

*На правах рукописи*

**КОВТУНЕНКО Алексей Сергеевич**

**ПРОМЕЖУТОЧНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
РАСПРЕДЕЛЕННОЙ ОБРАБОТКИ ДАННЫХ НА ОСНОВЕ  
АГЕНТНОЙ ТЕХНОЛОГИИ**

**Специальность 05.13.11 –  
Математическое и программное обеспечение вычислительных  
машин, комплексов и компьютерных сетей**

**АВТОРЕФЕРАТ  
диссертации на соискание ученой степени  
кандидата технических наук**

**Уфа – 2013**



## **ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ**

### **Актуальность темы исследования**

Одним из наиболее важных этапов жизненного цикла (ЖЦ) информационно-управляющих систем (ИУС) является проектирование и создание подсистем распределенной обработки данных. От эффективной организации этого этапа зависит качество создаваемой системы и эффективность ее эксплуатации. Наиболее известным программным решением для распределенных программных систем является использование трехзвенных программных архитектур «клиент-сервер» и создание дополнительного слоя программного обеспечения (ПО) – промежуточного слоя (ПС), который позволяет сократить затраты на проектирование и разработку программных модулей обработки, хранения данных, а также их интеграцию.

Однако, в классе программных систем распределенной обработки данных, критичных ко времени (ДКВ) – данных, для которых важное значение имеет время их получения и обработки (телеметрические данные, аудио и видеопотоки, промежуточные данные при распределенном компьютерном моделировании и т. д.), – клиент-серверная архитектура является недостаточно эффективной. Обусловлено это тем, что программно-аппаратные системы распределенной обработки ДКВ имеют высокие требования к скорости обработки и передачи данных, а задача их хранения имеет более низкий приоритет, тогда как клиент-серверная архитектура ориентирована преимущественно на хранение и весьма требовательна к сетевым ресурсам. Таким образом, существующие программные технологии не позволяют обеспечить эффективность ЖЦ ИУС, и одновременно обеспечить соблюдение всех требований, предъявляемых к распределенным программным системам обработки данных: открытости, гибкости, модульности, обеспечения качества предоставляемых услуг, управляемости, безопасности, прозрачности распределения ресурсов.

Более эффективной для использования в данном классе программных систем является архитектура типа «точка-точка», а также возможные гибридные решения. Она позволяет организовать децентрализованную обработку данных и обеспечить гибкое и адаптивное распределение ресурсов вычислительной сети. Наиболее известным подходом к созданию распределенных, децентрализованных систем обработки информации является агентно-ориентированный подход. Он подразумевает использование программных архитектур «точка-точка» в рамках агентной технологии разработки.

В настоящее время подход к проектированию и разработке больших программных комплексов распределенной обработки ДКВ является проблемно-ориентированным, а применимость программных комплексов ограничена предприятием или предметной областью. В связи с этим, задача разработки универсального инструментария для проектирования и построения широкого класса программных систем распределенной обработки ДКВ на основе агентной технологии, обеспечивающей требуемое качество проектируемых приложений является актуальной научной задачей.

### **Степень разработанности темы исследования**

Проблемам стандартизации и создания программных систем универсального назначения, которые бы позволяли проектировать и строить программные комплексы распределенной обработки ДКВ посвящены исследования научных школ как в России так и за рубежом. В частности, для решения задач распределенного компьютерного моделирования и управления разработан стандарт IEEE 1516–2010 (HLA – high-level architecture), определяющий интерфейсы и протоколы взаимодействия между локальными приложениями обработки данных. Ряд спецификаций данного стандарта реализован в виде программных платформ. Многие среды компьютерного моделирования реализуют интерфейсы стандарта, которые позволяет включать их в единую сеть обработки данных.

Наиболее известными в этой области являются работы Р. Фуджимото и др. В России системам распределенной обработки данных применительно к прикладным задачам посвящены работы Е. Б. Замятиной, А. И. Каляева, Г. Г. Куликова, А. И. Микова, В. В. Окольнишникова, С. В. Павлова, Р. Л. Смелянского, Н. И. Юсуповой.

Созданные российскими разработчиками программные комплексы «Мера», «Диана», «Triad.net» и «AGNES» обеспечивают высокую эффективность распределенной обработки ДКВ в частности при решении задач компьютерного моделирования, однако не всегда удовлетворяют одновременно требованиям открытости, гибкости, прозрачности и управляемости создаваемых на их базе вычислительных систем, в большинстве своем являются специализированными и ориентированы на решение узких задач распределенной обработки данных при компьютерном моделировании.

### **Цель исследования**

Целью работы является повышение эффективности жизненного цикла систем распределенной обработки данных, критичных ко времени за счет использования агентного подхода и разработки программного обеспечения промежуточного уровня.

### **Задачи исследования**

1. Разработать обобщенную архитектуру программно-аппаратного комплекса распределенной обработки данных, критичных ко времени с учетом специфики жизненного цикла распределенных программно-аппаратных систем.
2. Разработать язык представления спецификаций программных систем распределенной обработки данных, критичных ко времени.
3. Разработать программную архитектуру систем распределенной обработки ДКВ на основе агентно-ориентированного подхода.
4. Разработать программные средства и программную инфраструктуру для проектирования и построения систем распределенной обработки данных, критичных ко времени.

