

Санкт-Петербургский государственный университет

Математико-механический факультет

Кафедра системного программирования

Говейнович Сергей Геннадьевич

Мультиагентная система координации кооперативной логистики

Выпускная работа бакалавра

Допущен к защите

Зав. кафедрой:

д.ф.-м.н., проф. А.Н. Терехов

Научный руководитель:

к.ф.-м.н., доцент Д.Ю. Бугайченко

Рецензент:

к.ф.-м.н., доцент И. П. Соловьев

Санкт-Петербург

2013

Saint-Petersburg State University

Faculty of Mathematics and Mechanics

System engineering department

Multi-agent coordination system for carpool logistics

Graduate paper

Goveynovich Sergey

Scientific advisor D. Y. Bugaychenko

/ signature /

Reviewer I. P. Soloviev

/ signature /

“Admitted to proof” A. N. Terehov

Head of the chair, / signature /

Saint-Petersburg

2013

Оглавление

| | |
|--|----|
| Введение..... | 5 |
| Постановка задачи..... | 7 |
| Обзор предметной области..... | 8 |
| Сценарий реализации доставки товаров..... | 8 |
| Обзор мультиагентных систем | 9 |
| Обзор используемых технологий | 11 |
| JADE Фреймворк..... | 11 |
| Характеристики JADE | 11 |
| Архитектура JADE | 12 |
| Архитектура главного контейнера JADE | 13 |
| Сервис передачи сообщений (MTS)..... | 14 |
| Инструменты: | 14 |
| BDI4JADE Фреймворк [7]..... | 15 |
| Архитектура системы | 16 |
| Типы агентов..... | 16 |
| Агент-посредник | 17 |
| Агент-перевозчик | 17 |
| Агент-покупатель | 18 |
| Агент-продавец..... | 18 |
| Реализация системы | 19 |
| Добавление агента-перевозчика | 19 |
| Добавление агента-покупателя..... | 22 |
| Роль агента-продавца..... | 24 |

| | |
|------------------|----|
| Результаты..... | 25 |
| Литература | 26 |

Введение

В настоящее время практически у каждого есть компьютер или мобильное устройство с Интернет доступом. Количество Интернет пользователей уверенно растет. В 2012 году впервые было продано смартфонов больше чем обычных мобильных телефонов. На сегодняшний день всемирная паутина помимо обычного хранилища информации, как это было на ранних этапах развития сети, предоставляет пользователям огромное количество образовательных, поисковых, развлекательных, социальных, коммерческих и других сервисов. Среди них особо стоит отметить сайты и сервисы, позволяющие совершать покупки, не выходя из дома.

С каждым днем все большее количество покупок совершается через Интернет. Это действительно удобно и выгодно обеим сторонам, продавцам и покупателям. Продавец получает экономию на аренде помещения, на консультантах и т.п., а так же автоматически получает возможность рекламировать свой магазин в сети. Покупатели, в свою очередь, получают возможность совершать покупки не вставая с компьютерного кресла в любое время суток, совершать покупки в зарубежных магазинах и, что самое главное, по более низким ценам чем предлагают обычные магазины.

Наибольшей экономии при покупке товаров удастся достичь при так называемых кооперативных, т.е. совместных, покупках. Совместная закупка — действительно дешевый способ получить дорогие и качественные вещи. Но и тут существуют свои подводные камни.

Основная проблема заключается в том, что довольно сложно организовать подобную закупку. Нужно найти подходящий товар по приемлемой цене, найти людей которые хотят его приобрести, организовать сбор денег с покупателей, закупку и доставку товара. На первый взгляд все выглядит не

так уж и сложно, но стоит заняться этим самому как сразу понимаешь насколько это сложная задача.

Но даже после того, как закупка выполнена проблемы организатора на этом не заканчиваются. Одна из проблем следующего этапа кооперативных продаж с которой сталкивается продавец — проблема доставки товаров со своего склада, который часто является обычной квартирой, и следовательно, у организатора нет возможности хранить там товары длительное время. Дабы сэкономить средства и тем самым снизить стоимость для клиентов хочется организовать доставку силами самих клиентов — участников данной кооперативной закупки. Так обычно и происходит, ведь в течение дня большинство людей передвигаются в пределах города на работу, в магазины, к родственникам на дачу и т.д. и их не затруднит передать небольшую посылку если место передачи будет находится по пути. Главное при таком способе доставки найти варианты маршрутов и договориться с потенциальными перевозчиками.

В рамках данной работы ставится задача создания прототипа информационной системы, помогающей в нахождении эффективного плана перевозок, которая требовала бы минимального участия пользователя, т.к. предполагается, что пользоваться системой будут в том числе люди далекие от информационных технологий.

