

Міністерство освіти і науки України
Запорізький державний університет

До захисту допущений
Зав. кафедрою
ММ та ІТ

(підпис)

Борю Сергій Юрійович
(прізвище, ім'я, по батькові)

(дата)

ДИПЛОМНА РОБОТА

НА ТЕМУ: РОЗРОБКА МОДЕЛЕЙ І АЛГОРИТМІВ КООРДИНАЦІЇ
В СЦЕНАРІЯХ ДОКУМЕНТООБИГУ ВУЗУ

Виконав Плаксін Сергій Леонідович

ст. групи ПМ-2 С. Л. Плаксін
(шифр) (підпис і дата) (ім'я, по батькові, прізвище)

Керівник доцент В. А. Єрмолаєв
(посада) (підпис і дата) (ім'я, по батькові, прізвище)

Нормоконтролер І. А. Костюшко
(підпис) (ім'я, по батькові, прізвище)

Запоріжжя
2001

ЗАВДАННЯ

РЕФЕРАТ

Дипломная работа: 64 с., 13 рис., 20 источников.

Объект исследования - модели и алгоритмы координации в динамических мульти-агентских системах.

Цель работы - разработка математической модели и алгоритмов координации взаимодействия программных агентов, участвующих в выполнении заданий в сценариях документооборота ВУЗа.

Методы исследования - теоретические методы построения программных агентов и мультиагентских систем; методы дискретной математики; методы проектирования архитектуры программного обеспечения.

Данная дипломная работа посвящена актуальной теме исследований в области математического моделирования сложных интеллектуальных программных систем: разработке и алгоритмической реализации моделей координации в динамических, интеллектуальных, распределенных программных системах, построенных на базе агентского подхода. Главной задачей данной работы является разработать модели и алгоритмы координации взаимодействия агентов в процессе переговоров о распределении работ, в процессе выполнения работ, мониторинга взаимодействия исполнителей работ.

АГЕНТ, МУЛЬТИ-АГЕНТНАЯ СИСТЕМА, ПЕРЕГОВОРЫ, ДОКУМЕНТООБОРОТ, СЦЕНАРИЙ ДОКУМЕНТООБОРОТА, ВОЗДЕЙСТВИЕ, КООРДИНАЦИЯ АГЕНТОВ, КООРДИНАЦИОННЫЙ АГЕНТ.

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ	6
ВВЕДЕНИЕ	7
1 КООРДИНАЦИЯ В УПРАВЛЕНИИ И ПРИНЯТИИ РЕШЕНИЙ	10
1.1 Понятие о координации, актуальные приложения	10
1.2 Проблемы координации	11
1.2 Проблема координации в MAS	11
1.3 Известные методы координации в MAS	17
1.4 Задача моделирования координации в динамических сообществах агентов.....	22
1.5 Актуальность решения проблемы координации для реализации сценариев документооборота	23
1.6 Реализация документооборота предприятия с использованием технологии интеллектуальных агентов (краткий обзор проектов)	26
1.7 Вывод.....	28
2 МОДЕЛИ И АЛГОРИТМЫ КООРДИНАЦИИ	29
2.1 Динамические сообщества агентов для выполнения заданий.....	29
2.2 Виды координации в динамическом сообществе агентов.....	29
2.3 Модели координации и алгоритмы	31
2.3.1 Координация на стадии предварительного взаимодействия (переговоры).....	31
2.3.1.1 Объект переговоров	32
2.3.1.2 Протокол переговоров	34
2.3.1.3 Некоторые определения и понятия	37
2.3.1.4 Модель поведения агента-исполнителя в процессе переговоров	41
2.3.1.5 Модель поведения агента-инициатора в процессе переговоров	44
2.3.1.6 Алгоритм взаимодействия агентов при распределении работ	47
2.3.2 Координация на стадии выполнения работы	50
2.3.3 Мониторинг основных параметров взаимодействия агентов	54
2.4 Вывод.....	56
3 СЦЕНАРИИ ДОКУМЕНТООБОРОТА ВУЗА	58
3.1 Демонстрация применения модели координации сообщества агентов на сценариях документооборота ВУЗа	58
3.1.1 Модельная задача координации распределения работ при приеме документов у абитуриента	58
3.1.2 Модельная задача координации распределения работ во множительном центре ВУЗа	59
ЗАКЛЮЧЕНИЕ	61
СПИСОК ССЫЛОК	62

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

ACL - язык взаимодействия агентов (Agent Communication Language)

AI - искусственный интеллект

COA - координационный агент

KIF - формат обмена знаниями (Knowledge Interchange Format)

KQML - язык запросов и манипуляцией знаниями (Knowledge Query and Manipulation Language)

MAS - мультиагентская система (multi-agent system)

ВВЕДЕНИЕ

Проблема документооборота в последнее время всё чаще и чаще поднимается на страницах деловой и компьютерной прессы. Действительно, актуальность этой проблемы достаточно велика. Создание эффективной системы, ориентированной на оптимизацию процессов циркуляции документопотоков в рамках той или иной структуры, является сегодня одним и основополагающим фактором повышения конкурентоспособности фирмы, организации, учреждения. Современный уровень развития науки и техники позволяет решить эту проблему наиболее эффективным образом, а именно - с использованием передовых компьютерных и информационных технологий.

Систему электронного документооборота следует рассматривать как составную часть системы управления потоками работ (WorkFlow), т.е. системы, ориентированной на управление бизнес-процессами.

На сегодня существует программное обеспечение, предназначенное для автоматизации документооборота. В качестве примера подобного программного обеспечения можно назвать GroupWise, Optima WorkFlow, FlowMark, Derlina FormFlow, StaffWare, Action WorkFlow, KeyFlow и т.д. Перечисленные системы в той или иной степени позволяют автоматизировать документооборот.

Общими принципами построения подобных систем являются:

- централизованное управление (все они построены с использованием архитектуры «клиент-сервер»);
- статические, предопределённые сценарии документооборота (все сценарии в таких системах хранятся на сервере, и изменить их динамически нельзя);
- гомогенная структура компонентов системы.

Поэтому подобные системы не вполне пригодны для реализации динамических сценариев документооборота.

Кроме того, традиционные подходы к автоматизации документооборота, используемые в существующих системах, часто не применимы в

крупномасштабных и динамичных проектах. Это объясняется двумя основными причинами:

- стремлением к распределению и глобализации современных бизнес-процессов;
- высокой степенью динамичности современных бизнес-процессов.

Поэтому возникает необходимость в создании нового подхода к автоматизации документооборота. Таким подходом является идея использования технологии интеллектуальных программных агентов в системах workflow.

Суть этого подхода состоит в том, что распределением и выполнением подзадач, составляющих бизнес-процесс (или поток работ) занимаются агенты.

Преимущества этого подхода заключаются в том, что в такой системе нет жестких сценариев документооборота, то есть можно динамически, в процессе работы модифицировать существующие сценарии и добавлять новые. Гетерогенность агентов, составляющих сообщество, позволяет объединять агентов с разными архитектурами (чего нельзя сделать в традиционном подходе). Кроме того, использование агентской технологии позволяет достаточно просто масштабировать всю систему. Это означает, что такая система будет быстро реагировать на динамические изменения в реальности.

Динамическое сообщество агентов, реализованное для выполнения сценариев документооборота, с использованием моделей, предложенных в [1, 2, 3, 4] является децентрализованной системой, ориентированной на выполнение работ из потока. Поток работ, составляющих сценарий, образуют бизнес процесс. Между работами бизнес процесса всегда есть зависимости, влияющие на последовательность их выполнения. Поэтому агентов выполняющих работы из сценария документооборота необходимо координировать. То есть, необходим механизм, обеспечивающий правильную и непротиворечивую последовательность выполнения работ.

В такой динамической системе без централизованного управления возникает проблема распределения работ между агентами, более того эффективного распределения работ.

Распределением работ между собой агенты занимаются самостоятельно. Поэтому для эффективного распределения работ агенты должны взаимодействовать друг с другом – вести переговоры.

Эффективность распределения работ влияет на результат выполнения всего сценария документооборота. Поэтому механизм переговоров, обеспечивающий распределение работ является одним из типов координации в динамических сообществах агентов.

Задачами дипломной работы являются: разработать модели и алгоритмы координации взаимодействия агентов в процессе переговоров о распределении работ, в процессе выполнения работ, мониторинга взаимодействия исполнителей работ. А также опробовать разработанные модели и алгоритмы на модельных сценариях документооборота ВУЗа: прием документов у абитуриентов, печать документов в множительном центре

1 Координация в управлении и принятии решений

1.1 Понятие о координации, актуальные приложения

В последнее время среди исследователей различных дисциплин, таких как информатика, теория управления, экономика, психология, возрастает интерес к проблеме координации сложных систем. В каждой из этих областей координации требуют либо распределённые компьютерные системы, либо группа людей, либо и те и другие одновременно.

Понятие координации, является междисциплинарным понятием. Однако, несмотря на это на сегодня нет единого и общепринятого определения термину координация.

Каждый из нас имеет своё интуитивное понятие о координации. Так, например, когда мы видим хорошо организованную конференцию, или спортивную команду, победившую в соревновании, или функционирующую сборочную линию на заводе мы говорим о хорошей координации в группе людей и компьютерной системе. Наиболее явно мы замечаем координацию, когда её недостаёт. Так, когда мы тратим время в ожидании на взлетно-посадочной полосе аэропорта, пока диспетчер найдет свободный воздушный коридор для нашего самолёта, или когда в гостинице, в которой мы хотели остановиться, заняты все номера, или когда наш текстовый редактор отказывается работать в новой версии операционной системы, мы говорим о плохой координации.

Часто интуитивного определения координации достаточно, но для того чтобы исследовать проблему координации необходимо ввести точное определение.

Большинство исследователей, занимающихся этой проблемой, под координацией понимают процесс управления зависимостями между действиями [5].

Зависимости между действиями происходят во многих типах систем – социальных, компьютерных, биологических и т. д. Например, вопросами

относительно зависимостей между действиями людей занимаются теория управления, социология, экономика, социальная психология, политология и т. д. Зависимостями между действиями в компьютерных системах занимаются компьютерные науки.

1.2 Проблемы координации

Основной проблемой процесса координации является управление зависимостями между действиями элементов сложных систем. В качестве причин, по которым возникают зависимости между действиями можно выделить следующие:

- распределение общего ресурса;
- действия связаны отношением производитель/потребитель (когда результат одного действия необходим для выполнения другого);
- ограничения на одновременное выполнение действий (когда действия должны или, наоборот, не должны выполняться параллельно);
- группа действий является подзадачей для некоторой общей задачи.

Кроме того, в каждой конкретной системе могут возникать специфические зависимости между действиями.

1.2 Проблема координации в MAS

В последнее время резко возросло число компьютерных систем, разработанных и реализованных на базе теоретических моделей, архитектур и программных технологий, использующих концепцию агентов и мультиагентских систем. Использование технологии агентов упрощает процесс разработки программного обеспечения. Прикладные программы, написанные как программные

агенты, могут взаимодействовать с другими программными агентами, обмениваясь сообщениями на языке взаимодействия агентов. Агенты могут быть столь же простыми, как и подпрограммы, однако обычно агентами являются довольно крупные программные комплексы. Агенты применяются в областях, начиная с относительно простых персональных помощников, которые, например, фильтруют электронную почту, заканчивая сложными информационными системами, которые управляют воздушным движением, занимаются диагностикой неисправностей на атомных электростанциях и т. п.

Несмотря на такую распространённость на сегодня нет единого и общепринятого определения термину агент.

Большинство исследователей в этой области придерживаются следующего определения, данного Дженнингсом, Вулдриджем [6]: агент - объект для решения некоторой задачи (реализованный аппаратными средствами, программными средствами или их комбинацией), который обладает следующими свойствами:

- автономностью: агенты должны быть способны выполнять свои задачи без прямого вмешательства человека или других агентов, а также иметь определённую степень контроля над своими действиями и собственным внутренним состоянием;
- способностью взаимодействовать: агенты должны быть способны взаимодействовать с другими агентами или человеком для решения своих задач;
- реактивностью: агенты должны реагировать на изменения, происходящие в их среде;
- агент должен не просто реагировать на изменения в среде, его действия должны быть направлены некоторой целью.

С точки зрения решения проблемы координации в распределённых многоагентских системах имеет также смысл обратить внимание на определение агента в системе, данное Лессером в [7]: "Агент в системе представляет собой узел, предназначенный для активного решения проблем ..."

Однако при решении сложных задач, использование автономных (одиночных) агентов не всегда эффективно и поэтому используют мультиагентские системы

(MAS). Использование MAS позволяет решать более сложные, реалистические и крупномасштабные проблемы, которые не поддаются решению при помощи автономных агентов. Проблема моделирования деятельности реальных предприятий является одной из сфер применения интеллектуальных информационных агентов и мультиагентских систем.

Первоначально мультиагентские системы рассматривались в изоляции от их сред со статическим набором участников (агентов) и целей. Это усложняло применение технологии агентов в быстро изменяющихся, динамических предметных областях.