### **Научная новизна**

1. Обобщенная архитектура программно-аппаратного комплекса распределенной обработки данных, подразумевает прозрачность распределения вычислительных ресурсов и использование метапрограммирования для создания приложений, что позволяет, в отличие от существующих подходов, обеспечить гибкость создаваемых систем, уменьшить время на разработку и реконфигурацию, за счет увеличения степени повторного использования кода а также за счет виртуализации вычислительных ресурсов.

2. Язык представления спецификаций распределенных программных комплексов обработки данных, критичных ко времени, позволяющий, в отличие от существующих высокоуровневых средств разработки, осуществлять структурное проектирование распределенных программных систем.

3. Программная архитектура, реализует агентно-ориентированный подход, что позволяет, в отличие от существующих подходов, учитывать требования живучести и гибкости программных систем распределенной обработки данных, критичных ко времени.

4. Разработанные программные средства и программная инфраструктура относятся к промежуточному слою программного обеспечения, базируются на агентной технологии разработки и представляют собой законченный пакет программ который, в отличие от существующих, для программных комплексов обработки данных, критичных ко времени позволяют обеспечить выполнение требований открытости, гибкости и устойчивости к аппаратным и программным сбоям, а также прозрачности распределения ресурсов.

### **Теоретическая и практическая значимость работы заключается в**

- в разработанной обобщенной архитектуре программно-аппаратных комплексов распределенной обработки данных, которая позволяет снизить время и трудозатраты на создание распределенных программных систем за счет использование метапрограммирования;
- в языке представления спецификаций вычислительных систем, позволяющем осуществлять структурный дизайн программных систем распределенной обработки данных, критичных ко времени, не прибегая к алгоритмическому программированию;
- в программной архитектуре, которая за благодаря агентно-ориентированному подходу позволяет учитывать требования к гибкости и живучести программных систем распределенной обработки данных, критичных ко времени на стадии проектирования;
- в разработанных программных средствах промежуточного слоя, которые, благодаря использованной агентной технологии и языка разработки Java, являются кросс-платформенными и позволяют создавать программные комплексы распределенной обработки данных, критичных ко времени на базе гетерогенных вычислительных сетей с непредсказуемой загрузкой, реализовывать самоорганизующиеся агентные алгоритмы обработки данных, создавать иерар-

хические программные комплексы.

### **Методология и методы исследования**

В работе использовались теоретико-множественные, графовые и агентно-ориентированные методы представления моделей, объектно-ориентированные и агентные методологии проектирования и разработки программных систем, методы системного анализа, математического моделирования сложных динамических систем, а также теории организации параллельных вычислительных процессов.

### **Положения, выносимые на защиту**

1. Обобщенная архитектура программно-аппаратного комплекса распределенной обработки данных.
2. Язык представления спецификаций распределенных систем обработки данных, критичных ко времени.
3. Программная архитектура распределенных систем обработки данных, критичных ко времени.
4. Пакет программ промежуточного слоя для создания систем распределенной обработки данных, критичных ко времени.

### **Степень достоверности и апробация результатов**

Полученные в диссертационной работе результаты не противоречат известным теоретическим положениям и подтверждаются результатами расчетов, апробации и внедрения предложенных в диссертации методов на предприятиях и в учебном процессе Уфимского государственного авиационного технического университета.

Основные научные и практические результаты диссертационной работы докладывались и обсуждались на следующих научных конференциях.

- в 2010 г в рамках «International Workshop on Applied Informatics and Mathematical Methods in Economics», Уфимский государственный авиационный технический университет, г. Уфа
- в 2010, 2012 гг. на Всероссийской молодежной научной конференции «Мавлютовские чтения», Уфимский государственный авиационный технический университет, г. Уфа
- в 2011, 2012, 2013 гг. на Всероссийской зимней школе-семинаре аспирантов и молодых ученых «Актуальные проблемы науки и техники», Уфимский государственный авиационный технический университет, г. Уфа
- в 2011 г. на V Всероссийской научно-технической конференции молодых специалистов, Уфимское моторостроительное производственное объединение, г. Уфа
- в 2012 году на шестой международной конференции «Параллельные вычисления и задачи управления «РАСО'2012», Институт проблем управления

РАН, г. Москва

- в 2013 г. на международной конференции «Information Technologies for Intellegent Decision Making Support», Уфимский государственный авиационный технический университет, г. Уфа

Работа выполнена в рамках плановых исследований кафедры информатики в области построения многоагентных систем моделирования и управления силовыми установками летательных аппаратов, а также разработки эффективных методов компьютерного моделирования сложных организационно-технических систем. Исследования поддержаны грантом № 10-08-00928 «Анализ и синтез многоагентных систем управления газотурбинными двигателями летательных аппаратов»

### **Публикации по теме диссертации**

Основные результаты диссертационной работы докладывались и обсуждались и получили положительную оценку на всероссийских и международных конференциях. Список публикаций по теме диссертации включает 17 научных трудов, в том числе 12 статей (2 в рецензируемых журналах из списка ВАК) и 4 тезисов докладов на научных конференциях и одно свидетельство о государственной регистрации программы для ЭВМ.

### **Структура и объем работы.**

Диссертация состоит из введения, четырёх глав и заключения, изложенных на 126 листах машинописного текста, включая 37 иллюстраций и список используемой литературы из 125 наименований.

### **ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ**

Во **введении** обосновывается актуальность диссертационной работы, формулируются задачи и цели исследования, отмечается научная новизна и практическая ценность полученных результатов.

