Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра математического и компьютерного моделирования

Разработка мультиагентной системы для приложения «Торговля книгами» с применением платформы JADE

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 5 курса 561 группы

направление 09.03.03 – Прикладная информатика

механико-математического факультета

Баркова Алексея Александровича

Научный руководитель
доцент, к.ф.-м.н. С.В. Иванов

Зав. кафедрой
зав. каф., д.ф. – м. н. Ю.А. Блинков

Саратов 2017

Введение. Данная работа посвящена разработке мультиагентной системы для приложения «Торговля книгами». Работа состоит из введения, трёх разделов и заключения.

В введении содержится постановка задачи, цель и краткое описание предметной области.

Цель данной работы – создание приложения для покупки и продажи книг. Задачи – создание агентов, работа агентов между собой с использованием языка программирования JAVA и платформы JADE.

В первом разделе рассматриваются теория мультиагентных систем, которые представляют собой совокупность нескольких автономных сущностей, существующих и взаимодействующих в общей среде и платформы JADE, которая является одним из самых распространенных инструментов для создания мультиагентных систем, изначально созданный в Telecom Italia.

Во втором разделе рассматривается процесс запуска платформы JADE, описание нескольких встроенных в саму платформу агентов.

В третьем разделе разрабатываются агенты имеющие одни и те же свойства и правила поведения, которые будут взаимодействовать друг с другом, классы взаимодействия агентов, режимы при которых они выполняют действия, посылки сообщений, сервисы и приложение «Торговля книгами»

В заключение приводятся результаты проделанной работы.

Основное содержание работы.

Первый раздел. Мультиагентная система (МАС) представляет собой совокупность нескольких автономных сущностей, существующих и взаимодействующих в общей среде – программных интеллектуальных агентов. Агент автономно существует в рамках некой внешней среды, он может обмениваться с ней информацией, совершать определенные действия, но при это не может управлять состоянием среды.

Агент — это развитие известного понятия объект, представляющего абстракцию множества экземпляров предметов реального мира, имеющих одни и те же свойства и правила поведения. Свойства объекта описываются исходной системой, а правила поведения — порождающей системой, чаще всего структурированной. Состояние объекта определяется перечнем его свойств с текущими значениями. Объект со значениями всех его свойств определяет экземпляр, моделью которого является кортеж соответствующего реляционного отношения из системы данных. В число свойств объекта включаются его идентификатор, а также указывающие, описывающие и вспомогательные атрибуты. Последние два типа атрибутов делятся по отношению к методу объекта на входные и выходные. Описательные атрибуты определяют свойства, внутренне присущие объекту, а вспомогательные — его структурные связи с экземплярами других объектов.

Под интеллектуальным агентом в информатике и искусственном интеллекте понимаются любые физические или виртуальные единицы, способные, по крайней мере, поддерживать взаимодействие с окружающим миром, получая от него информацию, и, реагируя на нее своими действиями, проявлять собственную инициативу, посылать и получать сообщения от других агентов и вступать с ними во взаимодействие, действовать без вмешательства извне, в том числе и без вмешательства человека.

Агентная система – это платформа, которая может создавать, интерпретировать, запускать, перемещать и уничтожать агенты. Также как и агент, агентная система ассоциируется с полномочиями, которые определяют

организацию или персону, от имени которых работает система. Агентная система с полномочиями конкретного пользователя, реализовывает политику безопасности этого пользователя в плане защиты его ресурсов.

Агентная система однозначно идентифицируется именем и адресом. На одной машине могут размещаться несколько агентных систем.

Все общение между агентными системами осуществляется через коммуникационную инфраструктуру (КИ). Администратор сетевого региона определяет службы коммуникации для внутри региональных и межрегиональных взаимодействий.

Коммуникационная инфраструктура обеспечивает транспортные службы связи, службу имен и службу безопасности для агентных систем. Когда агент перемещается, он движется между стационарными процессами, которые называются местами. Место - контекст внутри агентной системы, в котором может быть запущен агент. Этот контекст может предоставлять набор функций, таких как контроль доступа, например. Выходное место и место адресации могут находиться как в рамках одной агентной системы, так и в разных агентных системах, которые поддерживают одинаковую совокупность параметров агента. Место сопоставляется с местом расположения, агентная система может содержать одно или несколько мест и место может содержать одного или более одного агентов.

Существуют стандарты, которые должны существовать в системе также как и агенты, и взаимодействовать между собой. Они были создна специально для мультиагентных систем и называются стандарты FIPA. А также, стандарты коснулись реализации систем, в которых агенты могли бы выполняться и действовать. Для таких платформ были специфицированы три выделенных служебных роли агентов:

- система управления агентами (Agent Management System - AMS) – этот агент отвечает за доступ к справочнику всех агентных идентификаторов (AID). Каждому AID в справочники сопоставляется адрес агента в системе, через который будет возможно сообщение по транспортному протоколу.