Открытая система (следуя определению Скотта [8]) - система, в которой структура и состав самой системы способна динамически изменяться. В открытой системе ее компоненты не известны заранее, могут изменяться через какое-то время и могут состоять из гетерогенных агентов, созданных разными людьми, в разное время, при помощи различных программных инструментальных средств и методов. Возможно, наиболее ярким примером такой системы является интернет. В подобной среде информационных источников агенты могут появляться и исчезать неожиданно. В настоящее время, агенты в интернет главным образом исполняют информационный поиск и фильтрацию. Эти задачи, как отмечает, например, Лессер [7] требуют, чтобы агенты были способны взаимодействовать друг с другом. Кроме того, способность агентов к координации совместных действий позволит агентам расширить круг решаемых ими задач.

В [1, 2, 3, 4] представлен подход к моделированию разнообразных процессов, протекающих в реальных/виртуальных предприятиях, базирующийся на динамических сообществах интеллектуальных программных агентов, ориентированных на выполнение заданий. Агенты в таком сообществе объединяются во временный консорциум автономных, возможно, географически рассредоточенных функциональных компонент, которые объединяют свои ресурсы и опыт для решения общих задач.

Растущая популярность MAS при построении сложных систем объясняется рядом преимуществ, которыми обладает MAS перед традиционными подходами. Использование MAS при построении сложных информационных систем позволяет:

- а) решить проблемы, которые являются слишком сложными для одиночного агента;
- б) решить проблемы, связанные с ограниченными ресурсами или явным наличием критических параметров эффективности;
- в) сохранять соответствие программного обеспечения реальному миру. Для этого система должна периодически модифицироваться. Полная перезапись такого программного обеспечения, как правило, очень дорогостоящая, а часто просто невозможна. Модифицировать аналогичную систему, включающую сообщество агентов, гораздо проще, так как необходимо изменить лишь отдельного агента/агентов.
- г) решить проблемы, которые могут естественно быть расценены как сообщество автономных взаимодействующих агентов (например, управление воздушным движением, моделирование деятельности предприятия, и т. п.);
- д) обеспечить решения проблем, которые эффективно используют информационные источники, которые пространственно распределены. Примерами таких областей могут быть сети датчиков, сейсмический контроль, информация, накапливающаяся из интернет.
- е) обеспечить решения в ситуациях, где обработка, знания распределены. Примеры таких областей включают параллельную разработку, здравоохранение и производство.
- ж) увеличить эффективность системы по таким аспектам:
 - *скорость вычислений*, так как используется параллелизм;
 - *надежность*, то есть простое восстановление данных при сбоях;
 - *масштабируемость*, потому что число и возможности агентов, работающих над проблемой, может легко меняться;
 - *ошибкоустойчивость*, способность системы допустить неопределенность, потому что агенты обмениваются необходимой информацией;

- *удобство сопровождения*, потому что систему, состоящую из множественных агентов проще поддержать из-за ее модульности;
- *быстрота реагирования*, потому что модульность позволяет обрабатывать аномалии в местном масштабе, не размножая их по целой системе;
- *гибкость*, потому что агенты с различными способностями могут адаптивно объединяться для решения текущих проблем;
- *многократное использование*, потому что функционально определенные агенты могут многократно использоваться в различных сообществах агентов для решения различных проблем.

Однако, несмотря на все перечисленные преимущества, проектирование и формирование систем, использующих MAS трудно. Такие системы не лишены проблем, связанных с формированием традиционных распределённых параллельных систем и дополнительных трудностей, которые являются результатом наличия гибких и сложных взаимодействий между автономными проблемными компонентами. Основной проблемой при использовании MAS является обеспечение непротиворечивого и эффективного взаимодействия автономных компонентов - агентов.

Одной из главных проблем, возникающих при использовании MAS, является координация агентов в сообществе - обеспечение их непротиворечивой и бесконфликтной работы. В [9] приведён пример, демонстрирующий необходимость координации динамического сообщества агентов.

Проблема координации возникает в следующих ситуациях:

- при выполнении некоторой задачи у агента есть выбор в действиях, и этот выбор влияет на эффективность результата выполнения задачи;
- порядок, в котором агент выполняет действия, влияет на эффективность результата;
- время, в течение которого агент выполняет действие, влияет на эффективность результата.

Как отмечает К. Сикара [10], изучение мультиагентских систем сосредотачивается на системах, в которых много интеллектуальных агентов

взаимодействуют друг с другом. Агенты, как рассматривается, являются автономными объектами, типа программ или роботов. Их взаимодействия могут быть или кооперативными или эгоистичными. То есть агенты могут совместно преследовать общую цель (как в колония муравьёв), или они могут преследовать собственные интересы (как в свободной рыночной экономике). Поэтому необходимо реализовать каждого муравья в колонии так, чтобы заставить их всех приносить продовольствие наиболее эффективным способом, или установить правила взаимодействия так, чтобы группа эгоистичных агентов работала вместе для выполнения поставленной задачи.

Таким образом, координация - центральная проблема в MAS в частности и в распределенном искусственном интеллекте (DAI) вообще. По существу, координация - *процесс*, в котором агенты участвуют, чтобы обеспечить сообществу непротиворечивую, согласованную последовательность действий. Это означает, что действия агентов спланированы так, что они не конфликтуют друг с другом, а всё сообщество ведёт себя как единое целое. Имеется несколько причин, по которым MAS должна быть скоординирована [11]:

- *Предотвращение анархии или хаоса*: координация необходима или желательна, так как в децентрализованной MAS анархия может установиться легко;
- *Наложение глобальных ограничений*: всегда существуют глобальные ограничения, которые MAS должна удовлетворить, чтобы успешно выполнить задание;
- *Распределённые знания, ресурсы, информация*: практически всегда в MAS имеются ресурсы, работа с которыми требует координации;
- *Зависимость между действиями агентов*: цели агентов часто взаимосвязаны.
- *Эффективность*: даже когда агенты могут работать независимо, координация позволяет сэкономить время.

Итак, механизм координации агентов в сообществе должен обеспечивать бесконфликтную последовательность выполнения работ агентами и среду для обмена промежуточными результатами.

1.3 Известные методы координации в MAS

Одним из видов классификации методов координации агентов в мультиагентских системах является их деление на централизованные и децентрализованные.

Централизованные методы координации состоят в том, что в мультиагентской системе имеется агент, который обладает знаниями о всей системе, распределяет задания между агентами и координирует их выполнение. Однако этот агент является критическим параметром всей системы.

Поэтому при проектировании систем без такого критического параметра, большинство исследователей в области DAI сконцентрировались на развивающихся сообществах, в которых и управление и данные распределены, а методы координации – децентрализованы.

При этом важно чтобы время, затраченное на координацию агентов в такой системе, было много меньше, чем время, затраченное на решение задач, для выполнения которых была создана эта система.

Недостатком распределённого управления и данных является то, что информация о полном состоянии системы распределена по всему сообществу, и каждый агент обладает лишь частичной и неточной информацией о дальнейшем развитии системы.

Существует и другая, предложенная в [11], классификация подходов обеспечения координации в MAS. В [11] методы координации классифицируются следующим образом:

- организационное структурирование;
- заключение контракта;
- планирование деятельности MAS;
- переговоры.

Рассмотрим каждый из этих подходов подробнее.

Организационное структурирование

Это самый простой метод координации, состоящий в предопределённых и долгосрочных отношениях между агентами (Durfee [11]). При этом часто используют иерархические структуры *master-slave* или *client-server*. Эта методика используется в двух вариантах:

- Главный агент планирует и распределяет задания (работы) между подчинёнными агентами. Подчинённые агенты, в конечном счете, должны сообщить главному агенту о результатах их работы. Кроме того, в отличие от главного агента, подчинённые агенты имеют частичную автономность.
- Использование классической доски объявлений - общего информационного поля - для обеспечения координации. В этой схеме агенты используют доску для обмена информацией. Главный агент (планировщик) назначает агентам операции чтения/записи с доской. Эта схема используется Веркман в его DFI системе [11]. Этот подход может использоваться, когда задача распределена, имеется центральный агент планирования или когда задачи уже были назначены, *априорно*, агентам. Sharp Multi-Agent Kernel (SMAK) [11] тоже использует эту стратегию.

Последний вариант показывает, что координация в организационном структурировании не всегда связана с иерархией. Например, в системе DVMT [11], использующей технологию доски координация проходит среди равных агентов.

Узким местом в системах, основанных на технологии доски без прямого взаимодействия агентов, может быть большое количество агентов в сообществе, даже в случае секционированной доски. Кроме этого все агенты должны иметь общее представление о доске. Поэтому большинство систем, основанных на этой технологии, используют небольших гомогенных агентов (как, например DVMT).

Дурфи [11] считает, что такое централизованное управление, как в иерархической методике, противоречит основным предположениям о DAI. В этой методике предполагается, что, по крайней мере, один агент имеет глобальное представление всего сообщества, однако во многих областях это не реально. Если всё-таки эта методика координации используется, проектировщик должен

гарантировать, что все подчинённые агенты имеют оптимальную степень детализации чтобы время, затраченное на декомпозицию общей задачи на более мелкие, не превысило времени её выполнения одним агентом.

Заключение контракта

В этом подходе, который предполагает децентрализованную структуру, агенты принимают две роли:

- менеджер, который делит поступившую задачу на подзадачи и ищет исполнителя, чтобы выполнить их;
- исполнитель, выполняет подзадачу. Однако исполнитель может рекурсивно стать менеджером и разбить поступившую подзадачу на ещё более мелкие задачи и поручить их другим агентам.

Поиск исполнителя выглядят таким образом (FIPA CNP [12]):

1. Менеджер объявляет задачу;
2. Исполнители оценивают эту задачу с точки зрения возможности её выполнить;
3. Менеджер получает таблицу исполнителей;
4. Оценивает полученные предложения, выбирает исполнителя и поручает выполнить ему задачу;
5. Ожидание результатов выполнения задачи.

Это полностью распределённая схема, в которой каждый агент может быть как менеджером, так и исполнителем. Этот подход применяется во многих приложениях.

Хухнс и Сингх [11] обращают внимание на то, что эта модель координации обеспечивает распределение общей задачи, и средства для самоорганизации группы агентов, и лучше всего применима когда:

- прикладная задача имеет четкий иерархический характер (природу);
- прикладная задача делима на крупные подзадачи;
- имеется минимальная взаимосвязь среди подзадач.

Преимущества этого подхода координации состоят в следующем: динамическое распределение задачи благодаря поиску оптимального исполнителя;

сбалансированная загрузка агентов (занятые агенты могут не участвовать в выполнении общей задачи); и надежный механизм для распределенного управления и восстановление результатов при сбое.

Агенты в такой системе довольно пассивны и не применимы для многих прикладных задач. Наконец, такое сообщество агентов довольно интенсивно использует сеть, что может снизить все его преимущества.

Планирование деятельности MAS

Имеется два типа планирования деятельности MAS:

- централизованное планирование деятельности MAS;
- распределенное планирование деятельности MAS.

В централизованном планировании деятельности MAS обычно имеется координационный агент, который, получив все индивидуальные планы от остальных агентов, анализирует их, чтобы обнаружить потенциальные противоречия и конфликты во взаимодействии агентов (например, конфликт между агентами по использованию общего ресурса). Потом координационный агент пытается изменить эти индивидуальные планы и объединяет их в план MAS, в котором противоречивые взаимодействия устранены. В заключительном плане MAS, добавлены команды связи, чтобы синхронизировать взаимодействия агентов в потенциально возможных конфликтах.

В распределенном планировании деятельности MAS, идея состоит в том, чтобы обеспечить каждого агента моделью планов других агентов. Агенты взаимодействуют, чтобы создавать и модифицировать их индивидуальные планы, не конфликтующие с планами других агентов.

Координация в распределенном планировании деятельности MAS - намного сложнее, чем в централизованной форме, так как там не существует агентов, обладающих глобальным представлением обо всей распределённой системе.

Переговоры

Очевидно, чтобы достичь координации в динамическом сообществе агенты должны взаимодействовать друг с другом. Наиболее мощным механизмом

взаимодействия агентов являются переговоры – процесс, благодаря которому группа агентов переходит к взаимоприемлемому соглашению по некоторому вопросу.

Дженнингс Н. Р. в [13] приводит следующую структуру процесса переговоров. Любой процесс переговоров состоит из таких компонентов:

- протокола переговоров – набора правил, по которым происходит взаимодействие агентов. Сюда входит количество участников, типы участников, состояния переговоров, правила, по которым изменяются состояния переговоров, возможные действия участников в каждом состоянии переговоров;
- объекта переговоров – диапазон проблем, по которым должно быть достигнуто соглашение. В общем случае участникам можно динамически изменять объект переговоров;
- модели принятия решения агентом – аппарат принятия решения, который используют участники, чтобы действовать в соответствии с протоколом переговоров.

Как видно, реализовать координацию динамического сообщества агентов при помощи процесса переговоров достаточно сложно. Кроме того, при координации агентов этим подходом часто используют разные методы AI, логики, случайно основанного рассуждения, направленного ограничением поиска, и т.д.