В **первой главе** выполнен анализ архитектурных особенностей построения распределенных программных комплексов обработки данных, выполнен анализ наиболее распространенных архитектур. Рассмотрены существующие программные средства и технологии построения распределенных программных систем. Обозначен класс прикладных задач распределенной обработки ДКВ: задач управления и компьютерного моделирования сложных технических и организационно-технических систем. Рассмотрены свойства живучести и гибкости распределенных программных систем, как аспекты фактора надежности программных средств, являющиеся наиболее важными для распределенных систем обработки ДКВ. Под живучестью подразумевается способность программной системы сохранять работоспособность в случае отказа ее компонентов. Под гибкостью программных систем понимается возможность наращивания функционала или реконфигурации системы в процессе ее эксплуатации, без остановки, перезапуска и дополнительных затрат.

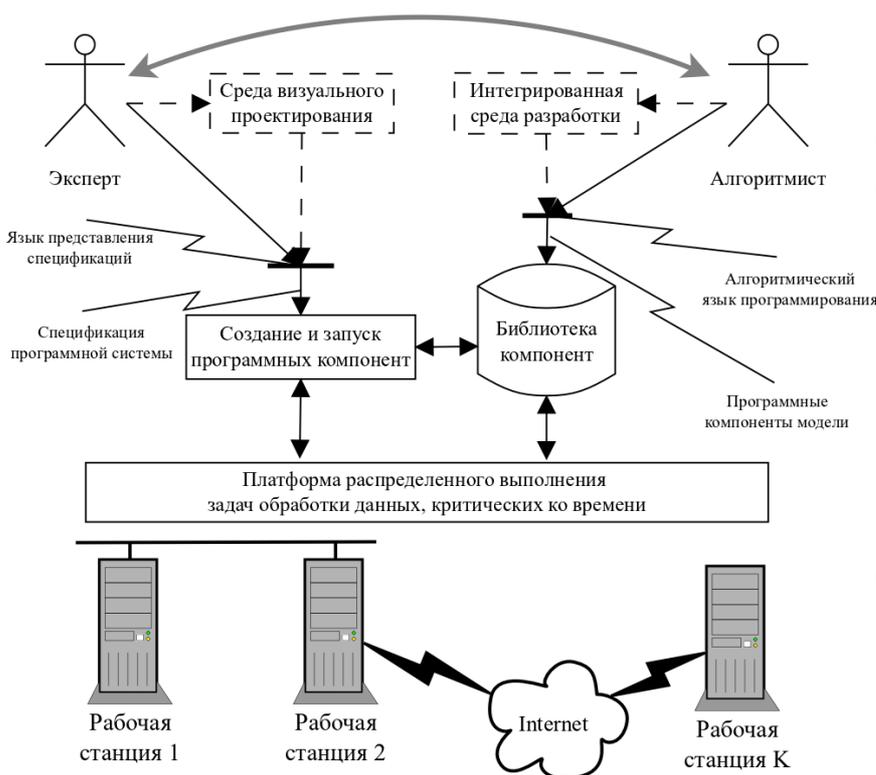


Рисунок 1 – Обобщенная архитектура программно-аппаратных комплексов распределенной обработки данных

ния подзадач обработки данных, которая обеспечивает необходимый уровень виртуализации вычислительных ресурсов сети для различных компонент приложения, передачу данных, прозрачность распределения ресурсов гетерогенной вычислительной сети с непредсказуемой загрузкой;

- библиотека компонент, в которых реализованы типовые подзадачи обработки данных, которые используются при построении приложения;
- компонента создания и запуска подпрограмм обработки данных, которая обрабатывает спецификацию программы и на базе платформы из компонент библиотеки строит распределенное приложение обработки данных;
- интегрированная среда программирования для реализации подзадач обработки данных на алгоритмическом языке и включения их в библиотеку;
- среда визуального проектирования распределенных программных систем, позволяющая создавать спецификацию приложения распределенной обработки данных в виде метапрограммы на специальном языке.

Показано, что агентно-ориентированный подход и агентная технология разработки наилучшим образом подходят для реализации предложенной обобщенной архитектуры.

Во **второй главе** предлагается математическая модель обработки ДКВ. Пусть множество ДКВ задано множеством атрибутов  $V$ , значения которых в процессе функционирования системы необходимо поддерживать актуальными.

Для улучшения показателей качества программных средств на всех этапах жизненного цикла (увеличения надежности, показателей сопровождения, удобства применения и эффективности) предлагается обобщенная архитектура программно-аппаратного комплекса (рисунок 1). В основе архитектуры распределенных систем обработки данных лежит концепция метапрограммирования.

Предлагаемая архитектура характеризуется наличием следующих программных компонент:

- платформа распределенного исполнения

Подзадачи обработки данных задаются множеством операций  $F$  по вычислению текущих значений совокупности исходящих атрибутов  $V_D \subseteq V$  с использованием текущих значений совокупности входящих атрибутов  $V_G \subseteq V$ , при этом  $V_G \cap V_D = \emptyset$ . Каждая операция может быть представлена кортежем  $f = \langle V_G, V_D \rangle$ . Кроме того, задается отображение  $T(f): F \rightarrow \mathbb{R}$ , которое каждой операции ставит в соответствие вещественное число, показывающее период ее повторения в секундах.