Все агенты платформы регистрируются в справочнике через AMS при появлении в системе, чтобы получить AID;

- справочный агент (Directory Facilitator - DF) – предоставляет сервис «Желтых страниц» другим агентам в системе. Каждый из агентов должен зарегистрировать свои сервисы в этом справочнике. При обращении к DF для каждого конкретного сервиса клиенту возвращается список агентов, реализующих этот сервис;
- сервис транспорта сообщений (Message Transport Service - MTS) – отвечает за передачу сообщений между агентами в рамках одной платформы, а также за связь с агентами в других платформах.

Также стандарты описывают язык общения между агентами - FIPA Agent Communication Language (ACL), однако только синтаксис и семантику, реализация этого языка не специфицирована. Помимо этого, спецификации FIPA описывают протоколы взаимодействия агентов. Наиболее известен FIPA Contract Net - он применяется для выбора агента-поставщика необходимого сервиса среди нескольких кандидатов.

JADE (Java Agent Development) - это один из самых распространенных инструментов для создания мультиагентных систем, изначально созданный в Telecom Italia. JADE написан на Java, поэтому является кроссплатформенным, может работать на беспроводных мобильных устройствах, поддерживает стандарты FIPA.

Основополагающие принципы платформы:

- функциональная совместимость – продукт JADE разработан в соответствии со спецификациями FIPA. Как следствие JADE-агенты могут взаимодействовать со сторонними агентами, поддерживающими этот стандарт;
- единообразность – продукт JADE предоставляет гомогенный набор прикладных программных интерфейсов, которые не зависят ни от базового устройства сети, ни от версии платформы Java. При

развертывании разработчики приложений должны определить тип среды исполнения Java;

- простота использования – набор простых и интуитивно-понятных интерфейсов прячет сложную логику ПО промежуточного слоя от пользователя.

Как платформа для выполнения интеллектуальных агентов JADE:

- обеспечивает возможность развертывания МАС и берет на себя функции управления жизненным циклом агентов (создание, выполнение, приостановка и завершение). Библиотека Java – классов позволяет реализовать агентов со сколь угодно сложным поведением, а также описать их взаимодействие. В основном, это классы, описывающие агентов на каждом этапе их жизненного цикла, а также их взаимодействие с графическим интерфейсом (если он есть);
- поддерживает использование различных архитектур агентов и мультиагентых систем – гибкая исходная структура JADE позволяет лего расширять платформу (например - интеграции с Jess, и). JADE не придерживается какой-то конкретной методологии и обладает достаточно высокой гибкостью в использовании;
- обеспечивает сообщение агентов в системе (механизм транспорта сообщний, языки и онтологии), представление знаний. Для передачи сообщений в JADE используется Agent Communication Channel (ACC). Все взаимодействия между агентами происходят посредством пересылки сообщений языка FIPA ACL;
- включает в себя инструменты разработки и отладки.

Помимо агентов и поведений, JADE реализует FIPA-протоколы взаимодействия между агентами, такие как FIPA Contract Net Interaction Protocol.

Область применения агентной платформы Jade можно охарактеризовать следующим образом:

- замена/расширение сервис-ориентированной архитектуры (SOA) для систем, в которых требуется больше интеллекта и более сложные взаимодействия между сервисами;
- Semantic Web;
- rich client приложения, в т.ч. мобильные;
- подходит для быстрого прототипирования.

Агентная платформа JADE не подходит, для:

- реал-тайм систем;
- высокопроизводительных высоконагруженных систем;
- 3D анимации, компьютерных игр, serious games, реалистичных симуляций, виртуальной реальности;
- моделирования финансовых рынков, социальных и иных систем.

Второй раздел. Среда JADE представляет собой набор инструментов, выполняющих управление агентным приложением и его отладку. Для того чтобы запустить среду JADE, необходимо:

- установить на свой компьютер J2EE не ниже версии 1.5;
- установить на свой компьютер среду разработки Java-приложений IntelliJ IDEA;
- установить на свой компьютер среду разработки JADE доступной версии, например, версии 3.6.1. Для этого необходимо скопировать\загрузить файлы среды JADE на свой компьютер, например, в папку C:\Jade;
- изменить конфигурационный параметр CLASSPATH;

Платформа JADE содержит в себе уже несколько стандартных агентов: remote management agent, sniffer agent, introspector agent.

В приложении действуют агенты-книготорговцы и агенты-покупатели, приобретающие книги по поручению своих владельцев.