В статье Дженнингса [14] рассматривается модель координации, основанная на гипотезе “Centrality of Commitments and Conventions Hypothesis”, утверждающей, что все механизмы координации могут быть сведены, в конечном счете, к обязательствам (commitments) и к связанным с ними соглашениям (conventions).

Обязательства рассматриваются как обещание (заявление) выполнить некоторое действие, в то время как соглашения обеспечивают средства контроля обязательств в изменяющихся условиях (обстоятельствах).

Для эффективной работы при изменяющихся внешней среде и целях агент должен обладать механизмом проверки, действительны ли ещё существующие обязательства.

Каждый из существующих методов координации пригоден только для предметных областей определённого типа, и не существует универсального метода координации одинаково подходящего к любой предметной области.

1.4 Задача моделирования координации в динамических сообществах агентов

Как и в квалификационной работе бакалавра [9] будем рассматривать динамические сообщества интеллектуальных программных агентов подход к моделированию которых предложен в [1, 2, 3, 4].

Агенты в таком сообществе объединяются во временный консорциум автономных, возможно, географически рассредоточенных функциональных компонент, которые объединяют свои ресурсы и опыт для решения общих задач.

В [9] рассмотрена модель координации сообщества агентов основанная на использовании координационного агента. Однако эта модель не отражает процесс распределения заданий между агентами в сообществе. В дипломной работе будет предложено использование процесса переговоров как мощного средства взаимодействия агентов на стадии распределения заданий. Кроме того на процесс распределения заданий могут влиять такие параметры взаимодействия агентов [1, 2, 3, 4] как матрица состояний системы, матрица невыполненных работ, матрица задержки работ и др.

То есть можно говорить о трёх видах координации в динамическом сообществе интеллектуальных программных агентов:

- координация на стадии распределения заданий, основанная на переговорах;
- координация выполнения заданий, основанная на использовании координационного агента в качестве активной доски объявлений;
- мониторинг основных параметров активности агентов в сообществе.

Все эти виды координации и модели поведения агентов в процессе переговоров будут описаны в разделе 2.

1.5 Актуальность решения проблемы координации для реализации сценариев документооборота

Проблема документооборота в последнее время всё чаще и чаще поднимается на страницах деловой и компьютерной прессы. Действительно, актуальность этой проблемы достаточно велика. Создание эффективной системы, ориентированной на оптимизацию процессов циркуляции документопотоков в рамках той или иной структуры, является сегодня одним из основополагающим фактором повышения конкурентоспособности фирмы, организации, учреждения. Современный уровень развития науки и техники позволяет решить эту проблему наиболее эффективным образом, а именно - с использованием передовых компьютерных и информационных технологий.

Однако, как показывает опыт, автоматизация документооборота обеспечивает решение лишь частных проблем, стоящих перед предприятием. Поэтому имеет смысл говорить об использовании более глобальной системы, которая в качестве одной из подсистем имела бы в своём составе систему электронного документооборота.

Такой системой является система управления потоками работ, или, если пользоваться общепринятой терминологией, система WorkFlow.

Основная концепция автоматизации документооборота должна базироваться на принципах, согласно которым управлению и планированию подлежат, в первую очередь, производственные процессы (бизнес-процессы, потоки работ), в ходе выполнения которых создаются и движутся документы. Это позволяет рассматривать автоматизацию документооборота не как абстрактную цель, а как

эффективное средство выполнения предприятием или организацией своей основной функции - производственной или управленческой работы.

Иными словами, это не просто автоматизация маршрутизации движения документов, но и организация и управление полномасштабными технологическими процессами создания документов на основе последовательной обработки различными пользователями других ранее существовавших документов, структурно и логически отличающихся друг от друга, сопровождающееся модификацией их вида, назначения и свойств. С учетом этого, маршрутные схемы документооборота могут пониматься как технологические карты процессов формирования и обработки документов во всем их многообразии.

Из всего сказанного выше следует, что систему электронного документооборота следует рассматривать как составную часть системы управления потоками работ (WorkFlow), т.е. системы, ориентированной на управление бизнес-процессами.

На сегодня существует программное обеспечение, предназначенное для автоматизации документооборота. В качестве примера подобного программного обеспечения можно назвать GroupWise, Optima WorkFlow, FlowMark, Derlina FormFlow, StaffWare, Action WorkFlow, KeyFlow и т.д. Перечисленные системы в той или иной степени позволяют автоматизировать документооборот.

Общими принципами построения подобных систем являются:

- централизованное управление (все они построены с использованием архитектуры «клиент-сервер»);
- статические, предопределённые сценарии документооборота (все сценарии в таких системах хранятся на сервере, и изменить их динамически нельзя).

Поэтому подобные системы не вполне пригодны для реализации динамических сценариев документооборота.

Кроме того, традиционные подходы к автоматизации документооборота, используемые в существующих системах, часто не применимы в крупномасштабных и динамичных проектах. Это объясняется двумя основными причинами:

- стремлением к распределению и глобализации современных бизнес-процессов;
- высокой степенью динамичности современных бизнес-процессов.

Поэтому возникает необходимость в создании нового подхода к автоматизации документооборота. Таким подходом является идея использования технологии интеллектуальных программных агентов в системах workflow.

Суть этого подхода состоит в том, что распределением и выполнением подзадач, составляющих бизнес-процес (или поток работ) занимаются агенты. Более подробно о использовании интеллектуальных программных агентов при автоматизации документооборота будет сказано в разделе 3.

Преимущества этого подхода заключаются в том, что в такой системе нет жестких сценариев документооборота, то есть можно динамически, в процессе работы модифицировать существующие сценарии и добавлять новые. Кроме того, использование агентской технологии позволяет достаточно просто масштабировать всю систему. Это означает, что такая система будет быстро реагировать на динамические изменения в реальности.

Динамическое сообщество агентов, реализованное для выполнения сценариев документооборота, с использованием моделей, предложенных в [1, 2, 3, 4] является децентрализованной системой, ориентированной на выполнение работ из потока. Поток работ, составляющих сценарий, образуют бизнес процесс. Между работами бизнес процесса всегда есть зависимости, влияющие на последовательность их выполнения. Поэтому агентов выполняющих работы из сценария документооборота необходимо координировать. То есть, необходим механизм, обеспечивающий правильную и непротиворечивую последовательность выполнения работ.

В такой динамической системе без централизованного управления возникает проблема распределения работ между агентами, более того эффективного распределения работ.

Распределением работ между собой агенты занимаются самостоятельно. Поэтому для эффективного распределения работ агенты должны взаимодействовать друг с другом – вести переговоры.

Эффективность распределения работ влияет на результат выполнения всего сценария документооборота. Поэтому механизм переговоров, обеспечивающий распределение работ является одним из типов координации в динамических сообществах агентов.

Задачей дипломной работы является разработка математических моделей и алгоритмов поведения агентов, участвующих в процессе переговоров.

1.6 Реализация документооборота предприятия с использованием технологии интеллектуальных агентов (краткий обзор проектов)

Методы реализации документооборота с использованием технологии интеллектуальных агентов в [15] классифицируются следующим образом:

1. Агенты как сотрудничающие компоненты (автономные агенты): В этом подходе, различные типы агентов разработаны для выполнения различных задач и выполняют различные роли. Например, специализированный агент для тиражирования документов выполняет задачи по тиражированию документов и, взаимодействуя с другими агентами, участвует в сценариях документооборота, в которых присутствуют подобные работы. Разработанные таким образом workflow системы пригодны для любой предметной области. Этот подход отражает непосредственно организационную структуру бизнес-процесса. Такие системы ориентированы на роль агента, но не на процесс выполнения работ. При взаимодействии агенты используют переговоры. По такому принципу построены следующие системы ADEPT [16], PEACE+ [15].

2. Технология Агента как ключевая технология для формирования гибких workflow систем (деятельностнооснованный подход; реактивные агенты координации задач): Этот подход следует за классическим ориентируемым действием моделированием процесса и workflow методологией управления. Действия, роли, и агенты - отдельные аспекты в пределах workflow схемы.

Управление и поток данных смоделированы явно между действиями. Для координации выполнения задач используется workflow схема. Технология Агента используется, чтобы реализовать гибкую и децентрализованную архитектуру. Реактивные агенты координируют выполнение задач так как это определено в workflow схемах без использования централизованной системы координации. Кроме того, они могут поддерживать сложные образцы потока управления, необходимые для координирования гибких и разнообразных процессов и обрабатывать динамические изменения сценариев документооборота.

Этот подход используется в WFMC [15]

3. Мобильные агенты, перемещение workflow: Мобильная технология агента – отправная точка для реализации концепции перемещения workflows. Мобильные агенты перемещаются в среде и выполняют очередной шаг сценария. Мобильные агенты могут управлять своим перемещением (в динамических сценариях), контролировать выполнение на каждом этапе сценария. Этот подход удобен для обеспечения общих бизнес-процессов между несколькими предприятиями. Недостатком такого подхода является сложность координации взаимодействия связанных потоков.

Этот подход используется в TRP Support Environment (TSE) [17]. Суть этого проекта заключается в использовании мобильных агентов для автоматизации документооборота. Каждый из агентов TSE перемещаясь от одного местоположения в среде к другому выполняет последовательность работ, составляющих сценарий документооборота. Каждый из таких агентов хранит в себе всё необходимую информацию о потоке работ. В разных участках среды агент TSE обладает различными правами. Агенты TSE взаимодействуют на языке взаимодействия агентов TRP Agent Communication Language (TACL), основанном на понятии workflow. Этот проект разрабатывался для связи и расширения существующих workflow систем.

1.7 Вывод

В этом разделе сделано введение в проблему координации вообще и в мультиагентских системах в частности. Описаны подходы координации в мультиагентских системах и их классификация.

В этом разделе так же обосновано применение технологии агентов к автоматизации документооборота, сделана классификация подходов применения этой технологии.

2 МОДЕЛИ И АЛГОРИТМЫ КООРДИНАЦИИ

2.1 Динамические сообщества агентов для выполнения заданий

Как и в [9] в этой работе мы рассматриваем подход (предложенный в [1, 2, 3, 4]) к моделированию разнообразных процессов, протекающих в реальных/виртуальных предприятиях, базирующийся на динамических сообществах интеллектуальных программных агентов, ориентированных на выполнение заданий. Агенты в таком сообществе объединяются во временный консорциум автономных, возможно, географически рассредоточенных функциональных компонент, которые объединяют свои ресурсы и опыт для решения общих задач.

Считается, что задания, которые выполняют агенты, являются набором атомарных работ. Каждый агент способен выполнить некоторую атомарную работу из множества атомарных работ всего сообщества.

Сообщество агентов предназначено для выполнения потоков работ.

2.2 Виды координации в динамическом сообществе агентов

Агенты – участники динамически формируют проблемно-ориентированные сообщества для выполнения заданий получения информации, интеграции, посредничества и обмена, которые возникают в рамках системы на том или ином уровне.

В реальной жизни такие задания распределяются и выполняются группами функциональных компонент. Традиционные модели процессов информационного обмена часто базируются на структурированных наборах жестких отношений между функциональными узлами. При моделировании с использованием большинства сред вышеуказанные задания (потоки работ) представляются как планы – предопределённые ориентированные графы. Эти подходы часто являются статическими и это является их недостатком при использовании в динамических,

быстро изменяющихся системах. Вторым большим недостатком традиционных систем является то, что жесткие модели взаимодействия плохо масштабируются. Поэтому большим преимуществом модели выполнения заданий в динамическом сообществе интеллектуальных программных агентов, предложенной в [1, 2, 3, 4], является отсутствие статических, предопределённых спецификаций заданий.

Как отмечалось в разделе 1 ключевой проблемой взаимодействия является согласованное выполнение работ (координация) агентами, действующими совместно для решения общей проблемы, или выполняющих общее задание.

В [9] была обоснована модель координации динамического сообщества агентов, основанная на использовании координационного агента, суть, которой заключается в следующем.

Взаимодействие агентов рассматриваемого сообщества выглядит как последовательность выполнения работ. Другими словами координация работы сообщества состоит в обеспечении непротиворечивой последовательности выполнения работ.

Обмен результатами выполненных работ происходит через координационного агента - активную доску объявлений. Выполнив работу, агенты отсылают результаты координационному агенту - помещают их на доску объявлений, тем самым, обеспечивая возможность выполнения других работ, параметрами которых являются эти результаты.

Агент, выполняющий работу, параметрами которой являются результаты другой работы, должен запрашивать их у координационного агента - активной доски объявлений.

Этот механизм обеспечивает правильную и непротиворечивую последовательность выполнения работ агентами и обмен промежуточными результатами между ними.

Как отмечалось выше, в динамическом сообществе агенты самостоятельно распределяют атомарные работы между собой. Часто одну и ту же работу в сообществе могут выполнить несколько агентов. Все эти агенты, на момент распределения работ, могут находиться в разных состояниях (например, в различной

степени загружены выполнением работ). Поэтому эффективность распределения работ влияет на результат выполнения общего задания.

Для того чтобы наиболее эффективно распределить работы между агентами они должны взаимодействовать друг с другом – вести переговоры. В процессе переговоров необходимо выявить оптимального исполнителя для каждой работы.