Постановка задачи

- Анализ предметной области и потребностей потенциальных пользователей кооперативных перевозок.
- Разработка модели системы поддержки кооперативных перевозок.
- Реализация прототипа системы поддержки кооперативных перевозок на основе мультиагентного [1, 2] подхода.

Обзор предметной области

Сценарий реализации доставки товаров

Обычно сценарий построения плана кооперативной доставки товаров выглядит приблизительно так: [3, 4]

1. Продавец привозит заказанные товары на свой склад или в квартиру.
2. Проверяет все ли заказанные товары он привез. Уведомляет покупателей о привезенном или отсутствующем товаре.
3. Сопоставляет привезенные товары со списком покупателей.
4. Составляет перечень людей готовых помочь в доставке, список и время их маршрутов движения.
5. Долго и утомительно, вручную составляет план перевозок.
6. Обзванивает каждого, уведомляя о том, что товар прибыл, и оглашает имеющиеся опции по доставке.
7. Принимает заявки на начальную отгрузку и координаты первой передачи.
8. Организует начальную отгрузку.
9. Дальнейшие этапы доставки - забота покупателя.

В идеальном случае нужно реализовать систему, работающую по следующему сценарию.

1. Продавец привозит заказанные товары на свой склад или в квартиру.
2. Проверяет все ли заказанные товары он привез. Вносит их в базу данных, автоматически уведомляя клиентов электронным письмом.
3. Имея базу клиент-товар, созданную на этапе покупки, одним нажатием клавиши генерирует план доставки.
4. Загружает автомобили товарами и контролирует их доставку.

При реализации системы необходимо учитывать следующие факторы:

- План перевозок может не существовать в текущем состоянии системы или частично реализовывать поставленную задачу
- Перечень людей готовых помочь в перевозке постоянно меняется, меняются их маршруты, планы и т.д.

Обзор мультиагентных систем

Как видно из описания сценария работы системы, мы имеем дело со сложной постоянно растущей, децентрализованной средой. Для решения нашей задачи каждая часть системы должна оперативно реагировать на внешние изменения без постоянного контроля пользователя. В таких условиях целесообразно использовать мультиагентный подход, что мы и попытаемся сделать.

Мультиагентная система - система состоящая из набора автономных сущностей (агентов), взаимодействующих друг с другом посредством сообщений.

Агент называется интеллектуальным если он проявляет следующие свойства:

- *Реактивность:* агент способен реагировать на изменения внешней среды.
- *Проактивность:* агент способен выполнять некоторый перечень действий, направленных на достижения своей цели.
- *Социальность:* агент может общаться с другими агентами, заключать сделки, вступать в коалиции и т.д. для достижения общих или собственных целей.

Преимущества такого подхода в данной задаче очевидны:

- Отсутствует централизованное хранилище данных. Следовательно стабильность такой системы будет на высоком уровне.
- Для работы интеллектуальных агентов не требуется постоянного контроля со стороны пользователя. Стоит лишь дать агенту задачу (цель) и можно о нем забыть до того момента пока он не предложит решение.
- Остается возможность неограниченно усложнять сценарии поведения агентов не меняя ядро системы.
- Агенты, общаясь со своими агентами-партнерами, обеспечивают высокую скорость реакции на изменяющиеся условия задачи.

Обзор используемых технологий

JADE Фреймворк

Java Agent Development Framework (JADE) — программная среда разработки мультиагентных систем и приложений, поддерживающая FIPA-стандарты для интеллектуальных агентов.

Характеристики JADE

- соответствие последним спецификациям FIPA 2.0
- открытость кода, LGPL
- поддержка беспроводных мобильных устройств вплоть до JME MIDP1.0 (сотовые телефоны)
- наличие протоколов взаимодействия агентов (RDF, XML и т.д.)