Решаемая прикладным программным комплексом задача обработки ДКВ может быть представлена ориентированным биграфом, множество вершин которого  $N = V \cup F$  соответствуют атрибутам и операциям их преобразования, а множество дуг  $C: V \rightarrow F$  – зависимостям между значениями атрибутов или компонентам потоков данных, передаваемым между подзадачами обработки. На основании графового представления задачи была предложена агентно-ориентированная программная архитектура распределенной обработки ДКВ. В основе данной программной архитектуры лежат следующие положения.

1. Задача распределенной обработки ДКВ представляется совокупностью подзадач – операций, каждой из которых соответствует программный агент, называемый *целевым*. Целевой агент содержит следующие элементы:

1) *вектор состояния* – текущие значения совокупности производных атрибутов, значения которых агент должен поддерживать актуальными;

2) *регистр входящих потоков* – текущие значения совокупности производящих атрибутов;

3) *главную процедуру* – периодическую вызываемую процедуру которая соответствует операции обработки данных;

4) периодическую процедуру поддержания актуальности входящих сигналов (обновления регистра входящих потоков).

2. Топология потоков данных между операциями задается, в соответствии с агентно-ориентированным подходом, для каждого целевого агента, посредством указания агентов-источников входящих потоков, а также периода обновления значений в регистре. Это позволяет обеспечить живучесть прикладного программного комплекса, поскольку сбой одного или нескольких целевых агентов не приведет к отказу. В этом случае для расчетов в других операциях будут использоваться устаревшие значения, что повлияет только на точность вычисления других атрибутов. Поддержание потока данных между целевыми агентами осуществляется с применением предложенного прикладного *протокола сигнального взаимодействия*.

3. Создание спецификации вычислительной модели обработки ДКВ и параметров целевых агентов осуществляется на основе метапрограммирования с использованием предложенного в работе языка описания спецификаций (Task Specification Description Language – TSDL). Он основан на стандарте разметки XML. В TSDL-описании распределенной системы обработки данных для каждого целевого агента указываются входящие потоки, размерность состояния, указатель на процедуру обработки из библиотеки компонент (согласной обоб-

щенной архитектуре, предложенной в первой главе), периодичность обновления регистра входящих сигналов и вектора состояния, а также указатель на узел вычислительной сети, на котором выполняются задачи данного целевого агента.

4. Управление распределенным программным комплексом: его запуск, останов и уничтожение производится согласно следующей трехуровневой иерархической структуре с использованием предложенного прикладного *протокола управляющего взаимодействия*.

1) Верхний уровень управления представлен агентом – *главным менеджером*, который осуществляет взаимодействие с пользователем, создание целевых агентов по TSDL-спецификации программной системы и трансляцию управляющих команд на средний уровень.

2) Средний уровень представлен периферийным менеджером, или *менеджером узла*, который осуществляет ретрансляцию управляющих команд целевым агентам вычислительного узла.

3) Нижний уровень представлен механизмами целевого агента, который при поступлении от среднего уровня управляющих команд запускает или приостанавливает выполнение своей главной процедуры, либо самоуничтожается.

5. В условиях локальной нехватки вычислительных ресурсов возможно перераспределение задач между вычислительными узлами сети на основе концепции подвижного кода. Если мощности вычислительного узла не позволяют целевому агенту поддерживать значения переменных актуальными во времени, механизм балансировки нагрузки приостанавливает выполнение главной процедуры и перемещает целевого агента на менее загруженный или более производительный узел. Предложен прикладной *протокол служебного взаимодействия*, который обеспечивает целостность иерархической структуры управления при перемещении целевых агентов между вычислительными узлами. Данный механизм позволяет использовать для исполнения распределенного программного комплекса гетерогенные сети с непредсказуемой загрузкой.

Язык представления спецификаций распределенных программ TSDL основан на нотации XML. Описание его в виде расширенной формы Бэкуса-Наура (РБНФ) приведено ниже.

<Цифра> ::= '0'|'1'|'2'|'3'|'4'|'5'|'6'|'7'|'8'|'9';

<Буква> ::= 'a'|'b'|...|'z'|'A'|'B'|...|'Z';

<Число> ::= <Цифра>, {<Цифра>};

<Имя агента> ::= <Буква>{<Буква>|<Цифра>};

<Название класса> ::= <Буква>, {<Буква>|<Цифра>|'.'};

<Входящий поток> ::= '<slot name='<Имя агента>'>';

<Блок входящих потоков> ::= '<input refreshperiod='<Число>,<'>', {<Входящий поток>}, '</input>';

<Параметр> ::= '<parameter>', <Число>|{<Буква>}</parameter>';

<Агент> ::= '<agent name='<Имя агента>,<' class='<Название класса>,<' refreshperiod='<Число>,<' container='<Число>,<' dimension='<Число>,<'>,<[<Блок входящих потоков>],<[ {<Параметр>} ]</>';

```

<Вычислительная модель> ::= '<model ', ['name=', <Буква>{<Буква>|<Цифра>|'.'_'-'|'(')}], [' author=', <Буква>{<Буква>|'.'_'}], refreshperiod=', <Число>', '>', {<Агент>}, '</model>';

```

В рамках предложенной обобщенной архитектуры предложенной в первой главе, интерпретатор языка TSDL должен входить в состав разрабатываемого пакета программ.

В **третьей главе** рассмотрена реализация предложенной программной архитектуры на языке Java с использованием агентной платформы и библиотеки JADE (Java agent development environment). Она представляет собой законченный программный комплекс – программную платформу. Диаграмма классов программного комплекса частично представлена на рисунке 2. Разработанная программная платформа включает в себя классы, реализующие механизмы и принципы программной архитектуры: протоколы взаимодействия, трехуровневую иерархическую систему управления выполнением подзадач обработки данных, интерпретацию TSDL-спецификаций, развертку платформы на базе гетерогенной вычислительной сети, создание и уничтожение целевых агентов, а также передачу сигналов между целевыми агентами.