Процесс переговоров при распределении работ – один из видов координации динамического сообщества агентов. Создания математической модели процесса переговоров будут посвящены разделы 2.2, 2.3.

При распределении работ между агентами необходимо также учитывать параметры активности агентов [1, 2, 3]: матрица состояния системы, матрица невыполненных работ, матрица задержек выполнения работ др. Функции мониторинга параметров активности в [4] возложены на координационного агента.

Учитывая всё вышесказанное можно говорить о трёх видах координации в динамическом сообществе агентов:

- координация при распределении работ;
- координация на стадии выполнения работ;
- мониторинг параметров активности агентов.

2.3 Модели координации и алгоритмы

2.3.1 Координация на стадии предварительного взаимодействия (переговоры)

Итак, при распределении задач между агентами в динамическом сообществе агентов используется процесс переговоров.

Дженнингс Н. Р. в [13] приводит следующую структуру процесса переговоров. Любой процесс переговоров состоит из таких компонентов:

- протокола переговоров – набора правил, по которым происходит взаимодействие агентов. Сюда входит количество участников, типы участников, состояния

переговоров, правила, по которым изменяются состояния переговоров, возможные действия участников в каждом состоянии переговоров;

- объекта переговоров – диапазон проблем, по которым должно быть достигнуто соглашение. В общем случае участникам можно динамически изменять объект переговоров;
- модели принятия решения агентом – аппарат принятия решения, который используют участники, чтобы действовать в соответствии с протоколом переговоров.

В следующих подразделах будут описаны протокол переговоров, объект переговоров, и модели принятия решений агентами–участниками переговоров.

2.3.1.1 Объект переговоров

Переговоры в рассматриваемой модели координации динамического сообщества агентов происходят на стадии распределения работ между агентами сообщества. В процессе переговоров агент-инициатор переговоров ищет потенциального агента исполнителя для выполнения некоторой работы.

Учитывая рациональность агентов разумным было бы предположить, что объектом переговоров агентов на стадии распределения работ будет размер вознаграждения, которое получит агент-исполнитель за выполнение работы.

В [18] предложена функция $tdf(t)$, описывающая зависимость размера вознаграждения от времени. Функция $tdf(t)$ задаётся в табличном виде парами чисел (t, tdf) , где t - время выполнения работы, tdf - вознаграждение, которое получит агент-исполнитель, выполнив работу w за время t .

Чаще всего функция $tdf(t)$ будет убывающей. Действительно, чем дольше агент-исполнитель выполняет работу w , тем меньшее вознаграждение он должен получить.

Каждая работа w характеризуется максимальной продолжительностью выполнения d_w , после истечения, которой выполнение работы w становится не актуальным. То есть если агент-исполнитель будет выполнять работу w дольше чем d_w , то его вознаграждение будет равным нулю.

График функции $tdf(t)$ изображен на рисунке 2.1.

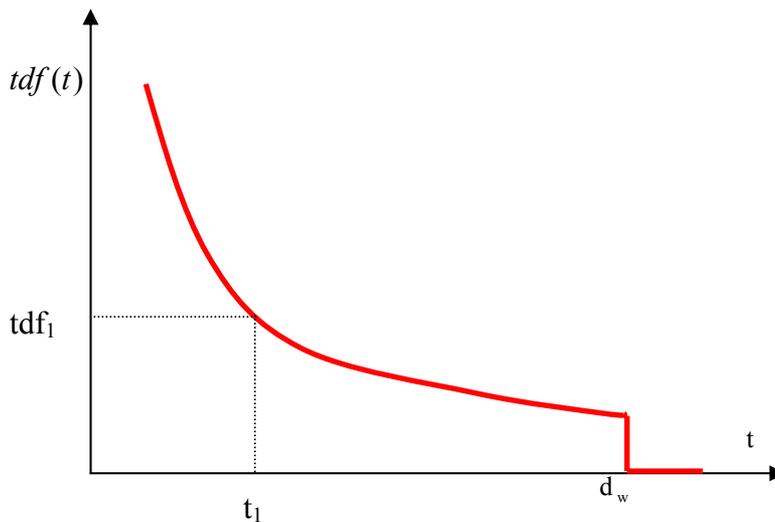


Рисунок 2.1 – График функции $tdf(t)$

Функция $tdf(t)$ не всегда может быть монотонно убывающей как это изображено на рисунке 2.1. Иногда функция $tdf(t)$ может быть, например, периодической. Это говорит о том, что актуальность результата выполнения такой работы тоже изменяется периодически.

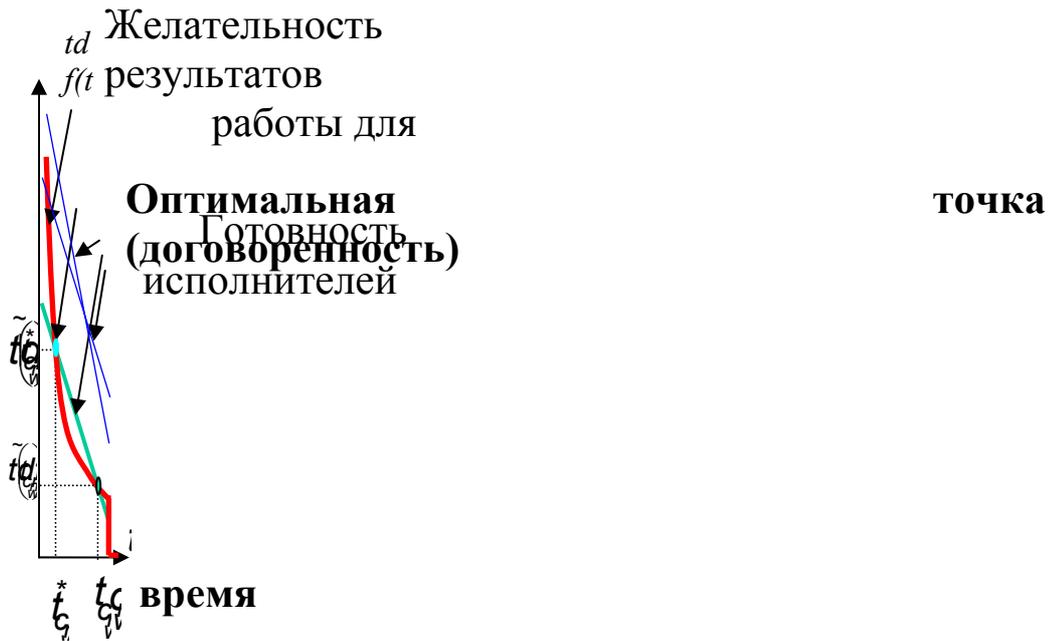


Рисунок
?? -

На рисунке 2.2 изображен процесс соглашения о вознаграждении за выполнение работы. Красной линией обозначена функция $tdf(t)$, описывающая желательность результатов выполнения работы w^n для агента-инициатора переговоров, который ищет исполнителя для её выполнения. Синими и зелёной линиями обозначены функции $tdf(t)$ агентов-исполнителей, описывающие их готовность выполнить работу w^n .

Точки пересечения функций $tdf(t)$ агентов являются точками соглашения. Эти точки описывают условия выполнения работы w^n , на которые согласны и инициатор переговоров и исполнитель. Процедура нахождения оптимальных условий выполнения работы будет дана в разделе 2.3.1.5.

2.3.1.2 Протокол переговоров

Переговоры используются на стадии распределения работ в динамическом сообществе агентов. Целью процесса переговоров является поиск исполнителя для выполнения работы.

В процессе переговоров участвуют агент-инициатор (далее будем называть его агент **I**), желающий поручить выполнение работы w , и агенты-исполнители (каждого из них будем обозначать агент **P**) – потенциальные исполнители работы w .

Итак, агент **I** желает найти исполнителя для работы w . Допустим, работу w могут выполнять несколько агентов – потенциальных агентов-исполнителей. Тогда, с каждым из потенциальных агентов-исполнителей **P** агент **I** должен начать процесс переговоров, протокол которого изображен на рисунке 2.3.

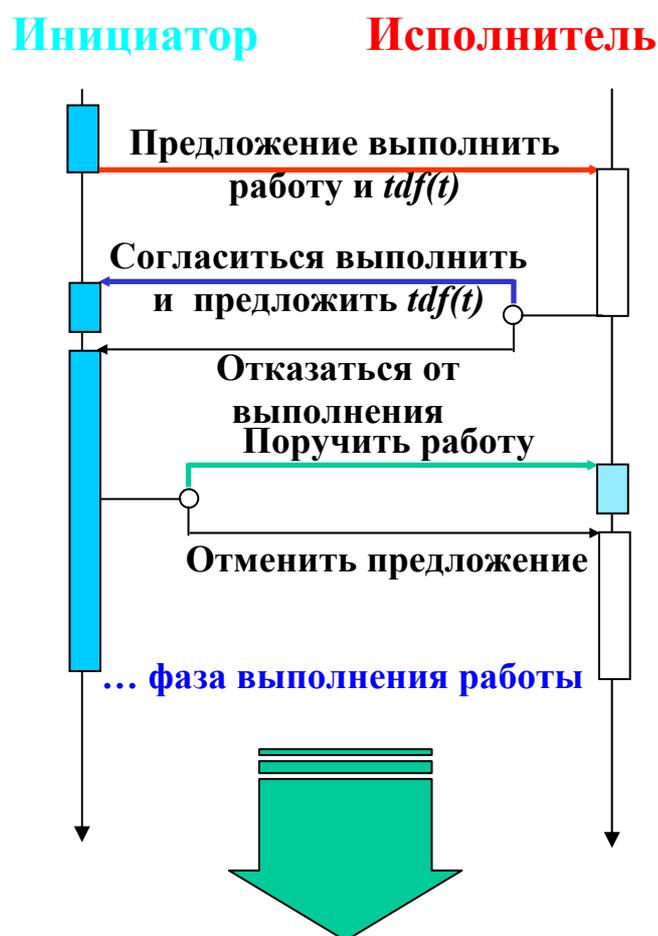


Рисунок 2.3 - Протокол переговоров

Процесс переговоров начинается с того, что агент **I** предлагает выполнить работу w каждому из агентов **P** (эта фаза переговоров обозначена красной линией на рисунке 2.3). Предложение выполнить работу должно состоять из названия работы, параметров её выполнения, и функции $tdf(t)$, описывающей

вознаграждение, которое агент **I** готов заплатить за выполнение работы. Это предложение может быть записано в виде следующего KQML сообщения:

```
(ask-one
  :sender      "I"
  :receiver    "P"
  :in-reply-to Null
  :reply-with  P-TDF
  :language    (XML)
  :ontology    (Trade-off)
  :contents    (См. Рис. 2.4) )

(
  <activity>
    <name>Evaluate_Activity</name>
    <parameter>
      <activity-descr>
        <name>Activity_Name</name>
        <parameter>Activity_Parameter</parameter>
        <result-template>Activity_Result_template</result-
template>
      </activity-descr>
    </parameter>
    <parameter>
      <tradeoff>
        <tdf-value>deswi[1]</tdf-value> <time-point>t[1]</time-point>
        . . .
        <tdf-value>deswi[1]</tdf-value> <time-point>t[1]</time-point>
      </tradeoff>
    </parameter>
  </activity>
)
```

Рисунок 2.4 - Содержимое KQML сообщения предложения работы

Агент **P**, получив предложение выполнить работу, должен на основании своей текущей загруженности, предыдущего опыта выполнения подобных задач, функции $tdf(t)$, предлагаемой агентом **I**, и либо отказаться от выполнения работы, либо согласиться и сформировать свою функцию $tdf(t)$, описывающую вознаграждение, которое агент **P** желает получить за выполнение работы. Эта фаза переговоров обозначена синей линией на рисунке 2.3. Модель, описывающая поведение агента **P** на этой фазе переговоров будет приведена в разделе 2.3.1.4.

KQML сообщения с ответом агента **P** выглядят следующим образом:

```
(tell
  :sender      "P"
  :receiver    "I"
```

```

:in-reply-to    P-TDF
:reply-with     Null
:language       (XML)
:ontology       (TDF-Feedback)

:contents      (  $\tilde{tdf}(t,d)$  )

```

Далее агент **I** должен проанализировать ответы согласившихся выполнить работу агентов **P**, определить среди них агента **P***, кто выполнит работу w с оптимальным сочетанием времени выполнения и размера вознаграждения и поручить ему выполнение этой работы. Эта фаза переговоров обозначена зелёной линией на рисунке 2.3. Модель, описывающая поведение агента **I** при выборе агента-исполнителя будет приведена в разделе 2.3.1.5.

Агенту **P*** отсылается сообщение с поручением выполнить работу w .

Остальные агенты получают информационное сообщение о том, что агент-исполнитель для работы w уже найден.

На этом процесс переговоров закончен и наступает фаза выполнения работы.

Если ни один из агентов **P** не согласился выполнить работу, то агент **I** должен либо пересмотреть свою функцию $tdf(t)$ и начать процесс переговоров с начала, либо отложить выполнение работы w на некоторое время.

Описанный протокол переговоров аналогичен протоколу FIPA CNP, предложенный в [12].