Архитектура JADE

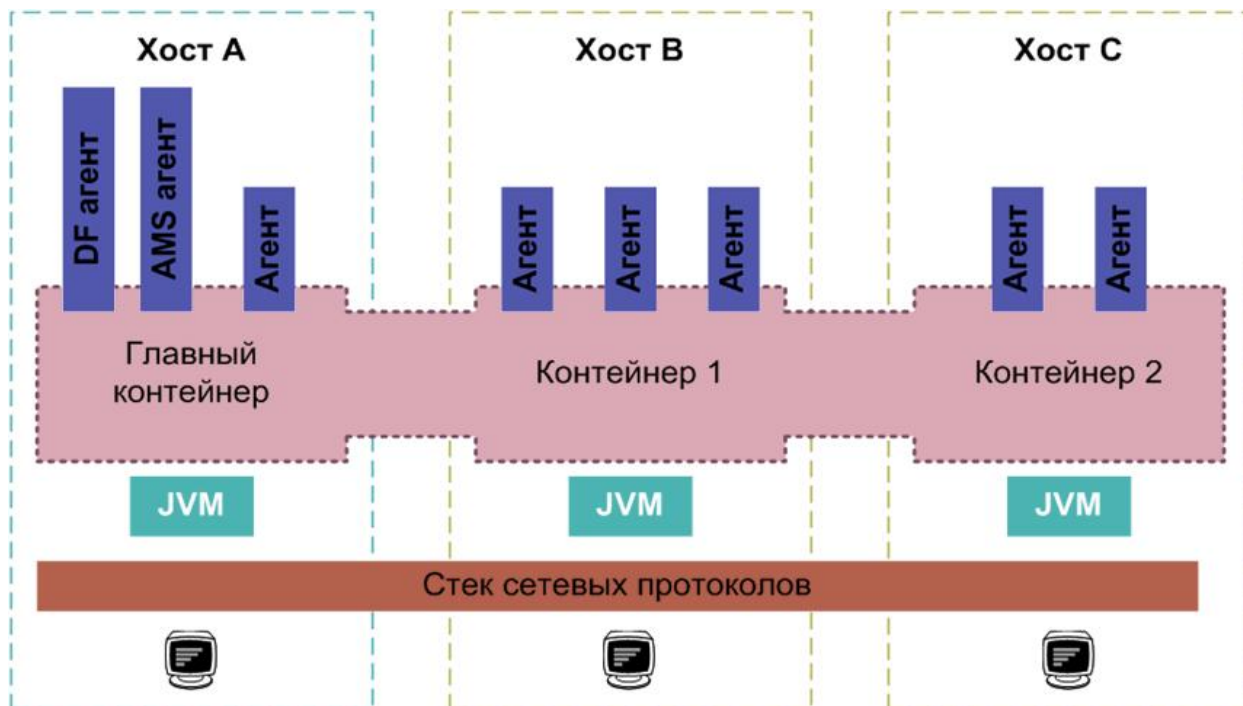


Рисунок 1: Архитектура JADE

Каждый агент - отдельный поток, исполняющийся внутри контейнера агентов, который представляет из себя обычное Java приложение. JADE создает несколько контейнеров для агентов. Вместе набор контейнеров образует платформу. Каждая платформа должна иметь основной контейнер, который содержит два специализированных агента AMS и DF.