На базе класса Agent библиотеки JADE создана система классов-наследников, которая реализует весь функционал программной архитектуры, а также предусматривает возможность его расширения. В основе платформы лежит абстрактный класс CommonAgent. Он реализует протокол управления, предложенный в рамках архитектуры, а также механизм обработки параметров агента. На этом уровне иерархическая система управления распределенной обработкой данных представлена абстрактными процедурами Start(), Stop(), Restart() и Finish(), которые соответствуют основным типам управляющих команд: запуск, останов, перезапуск, уничтожение прикладной программы. Каждая из процедур вызывается в соответствии с протоколом своей командой управления.

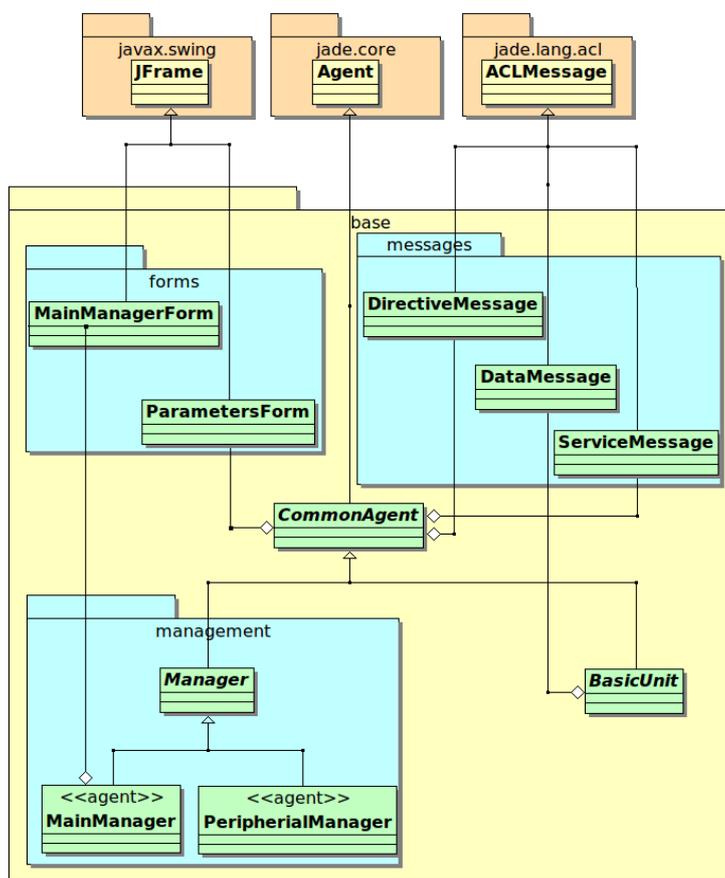


Рисунок 2 – Фрагмент диаграммы классов платформы ABSynth-base

Целевой агент реализован

в виде абстрактного класса звена общего назначения (BasicUnit). В нем реализуется третий уровень иерархической системы управления обработкой данных путем реализации абстрактных процедур запуска, останова, перезапуска и самоуничтожения. Если обработка запущена, с заданной периодичностью производится: поддержание значений в регистре входных значений актуальными во времени, вызов главной процедуры. Первые два уровня управления осуществляется с помощью главного менеджера (класс MainManager) и менеджера узла (класс PeripheralManager).

Создание программного комплекса обработки ДКВ на основании TSDL-спецификаций выполняется главным менеджером. Текстовый файл со спецификацией загружается через стандартный диалог открытия файла с экранной формы главного менеджера. Интерпретатор языка реализован в виде функции класса MainManager, которая на основании синтаксического анализа спецификации создает JADE команды на создание целевых агентов.

При реализации интерпретатора, синтаксис языка TSDL была расширен элементом `<include/>`, который позволяет создавать структурированные описания программ обработки данных, состоящие из нескольких файлов спецификаций. Данное расширение в РБНФ может быть представлено следующим образом.

```
<Имя файла> ::= <Буква> { <Буква> | <Цифра> | '.' | '_' | '/' };
<Включение> ::= ' <include filename=' , <Имя файла> , ' >';
<Вычислительная модель> ::= ' <model ' , [' name=' , <Буква> { <Буква> | <Цифра> | '.' | '_' | '-' | '(' | ')' } ], [' author=' , <Буква> { <Буква> | '.' | '_' } ], refreshperiod=' , <Число> , ' >' , { <Агент> | <Включение> } , ' </model> ';
```

Атрибутом данного элемента является указатель на файл со спецификацией подмодели, которую необходимо создать в составе основной модели. При появлении в составе root-элемента `<model>` элемента `<include/>` обработка основного файла приостанавливается и запускается обработка вложенного файла. Благодаря рекурсивности интерпретатора данный механизм позволяет создавать TSDL-спецификации любой сложности и глубины вложенности. Для обеспечения поименной целостности и непротиворечивости агент главный менеджер в процессе интерпретации TSDL-скрипта пополняет коллекцию заголовков целевых агентов, каждый из которых содержит имя и размерность вектора состояния. Благодаря этому при обработки нескольких вложенных файлов не возникает коллизий.