2.3.1.3 Некоторые определения и понятия

Для построения математических моделей поведения агентов, участвующих в процессе переговоров на предварительном этапе координации при распределении работ, необходимо ввести некоторые определения и обозначения.

Пусть $W_c = \{w^1, \dots, w^i, \dots, w^k\}$ - множество работ, которые агент **P** на момент времени t_c ещё не выполнил. Каждую из этих работ агент **P** выполняет в течение одного или нескольких интервалов времени $[t, t + \tau]$. Продолжительность выполнения работы w^i из множества W_c будем измерять количеством интервалов времени $[t, t + \tau]$, в течение которых агент **P** выполняет эту работу, и обозначать

символом l^i . Тогда, если t^i - время начала выполнения работы w^i , то $t^i + l^i\tau$ - время, когда агент **P** полностью выполнит работу w^i .

При выполнении работы w^i из множества W_c на интервале времени $[t, t + \tau]$ агент **P** затрачивает некоторую часть своей мощности. Этот факт будем обозначать $r_{w^i, t}$. Понятно, что величина $r_{w^i, t}$ должна быть неотрицательной ($r_{w^i, t} \geq 0$).

Для каждой работы w^i можно ввести дискретную функцию $R_{w^i}(t)$, заданную на множестве $\{t^i, t^i + \tau, \dots, t^i + l^i\tau\}$ следующим образом:

$$R_{w^i}(t) = \{r_{w^i, t^i}, r_{w^i, t^i + \tau}, \dots, r_{w^i, t^i + l^i\tau}\}.$$

Функция $R_{w^i}(t)$ является функцией распределения загрузки агента **P** работой w^i по времени.

Множество работ W_c характеризуется периодом активности $T_c = [t^s = \min_i t^i, t^e = \max_i (t^i + l^i\tau)]$ агента **P**. T_c - интервал времени, необходимый для выполнения всех работ из множества W_c .

Тогда дискретная функция $R_o(t)$ общей загрузки агента **P**, заданная на множестве T_c определяется следующим образом:

$$R_o(t) = \left\{ \sum_i r_{w^i, t^s}, \dots, \sum_i r_{w^i, t^e} \right\}.$$

В общем случае, общая загрузка агента **P** должна быть ограничена сверху некоторой функцией распределения максимальной загрузки агента $R_m(t)$:

$$R_o(t) \leq R_m(t).$$

Неиспользуемая часть мощности агента, резервируется, например, для погашения пиковых загрузок, срочного выполнения работ и т. п.

На рисунке 2.5 изображен пример распределения загрузки агента по времени, показаны функции $R_{w^i}(t)$, $R_o(t)$ (зелёная линия на диаграмме), $R_m(t)$ (синяя линия на диаграмме). При этом $W_c = \{w^1, w^3, w^3, w^4\}$, $T_c = [t_c - 2\tau, t_c + 2\tau]$.

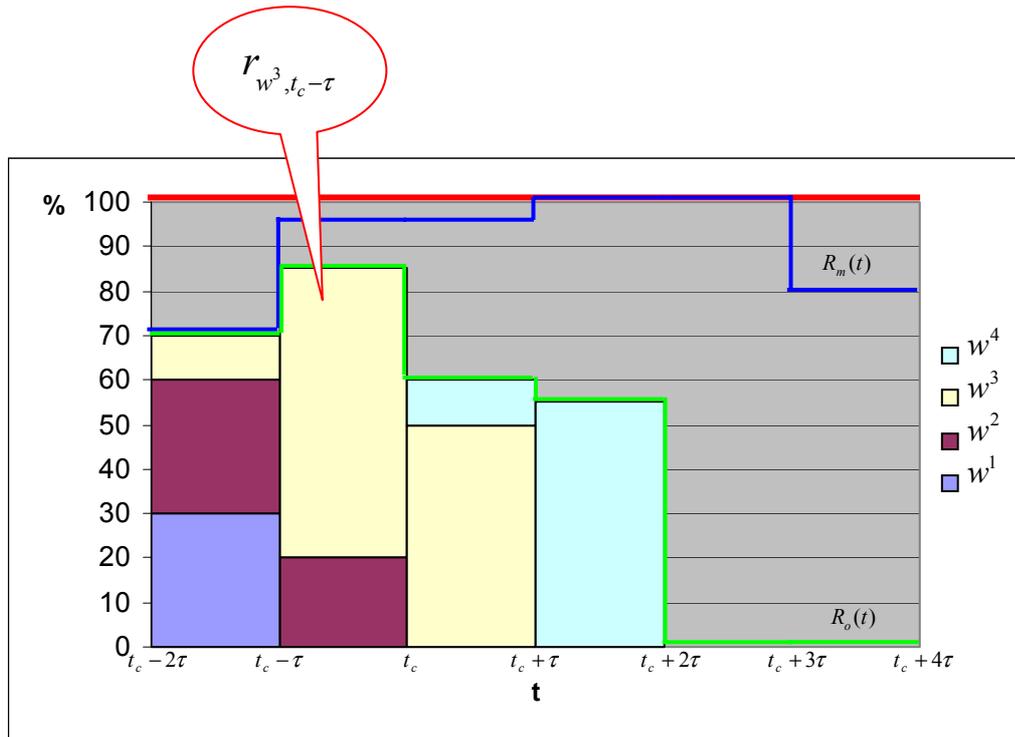


Рисунок 2.5 – Пример функций распределения работ, максимальной и общей загрузки агента

Для того чтобы запланировать выполнения новой работы w^n - получить её функцию распределения по времени, агент P должен на основании своего предыдущего опыта выполнения подобных задач, оценить минимальное время (количество l_n интервалов времени $[t, t + \tau]$), необходимое для выполнения работы w^n при работе в полную мощность.

Введем понятие накопительной мощности для выполнения работы w^i из множества W_c . Накопительной будем называть мощность M_{w^i} , которая была бы затрачена агентом на выполнение работы w^i из W_c в течение одного единичного интервала времени $[t, t + \tau]$. Величина M_{w^i} вычисляется следующим образом

$$M_{w^i} = \sum_{t^i \leq t \leq t^i + \tau} R_{w^i}(t).$$

Тогда для выполнения новой работы w^n необходимо затратить

$$M_{w^n} = 100l_n$$

накопительной мощности агента.

Учитывая функцию распределения общей загрузки агента $R_o(t)$, накопительную мощность M_{w^n} и функцию распределения максимальной загрузки агента $R_m(t)$ можно написать условие для вычисления времени t_{cm} окончания выполнения работы w^n :

$$\sum_{l=0}^{l_{cm}-1} (R_m(t_c + l\tau) - R_c(t_c + l\tau)) \geq M_{w^n}. \tag{1}$$

В качестве значения величины l_{cm} необходимо взять минимальное, удовлетворяющее этому условию.

Теперь можно написать дискретную функцию распределения $R_{w^n}(t)$ работы w^n , заданную на множестве $\{t_c, t_c + \tau, \dots, t_c + l_n \tau\}$:

$$R_{w^n}(t) = \begin{cases} R_m(t) - R_c(t), & t_c \leq t < t_c + l_{cm} \\ M_{w^n} - \sum_{l=0}^{l_{cm}-1} (R_m(t_c + l\tau) - R_c(t_c + l\tau)), & t = t_c + l_{cm} \end{cases}$$

Работа w^n с полученной функцией распределения будет выполнена к моменту времени $t_c + l_{cm} \tau$.

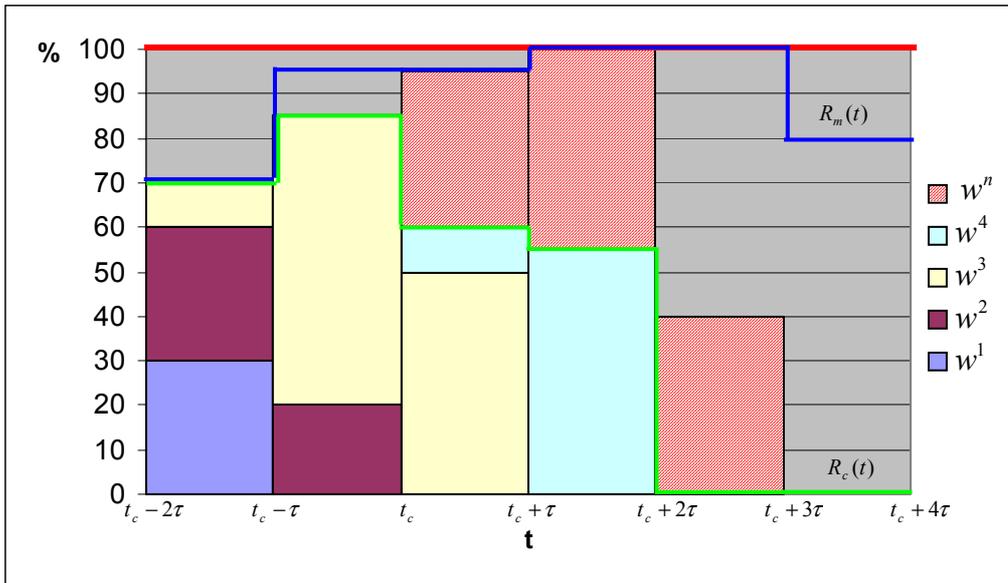


Рисунок 2.6 – Составление функции распределения для новой работы w^n

На рисунке 2.6 приведен пример получения функции распределения для новой работы w^n . Область, занимаемая функцией w^n , на диаграмме заштрихована красным цветом.

2.3.1.4 Модель поведения агента-исполнителя в процессе переговоров

Учитывая, что агент-исполнитель является рациональным агентом, становится очевидным, что агент P будет стремиться максимизировать количество вознаграждения, которое он получит за выполнение работ.

Прибыль вознаграждения за выполнение работы w^n к моменту времени $t_c + l_{cm}\tau$ будет составлять $tdf_{w^n}(t_c + l_{cm}\tau)$.

Пусть $W_n = W_c \cup \{w_n\}$. Мощность этого множества равна $k + 1$.

Тогда ожидаемая прибыль вознаграждения агента P будет равна следующей сумме:

$$\Pi = \sum_{w^i \in W_n} tdf_{w^i}(t_i + l_i\tau).$$

Следовательно. Ожидаемая прибыль вознаграждения зависит только от сроков выполнения работ.

С изменением сроков выполнения работ будет изменяться ожидаемая прибыль вознаграждения – уменьшаться или увеличиваться. Тогда, варьируя сроками выполнения работ можно добиться максимальной прибыли вознаграждения.

То есть те работы, которые принесут большую прибыль вознаграждения, необходимо выполнять как можно раньше (выделив для них больше мощности), а остальные – позже. Изменение величин l_i влечет за собой изменение величин $r_{w^i,t}$ - доли мощности агента, поглощенной работой w^i в момент времени t .

Для нахождения оптимального перераспределения работ из множества W_n необходимо решить следующую задачу:

$$\sum_{w^i \in W_n} t df_{w^i}(t_i + l_i \tau) \rightarrow \max.$$

Это выражение является целевой функцией агента **P** – максимизировать прибыль от выполнения работ.

При изменении функции распределения работы w^i будет изменяться и время завершения работы $t_i + l_i \tau$, которое вычисляется с использованием условия (1).

Для полученного множества W_n вычислим период активности $T_n = [t^s = \min_i t^i, t^e = \max_i (t^i + l^i \tau)]$ агента **P**.

Для каждого интервала времени из T_n введём $k + 1$ переменных $r_{w^i, t}$ - долю мощности агента **P** поглощенную работой w^i в момент времени t . Так как длина интервала T_n равна $t_e - t_s$, то общее количество переменных $r_{w^i, t}$ будет равно $\frac{t_e - t_s}{\tau} (k + 1)$.

Запишем ограничения, накладываемые на значения переменных $r_{w^i, t}$.

Ограничения по значению. Очевидно, что $r_{w^i, t} \geq 0$ и $r_{w^i, t} \leq R_m(t) \quad \forall t$ и $\forall w^i$.

Ограничение по общей накопительной мощности. Сумма всех $r_{w^i, t}$ должна совпадать с суммарной накопительной мощностью всех работ w^i из множества W_n , начиная с текущего времени t_c :

$$\sum_{w^i \in W_n, t_s \leq t \leq t_e} r_{w^i, t} = M_{W_n}.$$

Ограничение по максимальной загрузке. В каждый момент времени $t > t_c$ сумма значений $r_{w^i, t}$ по всем работам не должна превышать максимальной загрузки агента-исполнителя в этот же момент времени. То есть

$$\sum_{w^i \in W_n} r_{w^i, t} \leq R_m(t), \quad \forall t.$$

Ограничение по накопительной мощности для каждой работы. Сумма значений $r_{w^i,t}$ для любой работы w^i должна быть равна M_{w^i} . То есть

$$\sum_{t_s \leq t \leq t_e} R_{w^i}(t) = M_{w^i}, \forall w^i.$$

Таким образом, мы получили задачу нелинейного дискретного программирования с целевой функцией

$$\sum_{w^i \in W_n} tdf_{w^i}(t_i + l_i \tau) \rightarrow \max$$

и ограничениями

$$r_{w^i,t} \geq 0 \text{ и } r_{w^i,t} \leq R_m(t) \quad \forall t \text{ и } \forall w^i$$

$$\sum_{w^i \in W_n, t_s \leq t \leq t_e} r_{w^i,t} = M_{W_n}$$

$$\sum_{w^i \in W_n} r_{w^i,t} \leq R_m(t)$$

$$\sum_{t_s \leq t \leq t_e} R_{w^i}(t) = M_{w^i}$$

Учитывая, что число работ, которые агент-исполнитель одновременно может выполнять невелико, и эти работы атомарные (выполняются они не долго), то период активности T_n агента исполнителя будет небольшим. Кроме того, если предположить, что переменные $r_{w^i,t}$ могут принимать только значения кратные пяти, то полученную задачу можно решать методом полного перебора.