Архитектура главного контейнера JADE

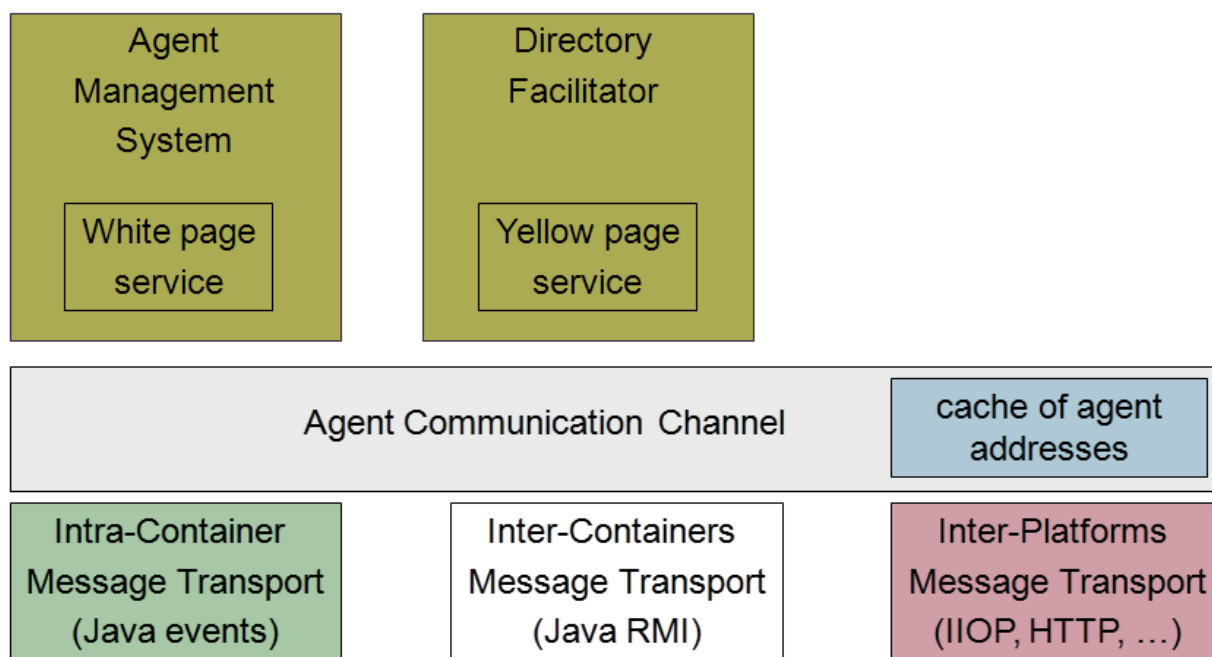


Рисунок 2: Архитектура главного контейнера JADE

Взаимодействия между агентами осуществляются посредством сообщений языка FIPA ACL. [5, 6]

Действия совершаемые агентами описываются с помощью поведений и исполняются последовательно в соответствии с очередью.

Существуют поведения выполняемые один раз, циклические поведения, композитные поведения (поведения состоящие из поведений других типов) и другие.

Агенты автономны, полностью контролируют свои потоки исполнения и сами решают когда и какое сообщение прочесть. Агенты многопоточны, могут быть вовлечены в несколько одновременных переговоров и выполнять несколько параллельных задач.

Взаимодействие агентов и ресурсов в JADE реализовано согласно спецификациям FIPA, работа системы не зависит от физического размещения агентов. Все сообщения передаются с помощью системы MTS.

Сервис передачи сообщений (MTS)

- Управляет очередью частных ACL сообщений агента
- Конкретная реализация выбирается исходя из ситуации
 - Минимизация стоимости передачи сообщения
- Распределенный канал связи
 - Протоколы передачи сообщений (MTP) могут быть активированы/деактивированы во время исполнения на любом контейнере через GUI
 - Несколько способов кодирования ACL: String-based, XML-based, bit-efficient

JADE предоставляет разработчикам некоторые инструменты для администрирования и отладки мультиагентных систем с графическим интерфейсом.

Инструменты:

❖ RMA (Remote Monitoring Agent)

- Просмотр сервиса белых страниц (в т.ч. на удаленных платформах)
- Управление жизненным циклом агента (удаленное создание, миграция, ...)
- Активация/деактивация транспортных протоколов MTP в контейнерах

❖ DF GUI

- Просмотр сервиса желтых страниц
- Создание объединений (federations) DF и просмотр удаленных DF

❖ DummyAgent

- Отправка, получение, сохранение сообщений ACLMessages

❖ Sniffer Agent

- Прослушивание, отладка, сохранение в файл мультиагентных разговоров

❖ **Introspector Agent**

- Отладка агента: очередь отправленных/принятых сообщений, очередь поведений агента, ...

BDI4JADE Фреймворк [7]

BDI4JADE - это надстройка над JADE позволяющая реализовать BDI-архитектуру, сохраняя функционал JADE. По умолчанию JADE ее не реализует.

BDI (belief–desire–intention software model) — это модель разработки интеллектуальных агентов.

BDI представлена следующими сущностями:

- Beliefs (убеждения) — представление агента об окружающем мире
- Desires (желания) — сценарии которые агент хотел бы достичь
- Intentions (намерения) — желания, для которых у агента есть активный план [8]

Архитектура системы

В ходе данной работы была разработана архитектура системы, помогающей в поиске плана развозки товара клиентам. Система обладает мультиагентной архитектурой — состоит из агентов, взаимодействующих друг с другом.

