

ТЕХНОЛОГИЯ АГЕНТНО-РЕЛЯЦИОННОГО ОТОБРАЖЕНИЯ ПРИ АГЕНТНО-ОРИЕНТИРОВАННОМ ПРОГРАММИРОВАНИИ

^{1,2}Лукьянчиков О.И., ^{1,2}Никульчев Е.В., ¹Филатов В.В.

¹ Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный университет информационных технологий, радиотехники и электроники» (МИРЭА), Москва, Россия (119454 Россия, г. Москва, проспект Вернадского, 78), e-mail: luk-it6@yandex.ru

² МТИ «Московский технологический институт», Москва, Россия (119334, Ленинский проспект, д. 38а), e-mail: luk-it6@yandex.ru

Существующие и часто используемые принципы модульного программирования вызывают трудности на этапе интеграции. Предлагается агентно-ориентированное программирование, базирующееся на агентно-реляционном отображении, объединяющее технологии удаленного доступа и объектного отображения реляционных баз данных. Данная технология определяет интерфейс взаимодействия между удаленными распределенными объектами с помощью принципов и операций работы с реляционной базой данных. В данной статье описаны теоретические положения технологии агентно-реляционного отображения и построена её объектная модель, которая содержит минимальный набор классов, необходимых для её реализации. Агентно-ориентированное программирование позволяет оперировать объектами на другом уровне абстракции. Разработчик практически не заботится о реализации операций, он только выполняет команды, оперируя данными: как в БД, так и на удаленных вычислительных узлах.

Ключевые слова: распределенные системы, модульное программирование, базы данных (БД), технология удаленного доступа, агентно-ориентированное программирование (АОП), объектно-реляционное отображение (ORM), агентно-реляционное отображение (АРО).

APPROACH TO THE TRAINING OF PERSONNEL MANAGEMENT SERVICES OPEN DISTRIBUTED INFORMATION SYSTEMS

^{1,2} Lukyanchikov O.I., ^{1,2} Nikulchev E.V., ¹ Filatov V.V.

¹ Federal State Educational Institution of Higher Education «Moscow State University of Information Technologies, Radio Engineering and Electronics» (MIREA), Moscow, Russia (119454 Russia, Moscow, Vernadskogo avenu, 78), e-mail: meomirea212@gmail.com

² MTI "Moscow Technological Institute", Moscow, Russia (119334, Leninsky Prospekt, d. 38a), e-mail: luk-it6@yandex.ru

Existing and often used principles of modular programming cause difficulties at the stage of integration. Proposed agent-oriented programming, based on the agent-relational mapping technology combines remote access and object-relational database mapping. This technology determines the interface between remote distributed objects using the principles and operations work with a relational database. This article describes the theoretical principles of technology agent-relational mapping and built its object model that contains a minimal set of classes required for its implementation. Agent-oriented programming allows you to operate the facilities at another level of abstraction. Developer practically takes care of the implementation of operations, it only executes the commands in terms of data: in the database, and remote computing nodes.

Key words: Distributed systems, modular programming, database (DB), the technology of remote access, agent-oriented programming (AOP), object-relational mapping (ORM), agent-relational mapping (ARC).

Введение

Масштабы современных информационных систем постоянно растут, увеличивается количество программных компонентов, и, разумеется, увеличиваются объемы программного кода. Поэтому в жизненном цикле разработки больших информационных систем используется способ, при котором программа разбивается на относительно независимые составные части – программные модули. Данный способ, имеет недостатки на этапе интеграции, которые в большинстве случаев вызваны несоблюдением правил, протоколов и стандартов, а также использованием разнотипных технологий и принципов программирования. Дело в том, что различные модули разрабатывают отдельные люди или группы людей (отделы на предприятиях). И каждая, в свою очередь, использует то, что им удобней, что лучше знают, что проще использовать. Мировое сообщество пытается стандартизовать интерфейсы модулей, приводя их к единой природе. На сегодняшний день известны два стандарта, создавшие свои IDL (*Interface Definition Language* – язык определения интерфейсов): COM+ (*Component Object Model* – объектная модель компонентов) и COBRA (*Common Object Request Broker Architecture* – общая архитектура брокера объектных запросов).