Гибкость создаваемых на базе предложенной платформы программных систем обеспечивается тем, что для некоторых операций обработки целевые агенты могут быть не созданы сразу. Синтаксис языка позволяет в описании агента не указывать название класса в библиотеке. В этом случае целевой агент не будет создан, однако его имя и размерность состояния будет занесено в коллекцию заголовков. Пока не будет обработан файл спецификации, в котором агент задан более полно поток данных от этой операции будет нулевым. В случае, если производится попытка создать агента с именем уже существующего

агента, старый агент уничтожается. Этот механизм позволяет создавать гибкие и динамически модифицируемые программные системы обработки ДКВ.

Живучесть программных систем обеспечивается агентно-ориентированным подходом и агентной технологией создания программных систем. Каждый целевой агент автономен и не хранит никакой информации о всей задаче обработки.

В четвертой главе рассматривается задача разработки библиотеки типовых компонент компьютерных моделей сложных технических систем. Библиотека представляет собой совокупность классов-наследников класса BasicUnit. В целом библиотека состоит из следующих пакетов.

- library – содержит абстрактные классы, реализующие функционал, общий для некоторой совокупности агентов.
  - Векторно-матричная операция обработки (VectorOperation).
  - Источник модельных сигналов (Source).
- library.units.static – содержит классы агентов, реализующих математические функции.
- library.units.dynamic – содержит классы агентов, реализующие динамические операции. Наиболее важным является следующий класс.
  - Линейное динамическое звено (LinearUnit). Может быть задан несколькими способами:
    - коэффициентами числителя и знаменателя передаточной функции;
    - матрицами коэффициентов уравнений состояния A, B, C, D и начальным состоянием.
- library.units.sources – содержит классы агентов реализующих модельные источники сигналов. В частности, реализованы следующие виды источников модельных сигналов.
  - Текущее модельное время в секундах (Clock).
  - Константа (Constant). Агент содержит экранную форму с элементом управления «ползунок», с помощью которого уровень модельного сигнала (значение константы) можно менять в заданных пределах в процессе выполнения вычислительной модели.
  - Гармонический сигнал (HarmGenerator).
  - Генератор случайных чисел с равномерным распределением (Random).
  - Ступенчатый сигнал (StepUnit). Агент имеет экранную форму с элементом управления «кнопка», которая отражает текущее состояние возвращаемого агентом сигнала, а также позволяет оператору управлять им в процессе выполнения вычислительной модели.
- library.units.sinks – содержит классы агентов, реализующих визуализацию данных. Наиболее общими по функциональности являются следующие агенты.
  - Временная диаграмма (Scope).

- Вывод на экран числовых значений атрибутов (Display).

Библиотека готовых компонент, входящая в состав программного комплекса ABSynth позволяет создавать распределенные компьютерные модели сложных технических систем без использования алгоритмического языка программирования Java, пользуясь исключительно языком описания спецификаций модели TSDL, и, как следствие, не требует от экспертов навыков алгоритмического программирования.

Построение программной системы распределенной обработки ДКВ с использованием разработанной платформы осуществляется в несколько этапов.

1. На основании математической или алгоритмической модели эксперт разрабатывает структурно-функциональную модель. В ней формулируются операции обработки и модель потоков данных между ними.

2. Если библиотека не обеспечивают всего необходимого функционала эксперт формирует алгоритмисту техническое задание на создание новых компонент.

3. На основании это технического задания алгоритмист создает в рамках платформы на языке Java агентные классы и возвращает эксперту их названия и параметры.

4. Эксперт создает текстовый файл со спецификацией модели на языке TSDL в виде xml файла.

5. На базе вычислительной интрасети разворачивается распределенная агентная платформа ABSynth.

6. Оператор с помощью диалогового окна загружает TSDL-файл со спецификацией вычислительной модели. На базе распределенной платформы создаются все целевые агенты системы. Каждый целевой агент при этом попадает на тот вычислительный узел, который для него указан в спецификации.

В качестве примера использования разработанного программного комплекса рассматривалась задача распределенного компьютерного моделирования. На основе известной кусочно-линейная многорежимной динамической модели была разработана компьютерная модель газогенератора ГТД. Была поставлена задача компьютерного моделирования газогенератора с разными параметрами поступающего воздуха на режимах от запуска до режима земного малого газа на базе небольшой корпоративной ЛВС, состоящей из персональных компьютеров (CPU Intel i3, RAM 2048 Mb), объединенных в сеть Ethernet 100M. Узлы сети находились под управлением как ОС Windows XP, так и CentOS 6.3. В результате была создана следующее TSDL-описание модели (рисунок 3).

```

<model name="Engine" author="A. S. Kovtunenko" refreshperiod="1000" >
  <agent name="ostanov" class="library.blocks.dynamic.Trigger" refreshperiod="500" container="1" dimension="1">
  <agent name="vts" class="library.blocks.dynamic.Trigger" refreshperiod="500" container="1" dimension="1">
  <agent name="gazproducer-1" class="examples.GazProducer" refreshperiod="400" container="2" dimension="6">
  <agent name="gazproducer-2" class="examples.GazProducer" refreshperiod="400" container="3" dimension="6">
  <agent name="gazproducer-3" class="examples.GazProducer" refreshperiod="400" container="4" dimension="6">
  <agent name="display" class="library.blocks.sinks.Display" refreshperiod="2000" container="1" dimension="1" >
  <agent name="nnd" class="library.blocks.dynamic.LinearUnit" refreshperiod="500" container="1" dimension="3" >
  <agent name="nvd" class="library.blocks.dynamic.LinearUnit" refreshperiod="500" container="1" dimension="3" >
  <agent name="ntvv" class="library.blocks.dynamic.LinearUnit" refreshperiod="500" container="1" dimension="3" >
  <agent name="scope nnd" class="library.blocks.sinks.Scope" refreshperiod="1000" container="1" dimension="1">
  <agent name="scope nvd" class="library.blocks.sinks.Scope" refreshperiod="1000" container="1" dimension="1">
  <agent name="scope ntvv" class="library.blocks.sinks.Scope" refreshperiod="1000" container="1" dimension="1">
  <agent name="gt" class="library.blocks.sources.Constant" refreshperiod="500" container="1" dimension="1">
</model>

