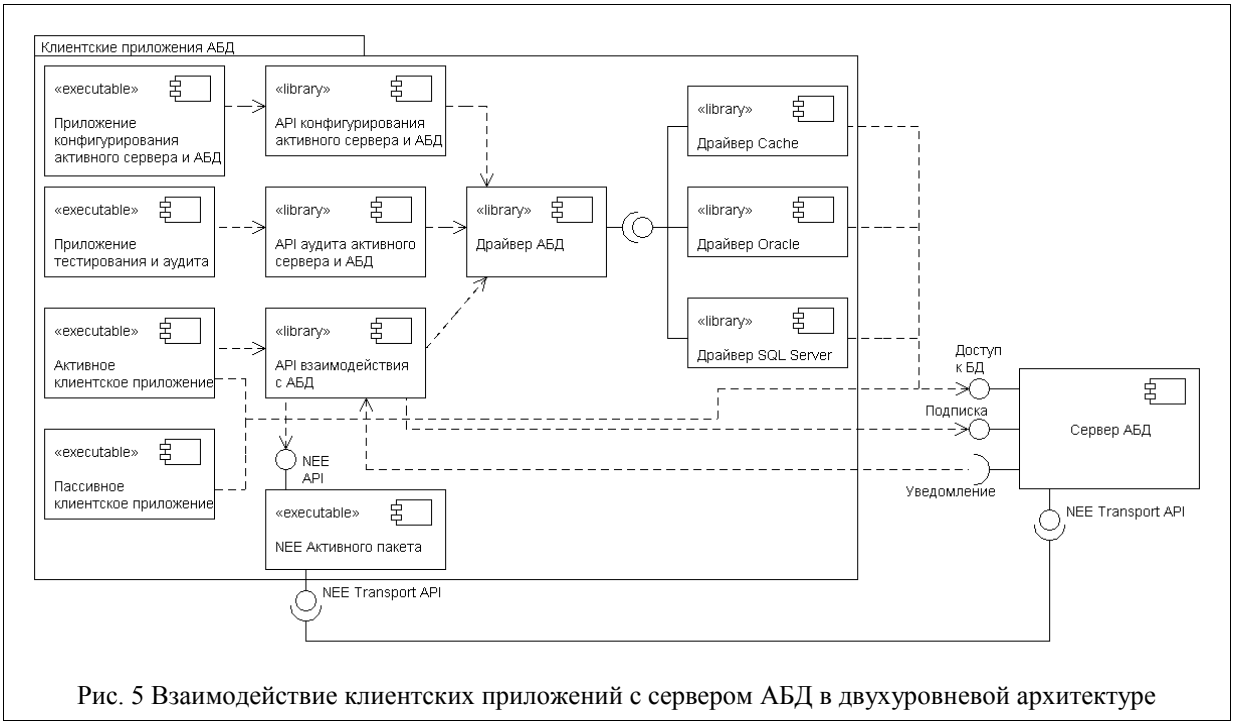

Рис. 5 Взаимодействие клиентских приложений с сервером АБД в двухуровневой архитектуре

Если информационная система построена в соответствии с трехуровневой архитектурой, то клиентские приложения взаимодействуют с активным сервером приложений через API сервера приложений. Активный сервер приложений, в свою очередь, взаимодействует с сервером АБД.

Заключение. В настоящее время программная платформа находится на стадии исследовательского прототипа. Отработана концепция построения СУАБД в виде надстройки над современными реляционными и объектно-ориентированными системами управления базами данных. Реализованы серверные компоненты платформы для реляционных СУБД Microsoft SQL Server 2008, Oracle 11g, постреляционной СУБД PostgreSQL 8.4 и объектно-ориентированной СУБД Cache 2010.1 [11].

В рамках общей концепции разработаны средства взаимодействия отдельных компонентов клиент-серверных приложений активных баз данных. Реализованы программные интерфейсы для построения активного сервера приложений и клиентских приложений.

Ведутся разработки клиентских приложений платформы, как для управления и сопровождения активных база данных, так и для конечных пользователей.

В настоящее время, в том числе на базе разработанного прототипа платформы, проводятся исследования и разработки по нескольким направлениям:

1. создание формальной модели активной системы и ее отображение в объектно-ориентированную и реляционную модели;
2. адаптация современных методик проектирования информационных систем для приложений активных баз данных;
3. поиск и реализация новых эффективных методов и средств проверки корректности активных правил;
4. разработка методики и средств тестирования и оценки производительности активных баз данных в рамках предлагаемой платформы.

В конечном итоге, предлагаемая в данной статье платформа может быть использована для разработки приложений активных баз данных с применением современных реляционных и объектно-ориентированных СУБД, в том числе в уже развернутых и эксплуатирующихся программных решениях.

ЛИТЕРАТУРА:

1. Шибанов С.В., Лысенко Э.В. Концептуальные особенности систем управления активными базами данных// Альманах современной науки и образования. – Тамбов: Грамота, №11 (42), 2010. – Ч. 2. – с. 106-114.
2. The Active Database Management System Manifesto: A Rulebase of ADBMS Features. A Joint Report by the ACT-NET Consortium. SIGMOD Record, Vol. 25, No. 3, September 1996.
3. Norman W. Paton, Oscar Diaz. Active Database Systems. ACM Computing Surveys, Vol. 31, No. 1, March 1999.
4. Шибанов С.В., Скоробогатько А.А., Лысенко Э.В. Архитектура программных средств управления активными базами данных// Математическое и программное обеспечение систем в промышленной и социальной сферах: междунар. сб. науч. трудов. – Магнитогорск: Изд-во Магнитогорск. гос. техн. ун-та им. Г.И. Носова, 2011. – Ч. I. – с. 47-52.

5. Шибанов С.В., Зудов А.Б. Проблемы построения правил в активных базах данных// Математическое и программное обеспечение систем в промышленной и социальной сферах: междунар. сб. науч. трудов. – Магнитогорск: Изд-во Магнитогорск. гос. техн. ун-та им. Г.И. Носова, 2011. – Ч. I. – с. 36-41.
6. Шибанов С.В., Зудов А.Б. Зацикливание правил в активных базах данных// Университетское образование (МКУО-2012): Сб. статей XVI международной научно- методической конференции. - Пенза, Изд-во ПГУ, 2012, с. 357-359.
7. М.Р. Когаловский Энциклопедия баз данных. М.: Финансы и Статистика, 2002.
8. Шибанов С.В., Вишняков П.В. Система тестирования серверного программного обеспечения активных баз данных // В кн.: Надежность и качество-2012: Труды международ. симпозиума, т. I. - Пенза, Изд-во Пенз. ГУ, 2012. – С. 434-436.
9. Шибанов С.В., Скоробогатько А.А., Лысенко Э.В. Интегрированная модель активных правил// Математическое и программное обеспечение систем в промышленной и социальной сферах: междунар. сб. науч. трудов. – Магнитогорск: Изд-во Магнитогорск. гос. техн. ун-та им. Г.И. Носова, 2011. – Ч. I. – с. 41-47.
10. Шибанов С.В., Лысенко Э.В. Исследование и разработка системы управления активными базами данных// Перспективы науки. – Тамбов, № 6 (21), 2011. с. 115-120.
11. Шибанов С.В., Лысенко Э.В., Скоробогатько А.А., Зудов А.Б., Вишняков П.В. Реализация абстрактной модели активных баз данных средствами современных СУБД// В кн.: Надежность и качество: Труды международ. симпозиума, т. I. - Пенза, Информац.-изд. центр Пенз. гос. ун-та, 2010. – с. 306-314.