Дело в том, что модульное программирование так и не смогло полностью реализовать все свои концепции в полной мере [1]. Например, не каждый модуль оформляется в виде самостоятельного программного продукта, в виде исполняемого файла – модули подгружаются в процессе выполнения программы и размещаются в том же адресном пространстве, что и основная программа. Независимость при этом достигается довольно условная. Разделение выполнения модулей на различных вычислительных узлах невозможно. В связи с чем необходим другой подход к организации программирования, а именно подход *агентно-ориентированного программирования (АОП)*. Изменение подхода к программированию требует новых методов проектирования и технологий для разработки информационных систем. В качестве технологии для АОП предлагается технология *агентно-реляционного отображения (АРО)*. Данная технология объединяет технологии объектно-реляционного отображения (ORM) [2] удаленного доступа, позволяя отображенным объектам реляционной БД выполнять реляционные операции к подобным удаленным объектам.

Агентно-ориентированное программирование

Платформой распределенных систем являются технологии промежуточного уровня, которые обеспечивают удаленное взаимодействие через сеть. К таким технологиям относятся: *CORBA, COM, DCOM, COM+, RPC, MOM*. Они используются в основе парадигм объектно-ориентированного программирования как инструмент для передачи информации по сети. Каждая из технологий предоставляет свой уникальный принцип удаленного взаимодействия:

либо через объекты, либо через удаленный вызов процедур, либо через подписки на события. Независимо от различий данных методов для удаленного взаимодействия, данные технологии позволяют решить ряд трудностей и проблем модульного программирования, так как позволяют модулям приобрести следующие свойства [3]:

- независимость модулей, обеспечивающая гибкость систем;
- сетевое взаимодействие, благодаря чему системы имеют широкое географическое распределение;
- распределение вычислений, что позволяет достигать высокой эффективности систем.

Обобщающим все эти свойства является агентно-ориентированный подход программирования (АОП), предложенный *Шохемом* в работах [7,8]. При данном подходе проектирования основополагающим понятием является агент, который находится в среде, позволяющей взаимодействовать с другими агентами и определяющей его поведение. В данной парадигме программирования под агентом понимается не обязательно «агент» из терминологии многоагентной системы. В данной парадигме минимальной единицей декомпозиции может так же выступать и объект, и актор (универсальный примитив параллельного численного расчёта, который может принимать и посылать сообщения, устанавливая, как следует реагировать на последующие сообщения, а так же создавать новые акторы) [6], и отдельный процесс или сервис.

Проектирование и разработка систем на основе принципов АОП направлено на рассмотрение систем с точки зрения размещения модулей, их взаимодействия и потоков данных. При данном проектировании разработчик оперирует другими устоявшимися понятиями агентов, такими как: убеждение, способность, обязательство, информирование, просьба, предложение, обещание, честность. Всё это выводит на иной уровень понимания и абстрагирования архитектуры системы. Однако на текущий момент такое проектирование и описание систем является бессмысленным в связи с отсутствием единой технологии для реализации АОП.

Технология агентно-реляционного отображения

Агентно-реляционное отображение (АРО) – отображение реляционной модели в объектную модель, экземпляры которой имеют возможность удаленного взаимодействия между собой [3,5].

Суть технологии, реализующей АРО, заключается в предоставлении отображенным объектам от таблиц реляционной БД возможностей удаленного взаимодействия методами с принципами выполнения операций к реляционной БД. Технология АРО является гибридом

технологий удаленного доступа и объектно-реляционного отображения (ORM). Для совместного использования этих двух технологий определены дополнительные возможности, сформулированные в следующих трех теоретических положениях.

Положение 1. Наличие базового набора операций при удаленном взаимодействии:

- выборка (*select*) – получение данных удаленного объекта или удаленное выполнение метода;
- изменение (*update*) – установка данных удаленного объекта;
- добавление (*insert*) – добавление в удаленный контейнер нового объекта;
- удаление (*delete*) – удаление из удаленного контейнера объектов.

Данный базовый набор операций предоставляет единый интерфейс для выполнения операций как к БД, так и к удаленным объектам, поэтому вся работа с технологией осуществляется принципами и методами работы с реляционной БД.