На каждой итерации этого метода необходимо будет вычислять значение целевой функции.

Значения переменных $t_{cm_{w^i}}$ в целевой функции вычисляются на основе уже полученных на этой итерации значений переменных $r_{w^i,t}$ по следующему алгоритму.

1. $t = t_e$.

2. если $r_{w^i,t} \neq 0$ то

$$t_{cm_{w_i}} = t + \tau, \text{ алгоритм закончен}$$

3. $t = t - \tau$, перейти на шаг 2

2.3.1.5 Модель поведения агента-инициатора в процессе переговоров

Агент-инициатор **I** хочет найти исполнителя для некоторой работы w_1 . Для этого, как было сказано в разделе 2.3.1.2, агент **I** должен сгенерировать функцию $tdf_I(t)$, описывающую зависимость максимального размера вознаграждения, которое, агент **I** согласен заплатить за выполнение работы w_1 , от времени выполнения работы.

Далее, в соответствии с протоколом переговоров, агент **I** должен отправить предложение выполнить работу w_1 потенциальным агентам-исполнителям **P**.

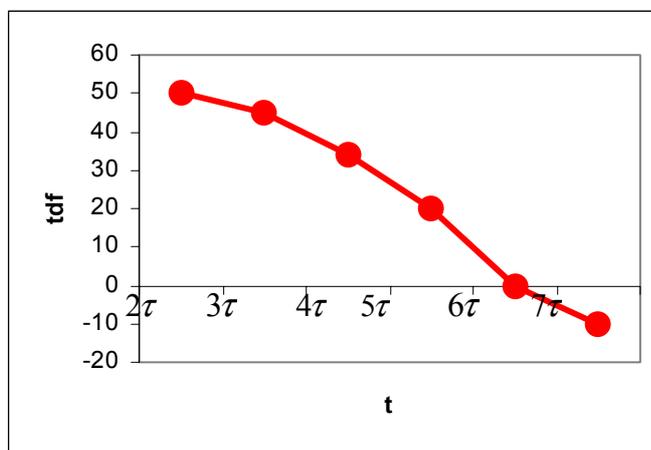


Рис. 2.7 - Пример функции $tdf_I(t)$

Пример функции $tdf_I(t)$ изображен на рисунке 2.7

Это означает, что если исполнитель (агент **P**) выполнит работу w_1 за время 2τ , то агент **I** готов заплатить ему не более чем 50 единиц вознаграждения, если же работа w_1 будет выполнена за время 3τ - то не более чем 45 единиц вознаграждения, и так далее, если работа w_1 будет выполнена за время 7τ , то агент

P должен заплатить штраф в размере 10 единиц (это обозначено отрицательным значением функции).

Каждый из агентов **P**, получив предложение выполнить работу, на основании модели, предложенной в разделе 2.3.1.4 должен либо сгенерировать ответную функцию $tdf_p(t)$, либо отказаться от выполнения работы.

На рисунке 2.8 приведен пример функций $tdf_p(t)$ для четырёх агентов.

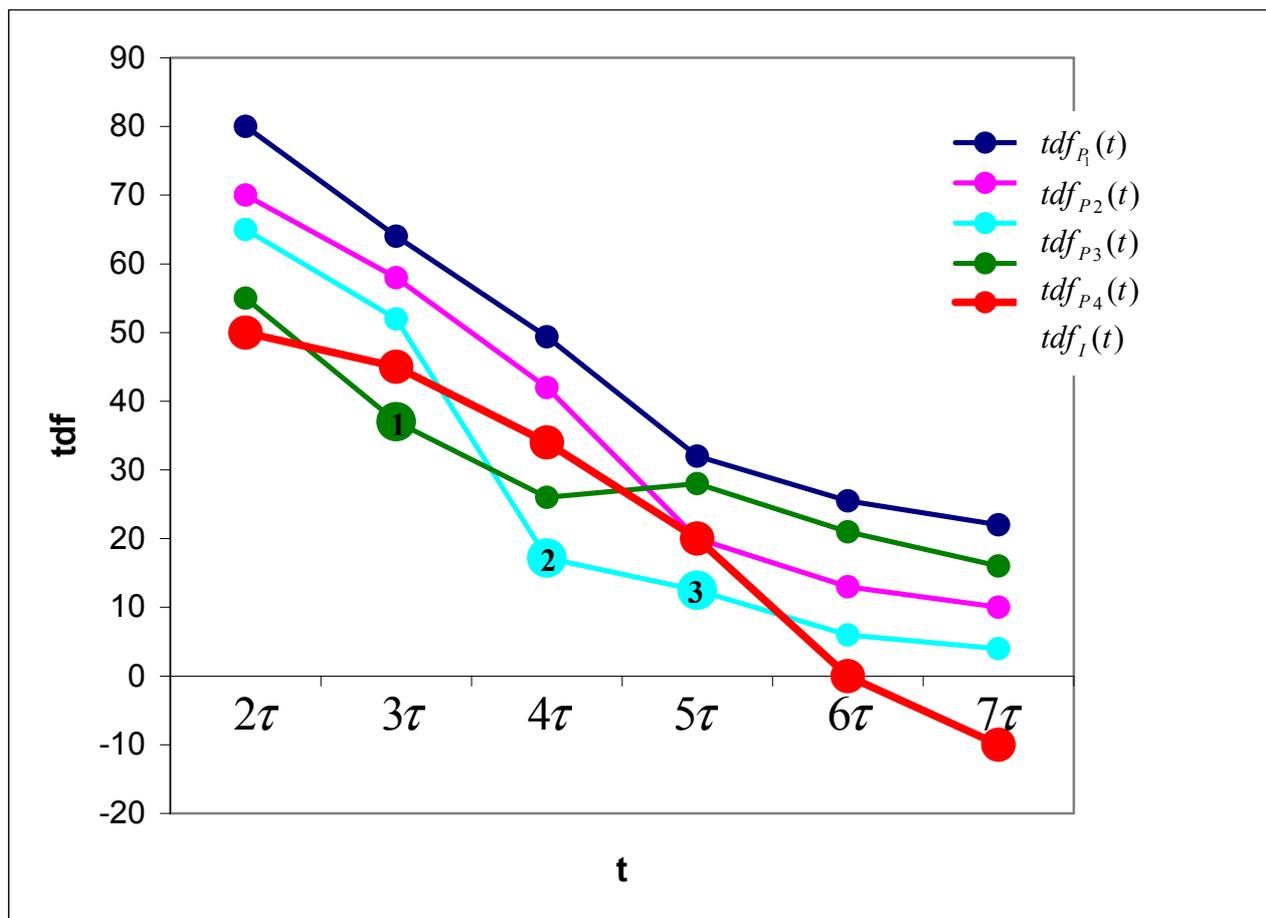


Рисунок 2.8 - Графики функций $tdf(t)$ агентов **P** и агента **I**

Получив от каждого из агентов **P** функцию $tdf_p(t)$, агент **I** должен определить, какому из агентов **P**, и на каких условиях (время выполнения и вознаграждение) поручить выполнение работы w_1 . Другими словами необходимо найти оптимальную точку $(t, tdf_p(t))$ из таблицы функций $tdf_p(t)$.

Очевидно, что ни одна из точек, которые находятся над графиком функции $tdf_I(t)$ (красная линия на рисунках 2.7, 2.8) не является допустимой так как агент **I** не согласен платить вознаграждение, большее, чем он указал в таблице функции

$tdf_I(t)$. На рисунке **X** такими точками являются все точки функции $tdf_{P_1}(t)$, и некоторые точки функций $tdf_{P_2}(t)$, $tdf_{P_3}(t)$, $tdf_{P_4}(t)$.

Множество X точек $(t, tdf(t))$ из таблицы функций $tdf_{P_i}(t)$, которые удовлетворяют условию

$$(t, tdf) \in X \text{ если } ((tdf = \min(tdf_{P_1}(t), tdf_{P_2}(t), \dots, tdf_{P_n}(t))) \wedge (tdf \leq tdf_I(t))) \\ (\forall t \in T)$$

будем называть множеством допустимых точек.

Таким образом, все допустимые точки лежат не выше графика функции $tdf_I(t)$.

Допустимые точки обозначены на рисунке 2.8 цифрами 1, 2, 3 и выделены жирным маркером.

Действительно в точках 1, 2, 3 на рисунке 2.8 агенты **P** готовы выполнить работу w_1 за вознаграждение не превышающее то вознаграждение, которое агент **I** указал в своей таблице функции $tdf_I(t)$.

Множество X допустимых точек удобно записывать в виде следующей таблицы.

Номер точки	Точка (t, tdf)	Агент, которому принадлежит эта точка
1	$(3\tau, 37)$	P_4
2	$(4\tau, 26)$	P_4
3	$(5\tau, 20)$	P_2
4	$(4\tau, 17.2)$	P_3
5	$(5\tau, 12.5)$	P_3

Если же множество допустимых точек окажется пустым, то агент **I** должен либо пересмотреть свою функцию $tdf_I(t)$ и предложить новые условия выполнения

работы, либо отложить выполнение этой работы на более поздний срок, когда агенты **P** будут менее загружены.

Из множества X точек $(t, tdf(t))$ необходимо найти оптимальную, решая двухкритериальную задачу:

$$F(t^*(t, w_i), tdf^*(t, w_i, tdf(t))) \rightarrow \max,$$

где $t^*(t, w_i)$ - функция, описывающая критерий оценки времени выполнения работы w_i , то есть чем ближе время t выполнения работы w_i к оптимальному, тем больше значение функции $t^*(t, w_i)$. Причём, при $t = t_{\text{оптимальное}}$ функция должна достигать максимума.

$tdf^*(t, w_i, tdf(t))$ - функция, описывающая критерий оценки размера премии агента выполнившего работу w_i за время.

Эту задачу будем решать при помощи метода линейной свертки. Предположим, что для агента **I** равнозначимы критерии времени и вознаграждения, поэтому задачу будем решать с параметрами 1, 1.

Таким образом, мы получим следующую целевую функцию двух переменных t и tdf :

$$F(t, tdf) = (t^*(t) + tdf^*(tdf)) \rightarrow \max$$

Итак, нужно найти максимум этой функции на некотором множестве X точек (t, tdf) . Формально решение этой задачи можно записать так:

$$\max_X (t^*(t) + tdf^*(tdf)).$$

2.3.1.6 Алгоритм взаимодействия агентов при распределении работ

Напишем общий алгоритм взаимодействия агентов при распределении работ. Пусть в процессе распределения работы w_1 участвуют агент-инициатор **I** и агенты-исполнители **P**.

1. [Агент-инициатор отправляет предложение выполнить работу агентам-исполнителям]

Агент **I** отправляет всем агентам **P** воздействие, в котором он предлагает выполнить работу w_1 за вознаграждение $tdf_I(t)$:

$$W_{P_i} = \{w = \text{”Оцените возможность выполнить работу”}, X, Y\}$$

где $X = \{w_1, X_{w_1}, Y_{w_1}, tdf_I(t)\}$ - параметры обработки этого воздействия (w_1 - работа которую необходимо выполнить, X_{w_1} - параметры выполнения этой работы, Y_{w_1} - описание ожидаемых результатов для работы w_1 , $tdf_I(t)$ - вознаграждение за выполнение работы)

$$Y = \{tdf_{P_i}(t)\} - \text{описание ожидаемых результатов – функции } tdf_{P_i}(t);$$

индекс i обозначает i -го агента-исполнителя.

2. [Решение задачи нелинейного программирования]

Каждый из агентов **P** оценивает свою загруженность и либо формирует свою функцию $tdf_{P_i}(t)$ либо отказывается от выполнения работы.

Первая точка функции $tdf_{P_i}(t)$ определяется, если функция распределения новой работы будет иметь вид:

$$R_{w^n}(t) = \begin{cases} R_m(t) - R_c(t), t_c \leq t < t_c + l_{cm} \\ M_{w^n} - \sum_{l=0}^{l_c-1} (R_m(t_c + l\tau) - R_c(t_c + l\tau)), t = t_c + l_{cm} \end{cases}.$$

Для нахождения второй точки, через которую будет проходить функция $tdf_{P_i}(t)$, агент **P** должен решить следующую задачу нелинейного программирования:

с целевой функцией:

$$\sum_{w^i \in W_n} tdf_{w^i}(t_i + l_i\tau) \rightarrow \max,$$

и ограничениями:

$$r_{w^i,t} \geq 0 \text{ и } r_{w^i,t} \leq R_m(t) \quad \forall t \text{ и } \forall w^i,$$

$$\sum_{w^i \in W_n, t_s \leq t \leq t_e} r_{w^i,t} = M_{W_n},$$

$$\sum_{w^i \in W_n} r_{w^i,t} \leq R_m(t),$$

$$\sum_{t_s \leq t \leq t_e} R_{w^i}(t) = M_{w^i}.$$

3. [Агент-исполнитель отправляет свою функцию $tdf_P(t)$ агенту-инициатору]

Определив точки, через которые будет проходить функция $tdf_{P_i}(t)$, агент **P** отправляет их агенту инициатору.