Каждый агент-покупатель получает название книги, которую необходимо купить («требуемая книга»), в качестве аргумента командной строки и периодически запрашивает предложения у всех известных ему агентов-

продавцов. Как только предложение получено, агент-покупатель принимает его и выдает заказ на покупку. Если несколько агентов-продавцов предоставляют предложение агенту-покупателю, он выбирает из них лучшее (с самой низкой ценой). После покупки агент-покупатель завершает свою работу.

Каждый агент-продавец имеет минимальный интерфейс, с помощью которого пользователь может добавить названия новых книг и цены на них в свой локальный каталог книг для продажи. Агент-продавец непрерывно проверяет поступление запросов от агентов-покупателей. При получении запроса на книгу каждый агент-продавец проверяет свой локальный каталог и, если запрашиваемая книга находится в его каталоге, отвечает на запрос, сообщая агенту-покупателю цену. После выполнения заказа купленная книга автоматически удаляется из каталога.

Третий раздел. Каждый агент имеет определенный «идентификатор агента», представляющий собой экземпляр класса jade.core.AID. Метод getAID() класса Agent позволяет получить идентификатор агента. Объект AID содержит глобальное уникальное имя, а также адрес. Идентификатор агента JADE имеет структуру: <имя_агента>@<имя_платформы>, поэтому агент с именем Peter, находящийся на платформе P1, будет иметь глобальное уникальное имя Peter@P1. Адрес, содержащийся в AID, – это адрес платформы, где находится агент. Этот адрес используется только в тех случаях, когда агент должен общаться с другими агентами, которые находятся на другой платформе.

При запуске среды JADE выполняется инициализация ядра сервисов JADE. Процесс завершается сообщением о том, что контейнер с именем «Main-Container» готов к работе. Когда среда JADE запускает пользовательского агента, он выводит своѐ приветствие с именем «Buyer», указанным при создании агента. Имя платформы «NBNT2004130496: 1099/JADE» назначается автоматически в соответствии с адресом хоста и порта, на котором работает среда JADE.

В сценарии будут действовать 4 продавца книг и 4 покупателя. Продавцы могут предлагать по различным ценам как одну, так и несколько книг трех наименований: «Hamlet», «Romeo and Juliet», «King Lear». Каждый покупатель

стремится купить одну из 3-х книг, причем двое из них хотят купить книгу одного наименования. Каждому продавцу и каждому покупателю сопоставляется собственный агент.

Агенты продавцов и покупателей имеют определенную модель поведения и критерии принятия решения по выбору приемлемого варианта решения.

Агент продавца имеет цель продать весь имеющийся в наличии товар и имеет параметры в виде списка продаваемых книг с указанием наименования и цены, а также критерий выбора при котором продажа книги производится первому обратившемуся покупателю по первоначально заявленной цене.

Агент покупателя имеет цель приобрести определенную книгу с заданным наименованием и имеет параметры в виде наименования книги, которую необходимо приобрести, а также критерий при котором он выбирает продавца, который предлагает книгу с подходящим наименование и по минимальной цене.

Для запуска агентов можно компилировать их непосредственно из .java файлов с необходимыми параметрами, передаваемыми в командной строке. Однако удобнее воспользоваться механизмом регистрации, предоставляемым системным агентом RMA. Для этого файлы классов агентов необходимо сначала скомпилировать в .jar файлы, при этом название файла должно совпадать с названием главного класса в файле (т.к. при регистрации агента с помощью графического интерфейса не предусмотрено поле для ввода имени jar-файла, только для ввода имени класса). Далее создаем агентов продавцов книг.

Создадем агента покупателя книг. Для этого выберем пункт главного меню Actions -> Start New Agent. Назовем агента BookBuyerHamlet. Наименование книги, которую хочет купить агент, указывается в поле Arguments.

После регистрации всех агентов покупателей они сразу вступают в процесс матчинга. Каждый агент покупателя находит наиболее подходящего продавца и создает с ним связь.

Заключение. В ходе выполнения работы, был проведен анализ и изучен теоретический материал о мультиагентных системах, которые представляют собой совокупность нескольких автономных сущностей, существующих и взаимодействующих в общей среде и платформе JADE, которая является одним из самых распространенных инструментов для создания мультиагентных систем, а также рассмотрена архитектура платформы и стандарты FIPA.

Описан метод запуска и установки платформы JADE, которая уже имеет ряд стандартных агентов. Разработаны агенты, классы, режимы, сервисы и модули системы, описана работа агентов между друг другом.

Проведен обзор платформы JADE и ее возможностей для разработки мультиагентной системы для различных приложений.

В результате работы получено приложение «Торговля книгами» с агентами – покупателями и агентами – продавцами, с описанием поведений данных агентов, посылки сообщений в приложении, получение сообщений, блокировки режима работы агентов, последовательность выполнения сценария, а также результат матчинга данного приложения.