```

Рисунок 3 – TSDL-описание распределенной компьютерной модели газогенератора ГТД

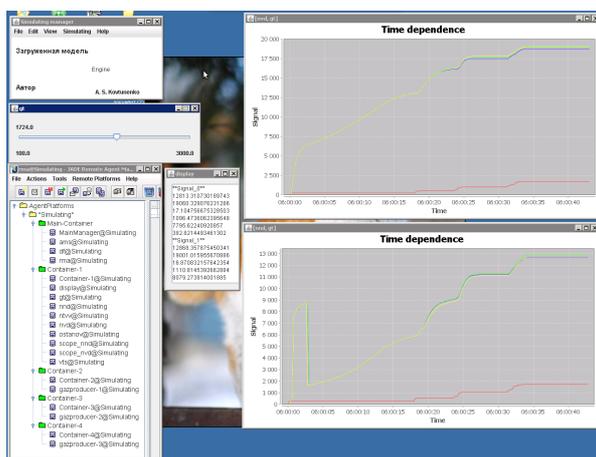


Рисунок 4 – Скриншот экрана главного узла

При этом вычислительные узлы не извлекались из рабочего процесса. Значения переменных модели поступали для отображения на главный узел под управлением CentOS 6.3.

## ОСНОВНЫЕ ВЫВОДЫ И РЕЗУЛЬТАТЫ РАБОТЫ

1. Предложенная обобщенная архитектура программно-аппаратных систем распределенной обработки данных позволяет повысить эффективность организации проектирования и создания ИУС за счет виртуализации вычислительных ресурсов сети и использования метапрограммирования при создании распределенных программных систем обработки данных.

2. Разработанный язык представления спецификаций программных систем распределенной обработки данных, критичных ко времени, благодаря использованной в его основе нотации XML, позволяет экспертам предметной области в интуитивно понятной форме проектировать и создавать распределенные

На рисунке 4 показан скриншот экрана главного узла. Слева расположены формы (сверху вниз): главного менеджера, константы, форма JADE навигатора платформы. Справа расположены формы агентов Scope, отображающие временные диаграммы скоростей вращения (сверху вниз) вала высокого давления, вала низкого давления. При моделировании на трех различных узлах под управлением Windows XP параллельно и асинхронно исполнялись три модели газогенератора с разными параметрами поступающего воздуха (температурой и давлением).

программные системы обработки данных, критичных ко времени.

3. Разработанная программная архитектура распределенных программных систем обработки данных, критичных ко времени, благодаря агентно-ориентированному подходу позволяет обеспечить создаваемым распределенным программным системам обработки данных, критичных ко времени качества гибкости и живучести.

4. Разработанная программная платформа, которая реализует предложенную программную архитектуру, благодаря использованной агентной технологии и кросс-платформенности, позволяет организовать распределенную обработку данных, критичных ко времени на базе гетерогенной компьютерной сети. Разработанная библиотека типовых компонент позволяет осуществлять распределенное компьютерное моделирование сложных технических систем.

#### **Перспективы дальнейшей разработки темы**

В рамках дальнейших исследований планируется наращивание синтаксиса разработанного языка метапрограммирования, создание визуальных средств проектирования и дизайна программных комплексов распределенной обработки данных, разработка многоагентных алгоритмов управления ресурсами вычислительной сети на основе предложенной программной архитектуры, пополнение библиотеки настраиваемыми типовыми компонентами обработки данных, критичных ко времени.

### **ОСНОВНЫЕ ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ**

#### **В рецензируемых журналах списка ВАК**

1. Построение одного класса моделей сложных систем применительно к объектам нефтегазодобычи и нефтепереработки / Ковтуненко, А. С. // Автоматизация, телемеханизация и связь в нефтяной промышленности. – М.: ВНИИО-ЭНГ, 2010. – №7. – С. 25-27.

2. Многоагентная платформа распределенной обработки данных реального времени / С. С. Валеев, А. С. Ковтуненко, В. А. Масленников // Естественные и технические науки. М.: «Спутник+», 2013. – №2. – С. 311-313.

#### **Свидетельства об официальной регистрации программ для ЭВМ**

3. Свидетельство о государственной регистрации программы для ЭВМ № 2013616395. Программная платформа для распределенного имитационного моделирования (ABSynth-base) / А. С. Ковтуненко, С. С. Валеев, В. А. Масленников. – М.: Роспатент, 2013.

#### **В других изданиях**

4. Многоагентная система управления, основанная на библиотеке JADE / С. С. Валеев, А. С. Ковтуненко [опубл. на англ. яз.] // Прикладная информатика и математические методы в экономике: Материалы международного семинара. – Уфа: УГАТУ, 2010. – С. 12-15.