Положение 2. Наличие механизма транзакций при выполнении операций, реализуемого командами зафиксировать (*commit*) и откатить транзакцию (*rollback*). В связи с тем, что система является распределенной, транзакции выполняются параллельно, и конфликты распознаются в процессе работы. Поэтому для технологии АРО предпочтительной является спекулятивная стратегия обработки транзакций. Последовательность операций при данной обработке транзакций показана на рис. 1.

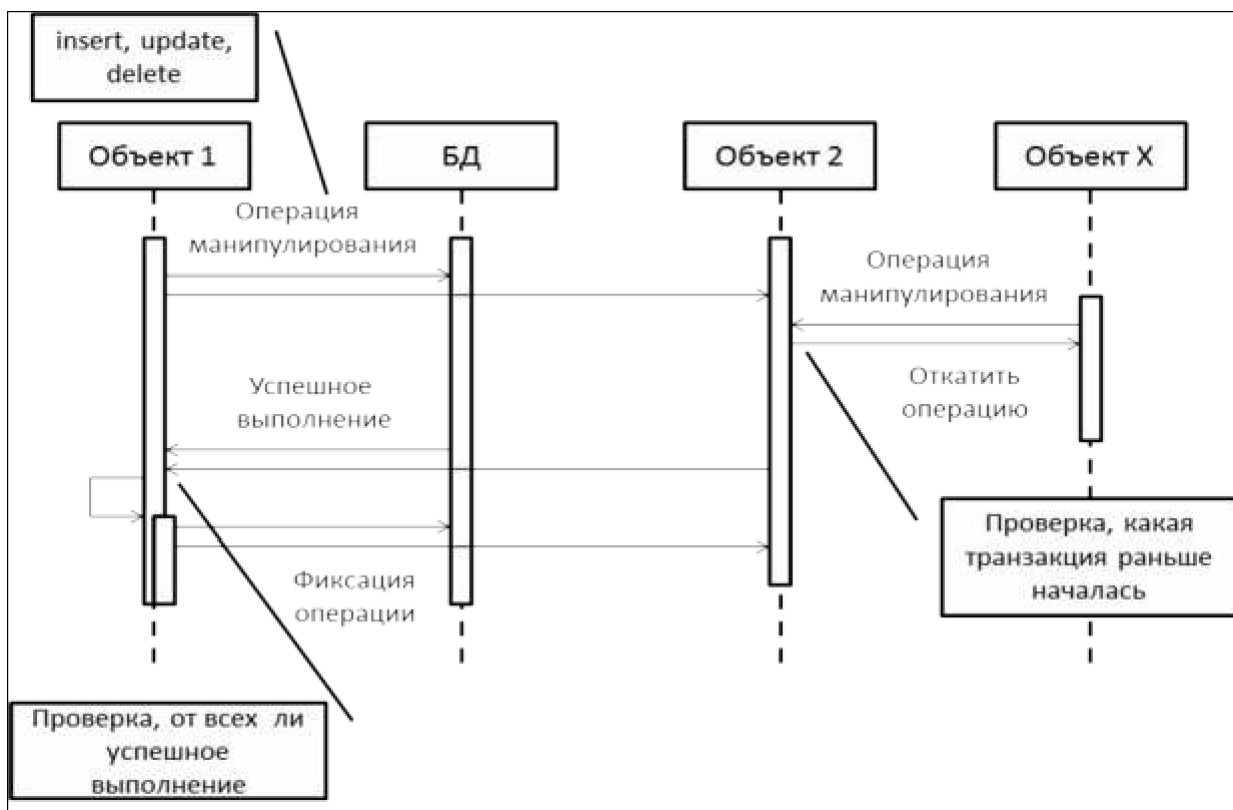


Рис. 1. Последовательность выполнения операций манипулирования данными.

Когда объект выполняет операции манипулирования данными, он рассылает удаленным объектам и БД команду на выполнения операции и при этом становится инициатором, который отвечает за выполнение транзакции.