4. [Поиск оптимального исполнителя. Решение задачи двухкритериальной оптимизации.]

Получив от каждого агента-исполнителя функции $tdf_{P_i}(t)$, агент-инициатор должен определить оптимальные условия выполнения работы w_1 .

Если множество допустимых точек оказалось пустым, то агент **I** должен либо пересмотреть свою функцию $tdf_I(t)$ и предложить новые условия выполнения работы, либо отложить выполнение этой работы на более поздний срок, когда агенты **P** будут менее загружены.

В случае, когда множество допустимых точек не пусто, необходимо решить следующую задачу двухкритериальной оптимизации:

$$F(t^*(t, w_i), tdf^*(t, w_i, tdf(t))) \rightarrow \max.$$

Или для метода линейной свертки с равнозначными критериями:

$$F(t, tdf) = (t^*(t) + tdf^*(tdf)) \rightarrow \max.$$

5. [Поручение выполнить работу]

Агент-инициатор **I** поручает выполнение работы w_1 , тому агенту-исполнителю **P**, которому принадлежит оптимальная точка, найденная на шаге 4.

2.3.2 Координация на стадии выполнения работы

Используя результаты первой главы, диакоптический подход к моделированию сообществ агентов, описанный в [1, 2, 3, 4], построим модель координации агентов, основанную на активной доске объявлений.

Использование активной доски объявлений отличается от классического подхода координации лишь тем, что все операции чтения/записи с общим полем памяти производит только один агент - координационный агент, который, по сути, инкапсулирует в себе саму доску объявлений, и обеспечивает методы работы с ней. Остальные агенты, при необходимости прочитать или записать на доску некоторую информацию, должны взаимодействовать с координационным агентом при помощи ACL/KQML сообщений.

Рассмотрим более подробно модель координации, последовательность действий агентов, взаимодействие агентов сообщества с координационным агентом.

Рассмотрим такую ситуацию. Пусть для выполнения работы w_1 агентом **A** необходимы результаты работы w_2 , которую выполняет агент **B** (см. Рис. 2.9).

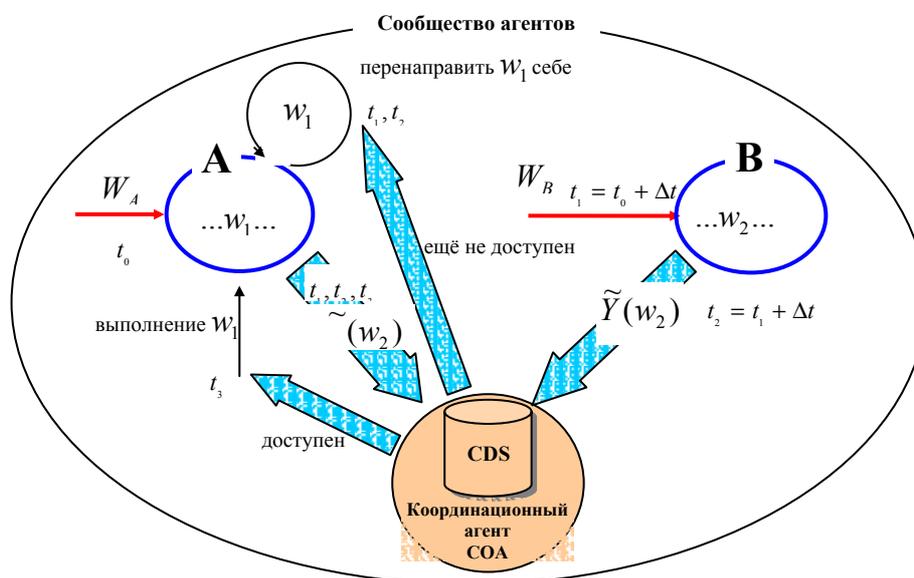


Рисунок 2.9 - Модель координации выполнения работ

Работа w_1 поступает к агенту **A** в момент времени t_1 . Однако она не может быть выполнена за время $]t_1, t_1 + \Delta t]$, так как для её выполнения необходим результат работы w_2 агентом **B**, который нужно запросить у координационного

агента - **COA**. Макромодель $F_O^A(w_1)$ выполнения работы w_1 агентом **A** может быть представлена следующим алгоритмом:

...

Avail := **Require**("COA" with ("ProvideResults", $\tilde{Y}(w_2)$))

IF .NOT. Avail THEN

Redirect w_1 **TO PM** // w_1 перенаправляется **A** на

//следующий момент времени $t_1 + \Delta t$.

ELSE

Execute(w_1) //выполнение w_1 агентом **A** (см. Рис. 2.9)

END IF

...

В этот же момент времени t_1 к агенту **B** поступает работа w_2 . Агент **B** выполняет работу w_2 в интервале времени $]t_1, t_1 + \Delta t]$ и отправляет результат $\tilde{Y}(w_2)$ координационному агенту. Это может быть описано следующим образом:

...

\tilde{Y} := **Execute**(w_2) //выполнение работы w_2

Invoke("COA" with ("AcceptResults", \tilde{Y}))//отправление

// \tilde{Y} агенту COA

...

Реакция координационного агента **COA** на полученное от агента **B** воздействие типа **AcceptResults** имеет такой вид:

...

Execute("AcceptResults", **B**, \tilde{Y})

Реакция **COA** в ответ на запрос **A** результатов работы W_2 , может быть описана в виде следующего алгоритма:

...

Avail := Execute("ProvideResults", $\tilde{Y}(w_2)$))

IF .NOT. Avail THEN

 Reply("A" with ("AcceptResults", "DUMMY"))

 ELSE

 Reply("A" with ("AcceptResults", $\tilde{Y}(w_2)$))

END IF

...

Таким образом, в момент времени t_2 реакция **COA** будет отвергающей, так как результатов работы W_2 ещё нет, а в момент времени t_3 , запрос агента **A** будет удовлетворён.

Необходимо отметить, что результаты выполненных работ, которые передаются между агентами, должны быть представлены в некотором универсальном формате, «понятном» всем агентам. Таким форматом может быть KIF (Knowledge Interchange Format) [19] - формат обмена знаниями - это формальный язык для обмена знаниями между несовместимыми компьютерными программами, написанными разными программистами, в разное время, на разных языках и т. д.

Возвращаясь к предыдущему примеру, напишем на языке KQML[20] сообщения, обеспечивающие взаимодействие с координационным агентом.

KQML сообщение для отсылки результатов выполненной работы координационному агенту может иметь такой вид:

(ask-one

```

:sender      "B"
:receiver   "COA"
:in-reply-to Null
:reply-with Null
:language   KIF
:ontology   COA ontology
:contents   ("AcceptResults",  $\tilde{Y}(w_2)$ ) )

```

KQML сообщение для запроса результатов работы у координационного агента:

(ask-one

```

:sender      "A"
:receiver   "COA"
:in-reply-to Null
:reply-with  $\tilde{Y}(w_2)$ 
:language   KIF
:ontology   COA ontology
:contents   ("ProvideResults",  $\tilde{Y}(w_2)$ ) )

```

KQML сообщение - возвращение результатов координационным агентом:

(tell

```

:sender      "COA"
:receiver   "A"
:in-reply-to Null
:reply-with Null
:language   KIF
:ontology   COA ontology
:contents   ("AcceptResults",  $\tilde{Y}(w_2)$ ) )

```

Более подробно модель координации при выполнении работ, основанная на использовании координационного агента изложена в [4, 9].

2.3.3 Мониторинг основных параметров взаимодействия агентов

В функции координационного агента входит и мониторинг параметров активности агентов сообщества.

К таким параметрам относятся: матрица состояния системы, матрица невыполненных работ, матрица задержек выполнения работ.

Эти понятия были введены в [1, 2, 3, 4].

Пусть $W = \{w_1, w_2, \dots, w_\sigma\}$ - множество допустимых атомарных работ для сообщества агентов.

Матрица состояний системы $\Omega(t_n + \Delta t)$, в момент времени $t_n + \Delta t$, размерностью $m \times \sigma$, где m - количество агентов, σ - количество атомарных работ в множестве W . Строки матрицы Ω (рисунок 2.10) есть вектор $\Theta_i = \{k_1, k_2, \dots, k_j, \dots, k_\sigma\}$, который отображает состояние агента, где k_j - состояние агента I при выполнении работы w_j . В простейшем случае роль параметра k_j может быть такой:

$k_j = 0$ - агент сейчас выполняет атомарную работу w_j ;

$k_j = l > 0$ - агент сейчас выполняет атомарную работу w_j , и ещё l таких работ ждут в очереди своего выполнения;

$k_j = l > 0$ - агент был способен выполнить работу w_j , но не выполнил l таких работ.

Матрица состояний системы $\Omega(t_n + \Delta t)$ формируется координационным агентом из матриц \mathbf{K}_i (размерностью $m \times \sigma$), которые описывают состояние агента. Матрицы \mathbf{K}_i формируются агентами, для обеспечения входных данных для формулы

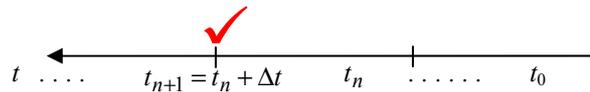


Рисунок 2.10 - Состояние системы при $t_n + \Delta t$

$$\Omega = \sum_{i=1}^n \mathbf{K}_i .$$

Агент может перенаправить одну или более работ w_j для выполнения в следующий момент времени. В этом случае вектор D_a задержек работ над процессом Π_a обновляется таким образом:

$$D_a[j] = D_a[j] + 1.$$

Матрица \mathbf{K}_i строится по правилу изображенному на рисунке 2.11, и отображает поведение агента на интервале времени $]t_n, t_n + \Delta t]$,

где:

$k_{lj}, l \neq i: \mathbf{1}$ – агент i отдаёт работу w_j агенту l ,

$\mathbf{0}$ – в противном случае

$k_{ij}: \mathbf{-1}$ – агент поглощает работу w_j на интервале $]t_n, t_n + \Delta t]$,

$\mathbf{1}$ - агент i перенаправил работу w_j себе,

$\mathbf{0}$ - агент i не может выполнить работу w_j на интервале времени

$]t_n, t_n + \Delta t]$.

$$K_i = \begin{matrix} \text{Агент 1} \\ \text{Агент 2} \\ \dots \\ \text{Агент } i \\ \dots \\ \text{Агент } n \end{matrix} \begin{bmatrix} w_1 & w_2 & \dots & w_\sigma \\ k_{11} & k_{12} & \dots & k_{1\sigma} \\ k_{21} & k_{22} & \dots & k_{2\sigma} \\ \dots & \dots & \dots & \dots \\ k_{i1} & k_{i2} & \dots & k_{i\sigma} \\ \dots & \dots & \dots & \dots \\ k_{m1} & k_{m2} & \dots & k_{m\sigma} \end{bmatrix}$$

Рисунок 2.11 - возможности и намерения агентов на $[t_n, t_n + \Delta t]$.

Анализ значений $\Omega(t_n)$ обеспечивает оценку загрузки агента в момент времени распределение состояний простоя в каждый интервал времени $[t_n, t_n + \Delta t]$.

2.4 Вывод

В этом разделе были рассмотрены три типа координации в динамической коалиции программных агентов, объединяющихся для выполнения задач: координация на стадии распределения работ, координация выполнения работ, мониторинг параметров активности агентов.

Структура процесса переговоров, используемого при распределении работ между участниками коалиции, состоит из объекта переговоров, протокола переговоров, моделей принятия решений агентами в процессе переговоров.

В качестве объекта переговоров предложено использовать вознаграждение. Которое получит агент-исполнитель за выполнение работы. Это вознаграждение описывается функцией $tdf(t)$.

В качестве протокола переговоров используется протокол, аналогичный FIPA CNP протоколу.

Модели принятия решений агентами – участниками переговоров описываются при помощи нелинейной дискретной задачи программирования для агента-исполнителя, и двухкритериальной задачи оптимизации для агента-инициатора процесса переговоров.

Задачей следующего раздела является применение разработанной модели процесса распределения работ на примере модельных задач документооборота ВУЗа.

3 СЦЕНАРИИ ДОКУМЕНТООБОРОТА ВУЗА

3.1 Демонстрация применения модели координации сообщества агентов на сценариях документооборота вуза

3.1.1 Модельная задача координации распределения работ при приеме документов у абитуриента

Предположим, что в процессе принятия документов у абитуриента участвуют такие агенты:

- **А** – агент, представляющий абитуриента;
- **ОС** – агент, представляющий ответственного секретаря приёмной комиссии;
- **ТС₁, ТС₂, ТС₃, ТС₄** – агенты, представляющие технических секретарей приёмной комиссии.