5. Построение компьютерных моделей сложных систем с использованием программных агентов / А. С. Ковтуненко, А. А. Лазарев // Мавлютовские

чтения: Материалы всероссийской молодежной научной конференции. Т. 3. – Уфа: УГАТУ, 2010. – С. 280-281.

6. Идентификация линейной модели динамического объекта повыборочным реализациям входного и выходного сигналов / С. С. Валеев, А. С. Ковтуненко, А. А. Лазарев // Актуальные проблемы науки и техники: Сборник трудов VI Всероссийской зимней школы-семинара аспирантов и молодых ученых. Т. 1. Информационные и инфокоммуникационные технологии, естественные науки. – Уфа: УГАТУ, 2011. – С. 294-298.

7. Многоагентные технологии моделирования и сопровождения сложных ОТС [Электронное издание] / С. С. Валеев, А. С. Ковтуненко, А. А. Лазарев // Транспорт 21 века: исследования, инновации, инфраструктура. Материалы Международной научно-технической конференции. Т. 2. – Екатеринбург: УР-ГУПС, 2011. – С. 710-717.

8. Построение распределенных систем автоматизированного управления на основе многоагентного подхода / А. С. Ковтуненко, А. А. Лазарев // V Всероссийская научно-техническая конференция молодых специалистов: Материалы конференции. Уфимское моторостроительное производственное объединение. – Уфа: УМПО, 2011. – С. 273-275.

9. Создание распределенных систем управления на базе многоагентных технологий реального времени / С. С. Валеев, А. С. Ковтуненко, Р. М. Сайфуллин // Актуальные проблемы науки и техники: Сборник трудов VII Всероссийской зимней школы-семинара аспирантов и молодых ученых. Т. 1. Информационные и инфокоммуникационные технологии, естественные науки. – Уфа: УГАТУ, 2012. – С. 211-214.

10. Проектирование промежуточного программного обеспечения на основе агентно-ориентированных технологий для САУ сложными техническими объектами / С. С. Валеев, А. С. Ковтуненко, В. А. Масленников // Параллельные вычисления и задачи управления: Материалы VI международной конференции. Т. 1. – М.: ИПУ РАН, 2012. – С. 250-259.

11. Построение адаптивной компьютерной модели реального времени / А. С. Ковтуненко, Л. Р. Гарипова // Мавлютовские чтения: Материалы Всероссийской молодежной научной конференции. Т. 3. – Уфа: УГАТУ, 2012. – С. 156-158.

12. Многоагентный подход к построению систем управления сложными техническими системами / С. С. Валеев, А. С. Ковтуненко // Мавлютовские чтения: Материалы Всероссийской молодежной научной конференции. Т. 3. – Уфа: УГАТУ, 2012. – С. 149-150.

13. Проектирование промежуточного программного обеспечения с применением агентно-ориентированного подхода для решения задач управления сложными техническими объектами / С. С. Валеев, А. С. Ковтуненко, В. А. Масленников // Техника и технология: новые перспективы развития. Материалы VII международной научно-практической конференции. – М.: «Спутник+», 2012. – С. 123-126.

14. Разработка промежуточного программного обеспечения для распределенного моделирования на основе агентных технологий / С. С. Валеев, А. С. Ковтуненко // Актуальные проблемы науки и техники: Сборник трудов VIII Всероссийской зимней школы-семинара аспирантов и молодых ученых. Т. 1. Информационные и инфокоммуникационные технологии, естественные науки. – Уфа: УГАТУ, 2013. – С. 186-189.

15. Построение многоагентной реализации клеточно-автоматных компьютерных моделей / А. С. Ковтуненко, Л. Р. Гарипова // Актуальные проблемы науки и техники: Сборник трудов VIII Всероссийской зимней школы-семинара аспирантов и молодых ученых. Т. 1. Информационные и инфокоммуникационные технологии, естественные науки. – Уфа: УГАТУ, 2013. – С. 98-101.

16. Распределенная обработка информации на основе многоагентной технологии / С. С. Валеев, А. С. Ковтуненко // Информационные технологии интеллектуальной поддержки принятия решений: Труды международной конференции. – Уфа: УГАТУ, 2013. – С. 136-137.

17. Распределенное имитационное моделирование больших организационно-технических систем с использованием многоагентной платформы ABSynth / С. С. Валеев, А. С. Ковтуненко, В. А. Масленников // Управление большими системами: Материалы X Всероссийской школы-конференции молодых ученых. Т. 1. – Уфа: УГАТУ, 2013. – С. 231-234.

Диссертант

А. С. Ковтуненко

КОВТУНЕНКО Алексей Сергеевич

ПРОМЕЖУТОЧНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
РАСПРЕДЕЛЕННОЙ ОБРАБОТКИ ДАННЫХ НА ОСНОВЕ  
АГЕНТНОЙ ТЕХНОЛОГИИ

Специальность 05.13.11 –  
Математическое и программное обеспечение вычислительных  
машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ  
диссертации на соискание ученой степени  
кандидата технических наук

Подписано к печати 26.09.2013 г. Формат 60x84 1/16.  
Бумага офсетная. Печать плоская. Гарнитура Times New Roman.  
Усл. Печ. л. 1,0. Усл. Кр-отг. 1,0. Уч.-изд. л. 0,9  
Тираж 100 экз. Заказ № 514