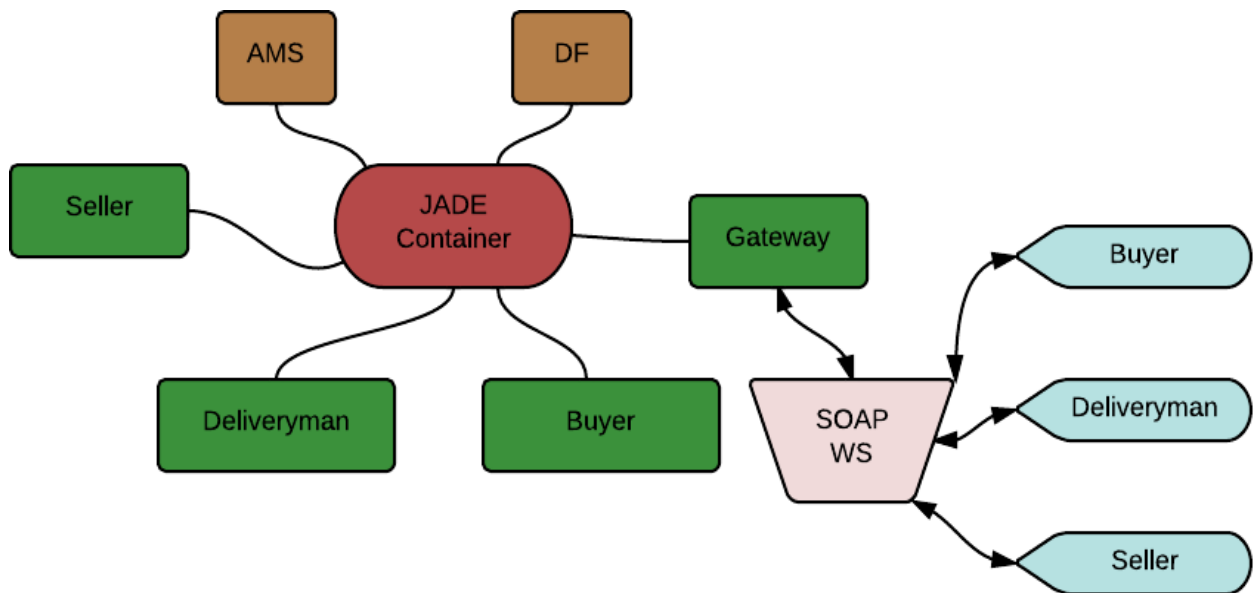


Рисунок 3: Архитектура системы

Типы агентов

Для реализации поставленной задачи в системе было создано 4 основных типа агентов:

- Агент-посредник(Gateway)
- Агент-перевозчик(Deliveryman)
- Агент-покупатель(Buyer)
- Агент-продавец(Seller)

Помимо этого в системе присутствуют стандартные агенты JADE, соответствующие спецификациям FIPA – AMS и DF [9], которые необходимы для управления платформой и агентами.

Агент-посредник

Наша система должна обладать возможностью взаимодействия с пользователем (продавцом, покупателем или перевозчиком). Для этой цели и был создан агент-посредник.

Он умеет получать сообщения из внешнего мира и передавать их агентам платформы, для создания, удаления или изменения агентов. Так же в его обязанности входит отправка ответов системы пользовательскому интерфейсу.

Агент-перевозчик

Агент-перевозчик был создан для представления в нашей системе людей, готовых помочь с развозкой товара. О них мы должны знать лишь маршрут их передвижения. Внутри себя этот агент хранит следующую информацию.

- Свой маршрут (упорядоченный набор географических координат точек через которые он проезжает)
- Список других агентов-перевозчиков с которыми у него имеется хоть одно пересечение маршрутов
- Список агентов-покупателей которые заявили о том, что данный агент может быть им полезен.

Агент-покупатель

Данный агент представляет интересы покупателей, ждущих доставки товара. При создании такой агент заявляет о желании построить маршрут перевозки от точки отгрузки агента-продавца до точки получения товара. Весь жизненный цикл этого агента ориентирован на поиск возможностей реализации желаемого маршрута. Для этого он хранит следующую информацию:

- Координаты точки в которой будет производиться загрузка товара
- Координаты желаемой точки получения товара
- Дерево, состоящее из агентов перевозчиков, построенное на основе пересечения их маршрутов, корнем которого является агент-перевозчик, проезжающий рядом с точкой в которой будет производиться загрузка товара

Агент-продавец

Агент-продавец представляет интересы продавца товара. В его обязанности входит опрос агентов-покупателей на наличие у них найденного плана доставки товара, сбора таких вариантов и выработки общего плана развозки всем заинтересованным покупателям. Хранит в себе следующие данные:

- Точку в которой будет производиться загрузка товара
- Список заинтересованных агентов-покупателей
- Текущую информацию о уже найденных покупателями частных маршрутов

Реализация системы

В данной работе был реализован прототип системы с предложенной архитектурой. Далее будет рассмотрен алгоритм работы системы.

Добавление агента-перевозчика

Добавление происходит через пользовательский интерфейс. Для создания данного типа агента пользователь должен указать маршрут следования этого агента. Маршрут в системе представлен набором двух и более упорядоченных географических координат. Координаты пользователь может указать как в числовом виде так и в виде адресов (например: Россия, Санкт-Петербург, Невский проспект д. 78).

В случае предпочтения пользователем варианта ввода маршрута посредством адресов после ввода данных производится операция геокодирования.

Операция геокодирования — процесс назначения географических координат, выраженных в виде широты и долготы, записям данных. Геокодирование реализовано посредством AJAX запросов к web-сервису Google Maps API.