Операция выполняется и на всех узлах, и в БД; об успешном или неудачном результате выполнения оповещается инициатор. Если некий другой объект попытается выполнить конфликтующую операцию, то транзакции сравниваются по моменту времени их начала, и та, которая началась позже, откатывается. Инициатор, получив подтверждения об успешном выполнении всех операций, фиксирует изменения у себя и отправляет всем объектам соответствующую команду.

Положение 3. Для функционирования механизма транзакций требуется информация обо всех удаленных объектах. Данная информация формируется при запуске и завершении работы приложений командами регистрации (*registration*) и командой отмены регистрации (*unregistration*). В результате каждое приложение содержит «карту» обо всех удаленных объектах, которая используется как для механизма транзакций, так и для более удобной адресации к удаленным объектам.

Объектная модель технологии агентно-реляционного отображения

На основе теоретических положений технологии АРО построена объектная модель в нотации языка UML, изображенная на рис. 2, содержащая набор классов для её реализации.

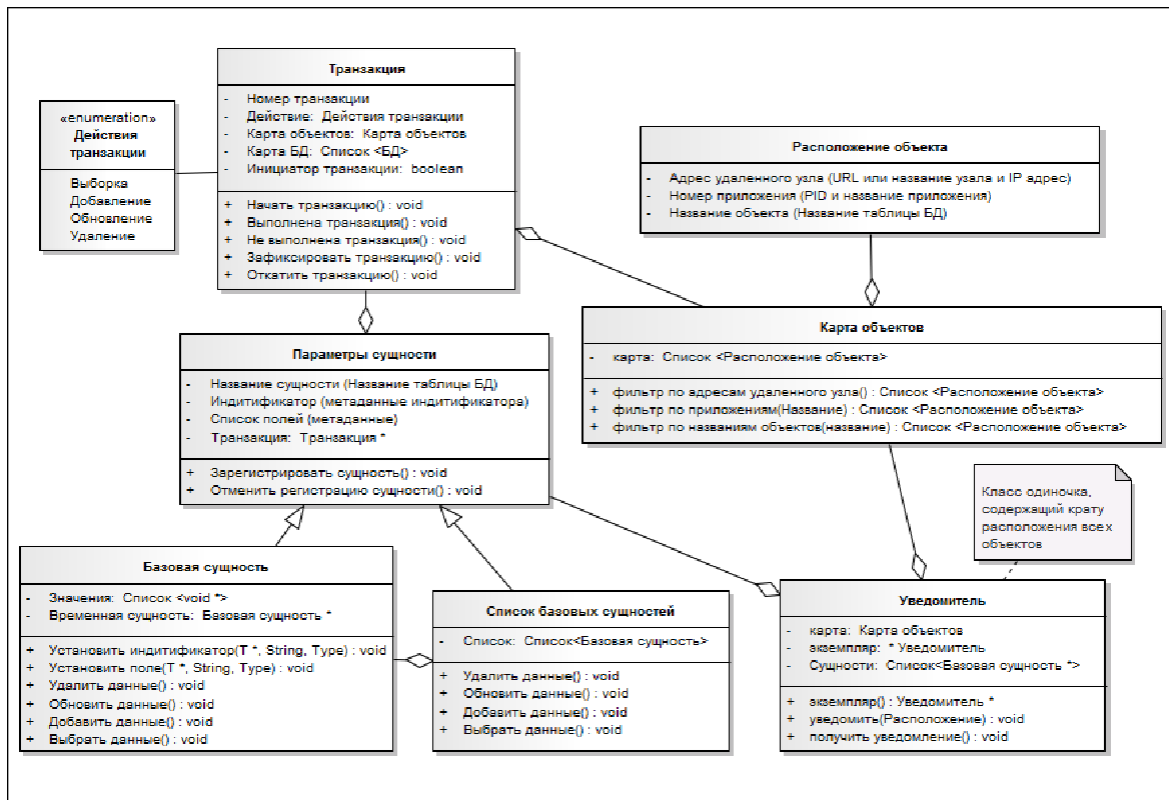


Рис. 2. Объектная модель технологии агентно-реляционного отображения в нотации языка UML

Класс «Карта объектов» содержит в себе список всех удаленных объектов, выполняя положение 3. Для примера приведена адресация для карты, за которую отвечает класс «Расположение объекта»: производится сначала на уровне вычислительных узлов, затем на уровне приложений, и затем на уровне объектов. Данная адресация предоставляет удобный способ для фильтрации объектов карты до необходимого уровня. Карта объектов формируется методами «Зарегистрировать сущность» и «Отменить регистрацию сущности», содержащимися в классе «Параметры сущности».