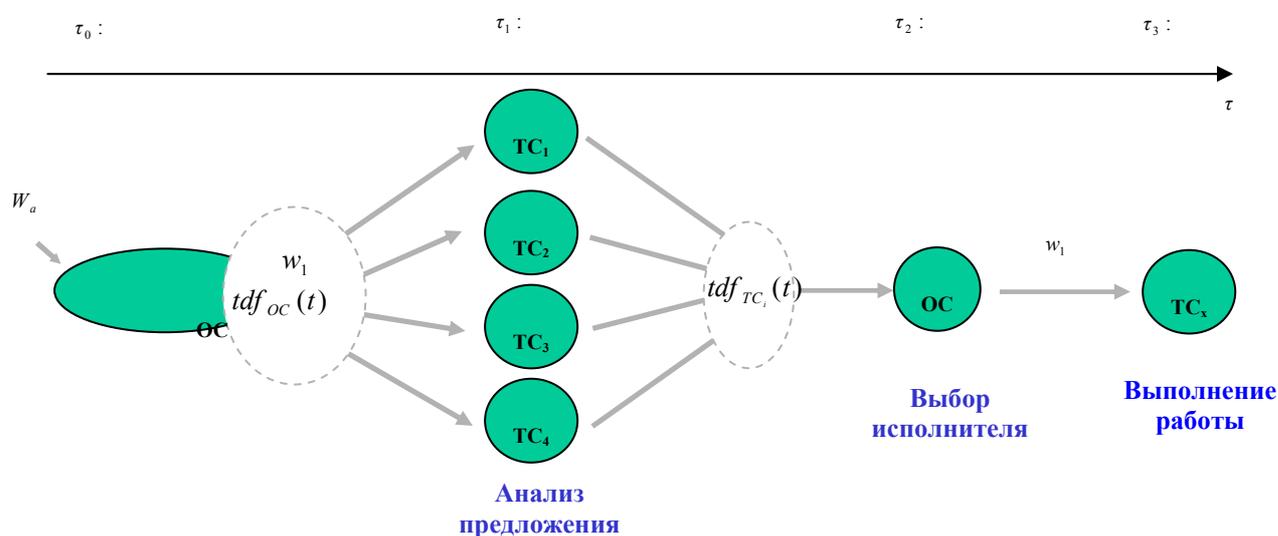


Рисунок 3.1 - Сценарий приёма документов у абитуриента

Пусть в момент времени t_0 абитуриент **А** желает сдать документы. Для этого агент **А** посылает агенту **ОС** воздействие.

Агент **ОС** получив это воздействие должен найти технического секретаря – агента **ТС**, который примет документы у абитуриента – выполнит работу $w_1 = \{\text{Принять_документы}\}$.

Тогда сценарий приёма документов в соответствии с разработанной моделью переговоров будет выглядеть так, как это изображено на рисунке 3.1.

Итак, агент **ОС** должен найти исполнителя для работы w_1 . Работу w_1 могут выполнять агенты **ТС1, ТС2, ТС3, ТС4**. Значит агент **ОС** должен инициировать процесс переговоров для поиска оптимального исполнителя для работы w_1 . Для этого агент **ОС**

3.1.2 Модельная задача координации распределения работ во множительном центре ВУЗа

Рассмотрим в качестве примера сценарий потока работ, возникающий в множительном центре ВУЗа при поступлении заказа на тиражирование документа.

Для демонстрации применения разработанной модели координации при распределении работ достаточно рассмотреть лишь первый этап – поиск агента, который будет выполнять заказ.

Предположим, что в процессе заказа на тиражирование документа участвуют такие агенты:

- **A** – агент, представляющий заказчика;
- **M** – агент, представляющий администратора множительного центра ВУЗа;
- **P1, P2, P3, P4** – агенты, представляющие персонал множительного центра ВУЗа.

Пусть в момент времени t_0 агент **A** делает заказ на тиражирование документа. Для этого агент **A** посылает агенту **M** следующее воздействие.

Агент **M** получив это воздействие должен найти исполнителя – агента **P**, который выполнит тиражирование документа – выполнит работу $w_1 = \{\text{Сделать_копию_документа}\}$.

Тогда сценарий поиска исполнителя для тиражирования документа в соответствии с разработанной моделью переговоров будет выглядеть так, как это изображено на рисунке 3.2.

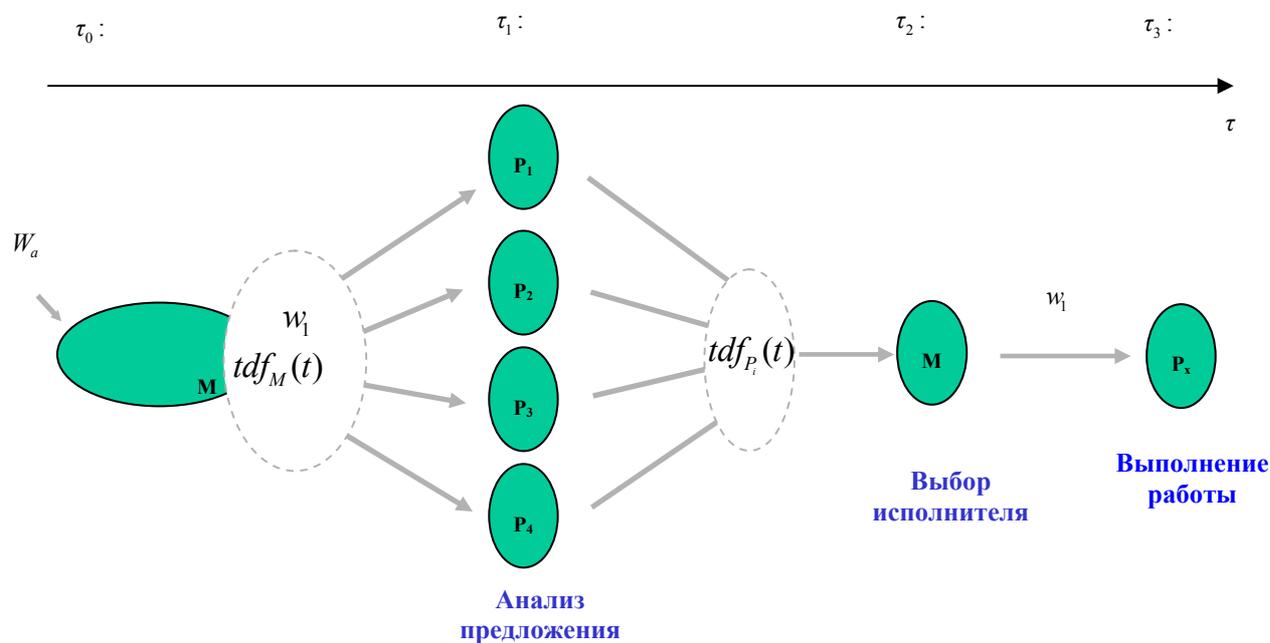


Рисунок 3.2 - Сценарий приёма документов у абитуриента

Итак, агент **М** должен найти исполнителя для работы w_1 . Работу w_1 могут выполнять агенты **P₁**, **P₂**, **P₃**, **P₄**. Значит агент **М** должен инициировать процесс переговоров для поиска оптимального исполнителя для работы w_1 .

ЗАКЛЮЧЕНИЕ

Данная дипломная работа посвящена разработке математических моделей и алгоритмов координации взаимодействующих программных агентов, участвующих в сценариях документооборота ВУЗа. Актуальность работы обусловлена растущей популярностью подходов на базе концепции мультиагентских систем для проектирования и реализации сложных интеллектуальных программных систем. Особенно интересным представляется применение технологии агентов при моделировании документооборота предприятия.

В процессе выполнения дипломной работы были решены следующие задачи и получены следующие результаты:

- сделан сравнительный анализ моделей автоматизации документооборота;
- предложены модели трех видов координации динамического сообщества агентов, участвующих в сценариях документооборота;
- разработана модель поведения агента-исполнителя в процессе переговоров при распределении работ;
- разработана модель поведения агента-инициатора в процессе переговоров при распределении работ;
- разработан алгоритм взаимодействия агентов при распределении работ;
- модели и алгоритмы координации апробированы на двух модельных задачах - сценариях документооборота в ВУЗе

Результаты проведенных в процессе выполнения дипломной работы исследований представляются новыми, актуальными и могут быть использованы при реализации таких сложных программных интеллектуальных систем, как динамические сообщества интеллектуальных программных агентов. Областью применения таких интеллектуальных программных систем является моделирование бизнес процессов и автоматизация деятельности реальных и виртуальных предприятий.

СПИСОК ССЫЛОК

1. S. U. Borue, V. A. Ermolayev, V. A. Tolok: Application of Diakoptical MAS Framework to Planning Process Modelling // in: "Problems of Programming" Scientific Journal №1-2, 2000, ISBN 966-02-1244-5, Special Issue: the Proc. of the 2-nd Intl. Scientific - Practical Conference on Programming (UkrPROG'2000), Kiev, 23-26 May 2000, p. 488-500
2. С. Ю. Борю, В. А. Ермолаев, В. А. Толок: "Диакоптический подход к моделированию процессов в многофункциональных информационных системах". // "Artificial Intelligence" - a theoretical journal №2, 1999, ISSN 1561-5359, spec. issue: Proceedings of International Conference "Knowledge-Dialog-Solution" - KDS'99. Katsiveli, 13-18.1999, pp.211-219
3. V. A. Ermolayev, S. U. Borue, V. A. Tolok, N. G. Keberle: "Use of Diakoptics and Finite Automata for Modelling Virtual Information Space Agent Societies", Вестник ЗГУ №1, 2000
4. В. О. Толок, С. Ю. Борю, В. А. Ермолаев, Н. Г. Кеберле, С. Л. Плаксін, В. В. Михайліченко: Формальні принципи і методи взаємодії моделей функціональних об'єктів єдиного інформаційного простору ВУЗу. Проміжний звіт з д/б проекту № 10/99 Міністерства освіти і науки України, ЗДУ, Запоріжжя, 2000 р. 49 ст.
5. Thomas W. Malone and Kevin Crowston. Toward an Interdisciplinary Theory of Coordination. Center for Coordination Science. Technical Report 120, MIT Sloan School of Management, 1991.
6. Mike Wooldridge and Nick Jennings: "Intelligent Agents: Theory and Practice", Knowledge Engineering Review, v10n2, June 1995.
7. Lesser, V. R. Cooperative Multiagent Systems: A Personal View on the State of the Art // IEEE Trans. On Knowledge and Data Engineering, Vol. 11, No 1, Jan.-Feb. 1999, pp. 133-142.
8. W. Richard Scott. Organizations: Rational., Natural and Open Systems. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1987.

9. С. Л. Плаксин, Реализация среды информационного обмена в динамических мультиагентских системах // Квалификационная работа бакалавра, Запорожский государственный университет, Запорожье, 2000
10. http://www.aaai.org/Pathfinder/html/multi-agent_systems.html
11. Hyacinth Nwana, Lyndon Lee, and Nick Jennings: "Coordination in Software Agent Systems", BT Technology Journal, 14(4), 1996, pp 79-88.
12. "Agent Communication Language", FIPA Spec 2 - 1999
13. N. R. Jennings: Coordination Techniques for Distributed Artificial Intelligence, in Foundations of Distributed Artificial Intelligence(eds. G. M. P. O'Hare and N. R. Jennings), Wiley, 1996, pp. 187-210.
14. R. A. Bourne, C. B. Excelente-Toledo and N. R. Jennings (2000) "Run-Time Selection of Coordination Mechanisms in Multi-Agent Systems" Proc. 14th European Conf. on Artificial Intelligence (ECAI-2000), Berlin, Germany, 348-352.
15. Agent-Oriented Information Systems 2000 Proceedings of the 2nd International Workshop at CAiSE*00, Stockholm, June 2000. Edited by G. Wagner, Y. Lesperance and E. Yu. iCue Publishing, Berlin ISBN 3-8311-0093-4
16. N. R. Jennings, P. Faratin, T. J. Norman, P. O'Brien, B. Odgers and J. L. Alty (2000) "Implementing a Business Process Management System using ADEPT: A Real-World Case Study" Int. Journal of Applied Artificial Intelligence 14 (5) 421-465.
17. Jin W. Chang, Colin T. Scott: Agent - based Workflow:TRP Support Environment (TSE) Computer Networks and ISDN Systems, Volume 28, issues 7–11, p. 1501.
18. •В. А. Ермолаев, С. Л. Плаксин, Координация размещения работ в коалициях агентов, выполняющих задания // «Вісник Запорізького державного університету. Фізико-математичні науки», № 2, 2001, ISBN 966-599-058-4, 6 с.
19. M. R. Genesereth: Knowledge Interchange Format. Specification. - represents the consensus of the X3T2 Ad Hoc Group on KIF as of March 1995.

20.Labrou, Y. and Finin, T. A Proposal for a new KQML Specification. TR CS-97-03, Computer Science and Electrical Engineering Department, University of Maryland Baltimore County, Baltimore, February 1997.