После того как пользователь так или иначе ввел опорные точки маршрута система с помощью того же web-сервиса строит маршрут, проходящий через эти точки.

Полученный маршрут может содержать большее количество опорных точек чем было введено пользователем т.к. содержит в себе информацию о всех изменениях направления движения, которые происходят на данном маршруте, в то время как пользователь мог указать лишь некоторые опорные точки такие, что прямого проезда между ними не существует.

На рисунке 4 изображена схема добавления агента перевозчика в работающую систему.

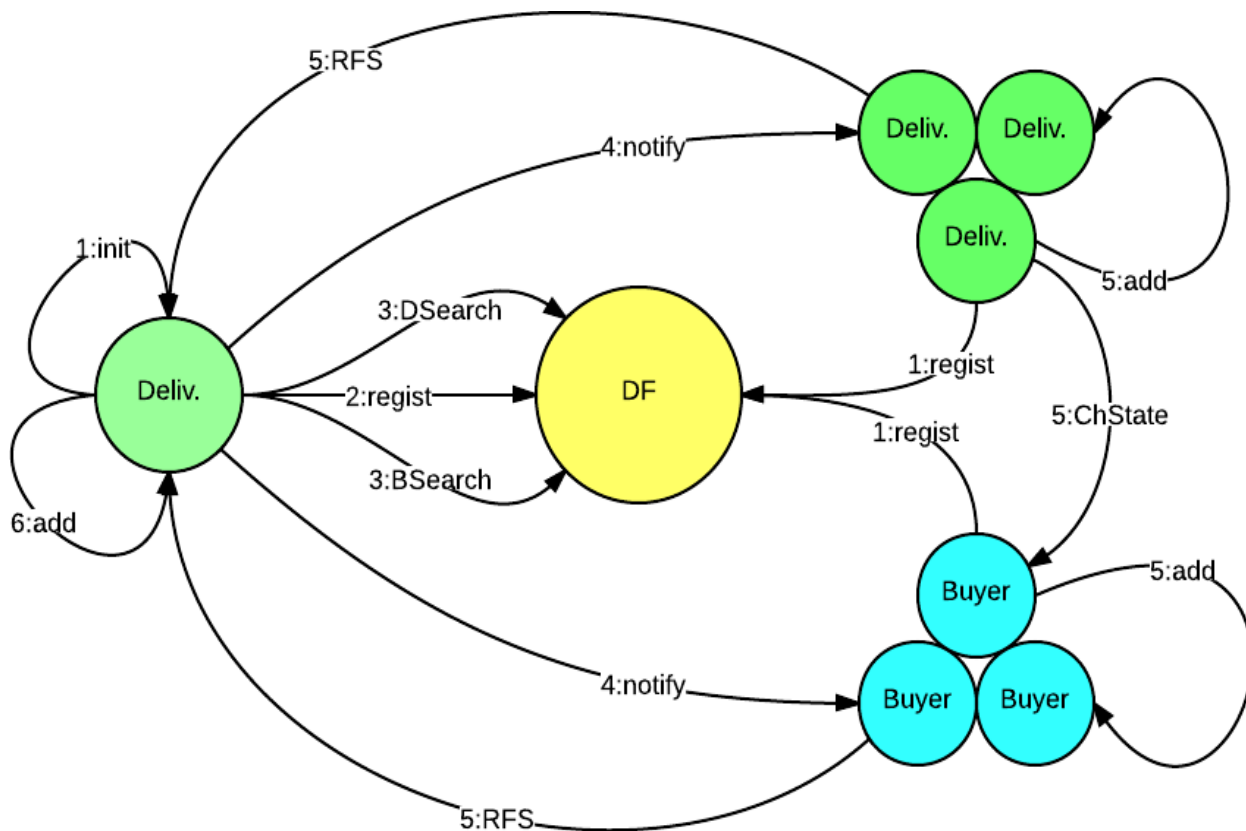


Рисунок 4: Добавление агента-перевозчика

- 1) Первым делом агент-перевозчик производит инициализацию (сохраняет у себя данные о своем маршруте следования). Для наглядности далее полагаем, что на этом этапе в системе уже существуют другие агенты.
- 2) Далее новоиспеченный агент регистрирует себя в сервисе желтых страниц, чтобы другие агенты, которые могут появиться в системе в любой момент смогли о нем узнать.
- 3) Агент обращаясь к сервису желтых страниц запрашивает имена других существующих в системе агентов-перевозчиков и агентов-покупателей.
- 4) Агент используя полученный список существующих отправляет им уведомление о том, что он появился в системе, сообщает свой тип и свои параметры.