Класс «Параметры сущности» содержит в себе метаданные сущности: название сущности, название полей и атрибутов, их типы данных и информацию об идентификаторе. Данный класс является родительским классом для «Базовая сущность» и «Список базовых сущностей». Класс «Базовая сущность» содержит в себе значения сущности и базовые методы технологии АРО по выборке, добавлению, удалению и изменению данных, реализуя непосредственно само АРО сущностей. Класс «Список базовых сущностей» содержит в себе список из объектов класса «Базовая сущность» и используется в качестве контейнера сущностей АРО.

Каждая сущность содержит в себе объект «Транзакция». Класс «Транзакция» содержит в себе его уникальный номер, действие, время начала выполнения транзакции, списки параметров БД и удаленных объектов, по которым объект транзакции проверяет, везде ли успешно выполнена операция, чтобы вызвать метод «Зафиксировать транзакцию». Если хоть один удалённый объект или элемент в списке или БД не выполнил успешно операцию или превысил время её операции, то вызывается метод «Откатить транзакцию».

За всеми сущностями в приложении следит объект одиночка (может быть только один в приложении) класса «Уведомитель». Также данный класс отвечает за отправку команд и за получение их от других удаленных объектов сущностей, выполняя доставку их к требуемому объекту на уровне приложения.

Данная модель отображает основные классы и зависимости между ними, которые позволят реализовать все требуемые возможности технологии АРО.

Заключение.

Гербер Шилдт, известный американский программист, принимавший стандарты С и С++, сказал о появлении ООП: «На каждом этапе развития программирования появлялись методы и инструментальные средства для “обуздания” растущей сложности программ. И на каждом таком этапе новый подход вбирал в себя всё самое лучшее из предыдущих, знаменуя собой прогресс в программировании»[4]. Так же и АРО – в соответствии с текущими тенденциями развития технологий разработки программного обеспечения, получив лучшее от современных технологий – порождает новый подход в

программировании. Технология АРО позволяет оперировать объектами на другом уровне абстракции. Разработчик практически не заботится о реализации команд, он только выполняет их, оперируя данными: как в БД, так и на удалённых вычислительных узлах. Такая прозрачность при разработке приложений позволяет технологии АРО стать основной стандартной технологией при АОП.

Список литературы

1. Бреслер И.Б., Семенов С.А., Корниенко В.В., Борисов В.В. Перспективный подход к организации программных комплексов // Радиопромышленность, 2009. Вып. 1. С. 72–88.
2. Лукьянчиков О.И. Технология агентно-реляционного отображения // Современные проблемы науки и образования. 2015. № 1; Режим доступа: URL: <http://www.science-education.ru/121-18623>.
3. Лукьянчиков О. И. Технология объектно-реляционного отображения реляционных данных при агентно-ориентированном программировании // Задачи системного анализа, управления и обработки информации. Межвузовский сборник научных трудов. Выпуск 5.М.: Изд. МТИ, 2015. С. 85–96.
4. Шилдт Г. С# 4.0 полное руководство, СПб.: Вильямс, 2011. 1056 с.
5. Ambler S. W. Mapping Objects to Relational Databases: O/R Mapping In Detail. —Ambysoft Inc. 2010.
6. Hewitt C., Bishop P., Steiger R. A. Universal Modular Actor Formalism for Artificial Intelligence.// IJCAI'73 Proceedings of the 3rd international joint conference on Artificial intelligence. - Morgan Kaufmann Publishers Inc., 1973. — P. 235–245.
7. Shoham Y. Agent Oriented Programming // Computer Science Department, Stanford University - 1990.
8. Shoham, Y. Agent-Oriented Programming // Artificial Intelligence. 1993 pp. 51-92.