5) Каждый из адресатов уведомления после его получения проверяет может ли этот агент быть ему чем-либо полезен. А именно:

- а) Агенты-перевозчики выполняют проверку на наличие пересечений их маршрутов с маршрутом нового агента. В случае если таковых пересечений нет — игнорируют агента. Если такие пересечения были найдены, агенты сохраняют в своем локальном списке пересекаемых перевозчиков имя нового агента и отправляют ему сообщение о том, что пересечение найдено и они хотят сотрудничать. Еще агенты-перевозчики обязаны отправить уведомление о добавлении в свой список пересекаемых перевозчиков нового агента всем агентам-покупателям, которые присутствуют в списке агентов-партнеров.
- б) Агенты-покупатели выполняют проверку на наличие пересечений маршрута нового агента с их пунктом загрузки товара, предоставленным им их агентом-продавцом. В случае если таковых пересечений нет — игнорируют агента. Если такие пересечения были найдены, агенты добавляют в свое локальное дерево пересечений маршрутов агентов нового агента и посылают агенту уведомление о том, что хотят чтобы тот добавил их в свой список партнеров.
- б) Агент принимает и обрабатывает все отправленные ему сообщения, добавляя приславших их агентов в свои соответствующие списки партнеров.

Таким образом агент формирует свое локальное представление об интересной ему части мира. При любом значимом изменении своего состояния он уведомляет об этом своих партнеров.

За счет такого поведения достигается высокая скорость реакции системы на меняющиеся условия задачи, ведь остальным агентам не приходится постоянно следить за ситуацией. Они рассчитывают, что при любом значимом изменении их уведомят об этом и они отреагируют

соответствующим образом, уведомив в свою очередь своих партнеров о своем изменении, если таковое случилось.

Добавление агента-покупателя

Добавление происходит через пользовательский интерфейс. Для создания данного типа агента пользователь должен указать желаемую точку получения товара. Как и в предыдущем примере пользователь может указать координаты в числовом или адресном виде.

На рисунке 5 изображена схема добавления агента покупателя в работающую систему.

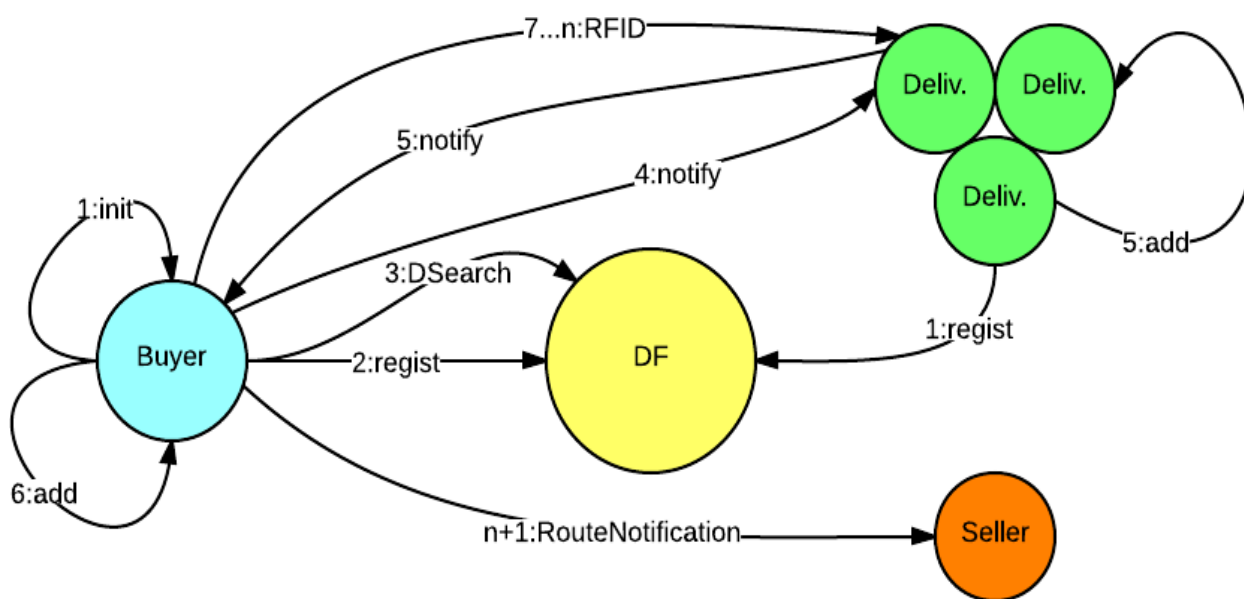


Рисунок 5: Добавление агента-покупателя

- 1) Первым делом агент-покупатель производит инициализацию (сохраняет у себя данные о начальной и конечной точке искомого маршрута). Для наглядности далее полагаем, что на этом этапе в системе уже существуют другие агенты.

- 2) Далее агент регистрируется в сервисе желтых страниц, чтобы другие агенты, которые могут появиться в системе в любой момент смогли о нем узнать.
- 3) Агент обращаясь к сервису желтых страниц запрашивает имена существующих в системе агентов-перевозчиков.
- 4) Далее агент сообщает агентам-перевозчикам о своем появлении в системе и передает им свои параметры (координаты точек начала и конца искомого маршрута).
- 5) Агенты перевозчики, получившие сообщение о появлении в системе нового агента-покупателя, проверяют не проходит ли их маршрут вблизи точки отгрузки товара, которую сообщил им агент-покупатель. Если да — добавляют этого агента в список своих партнеров и отправляют ему сообщение о том, что их маршрут проходит рядом с его точкой отгрузки товара.
- 6) Агент-покупатель получает такие сообщения, добавляет отправителей к вершине своего локального графа пересечений маршрутов агентов-перевозчиков.
- 7) Следующим шагом агент проверяет не проходит ли маршрут кого-нибудь из только что добавленных агентов через желаемую точку получения товара. Если нет — отправляет сообщения только что добавленным агентам с просьбой передать ему имена агентов-перевозчиков, которые являются их партнерами, а значит имеют с ними точки пересечения. Получив ответ добавляет новых агентов в свой локальный граф пересечений и повторяет запрос (6), но уже другим агентам. Если на какой-то итерации агент понимает, что удалось добавить в дерево агента маршрут которого проходит через желаемую точку получения товара, то агент отправляет об этом уведомление агенту продавцу, передавая упорядоченный список агентов-перевозчиков, готовых реализовать данную доставку.

Роль агента-продавца

Как видно из описанного выше агенты-покупатели совместными усилиями с агентами-перевозчиками каждый сам для себя находят варианты маршрутов доставки товара и передают найденные варианты на обработку агенту-продавцу. Продавец лишь выбирает из предложенных вариантов (если их несколько) те с помощью которых можно составить общий план развозки товара, оптимальный по требуемой характеристике. Например по параметру количества задействованных перевозчиков. После чего формирует общий план развозки и уведомляет пользователя о готовности варианта доставки.

Так как может возникнуть ситуация при которой не все агенты-покупатели смогут найти для себя хотя бы один вариант доставки, пользователь может в любой момент сформировать общий план развозки игнорируя тех клиентов, до которых вариантов доставки найдено не было. Таким образом можно частично удовлетворить потребности покупателей, разгрузить квартиру от части товаров и после этого ждать, пока оставшиеся клиенты смогут построить маршрут доставки, или предложить им забрать товар самостоятельно.

Результаты

- Проведен анализ предметной области и потребностей потенциальных пользователей кооперативных перевозок.
- Предложен метод поиска плана перевозок с использованием мультиагентного подхода.
- Разработана архитектура мультиагентной системы поддержки кооперативной логистики.
- Реализован прототип системы с предложенной архитектурой.
- Реализован прототип web-интерфейса системы.

Литература

- [1] Бугайченко Д. Ю., Соловьев И. П. Абстрактная архитектура интеллектуального агента и методы ее реализации // Системное программирование. – 2005. – С. 36-67.
- [2] Michael Wooldridge. An Introduction to MultiAgent Systems. – 2002. – С. 365.
- [3] Форум кооперативных закупок [Электронный ресурс]. – Режим доступа: <http://sptovarov.ru/> (Дата обращения: 19.12.2012).
- [4] Группа кооперативных закупок [Электронный ресурс]. – Режим доступа: <http://vk.com/shopogolike> (Дата обращения: 17.12.2012).
- [5] Bellifemine F.L., Caire G., Greenwood D. Developing Multi-Agent Systems with JADE // Wiley Series in Agent Technology. – 2007. – С. 303.
- [6] JADE фреймворк [Электронный ресурс]. – Режим доступа: <http://jade.tilab.com/> (Дата обращения: 13.04.2013).
- [7] Nunes I., Lucena C. J. P., Luck M. BDI4JADE: a BDI layer on top of JADE. – 2011. – С. 88-103.
- [8] Rao M. P., Georgeff. BDI-agents: From Theory to Practice // Proceedings of the First International Conference on Multiagent Systems. – 1995. – С. 14.
- [9] Спецификации FIPA [Электронный ресурс]. – Режим доступа: <http://www.fipa.org/specs/> (Дата обращения: 12.04.2013).