ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

На правах рукописи

РАДЧЕНКО Глеб Игоревич

**СЕРВИСНО ОРИЕНТИРОВАННЫЙ ПОДХОД**

**К ИСПОЛЬЗОВАНИЮ СИСТЕМ**

**ИНЖЕНЕРНОГО ПРОЕКТИРОВАНИЯ И АНАЛИЗА**

**В РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СРЕДАХ**

Специальность 05.13.11 - математическое и программное обеспечение

вычислительных машин, комплексов и компьютерных сетей

Диссертация на соискание ученой степени

кандидата физико-математических наук

Научный руководитель:

СОКОЛИНСКИЙ Леонид Борисович,

доктор физ.-мат. наук, профессор

Челябинск - 2009

2

**ОГЛАВЛЕНИЕ**

Введение ............................................................................................................ 5

Глава 1. Инженерное проектирование в распределенных вычислительных

средах............................................................................................................... 11

1.1. Технологии распределенных вычислений ........................................... 11

1.1.1. CORBA .............................................................................................15

1.1.2. Java RMI............................................................................................17

1.1.3. OGSA ................................................................................................18

1.1.4. P2P-технологии (одноранговые сети) .............................................23

1.2. Системы совместного проектирования ............................................... 27

1.2.1. Распределенная оптимизация инженерных систем .......................27

1.2.2. Потоки задач в инженерном проектировании ................................28

1.2.3. Внедрение инженерных систем в распределенные среды .............29

1.3. Выводы по главе 1 ................................................................................ 31

Глава 2. Технология CAEBeans ...................................................................... 32

2.1. Основные концепции технологии CAEBeans ...................................... 32

2.1.1. Задача инженерного моделирования ..............................................32

2.1.2. Распределенный виртуальный испытательный стенд ...................34

2.2. Архитектура CAEBeans ........................................................................ 36

2.2.1. Концептуальный слой ......................................................................36

2.2.2. Логический слой ..............................................................................41

2.2.3. Физический слой ..............................................................................45

2.2.4. Системный слой ...............................................................................46

2.2.5. Взаимодействие слоев РаВИС.........................................................47

2.3. Программные средства CAEBeans ....................................................... 49

2.3.1. Разработка РаВИС............................................................................49

2.3.2. Исполнение РаВИС ..........................................................................51

2.4. Организация работ в системе CAEBeans ............................................. 52

3

2.4.1. Инженер ............................................................................................53

2.4.2. Прикладной программист ................................................................53

2.4.3. Системный программист .................................................................54

2.5. Параметрические модели производительности Грид ......................... 54

2.5.1. Метрики, зависящие от времени .....................................................55

2.5.2. Метрики, зависящие от объема работы ..........................................58

2.5.3. Адаптация моделей производительности .......................................60

2.5.4. Оценка производительности технологии CAEBeans .....................61

2.6. Выводы по главе 2 ................................................................................ 63

Глава 3. Система CAEBeans ........................................................................... 64

3.1. Структура системы CAEBeans ............................................................. 64

3.1.1. Состав системы CAEBeans ..............................................................64

3.1.2. CAE-проект ......................................................................................64

3.1.3. CAE-параметр ..................................................................................66

3.1.4. Проблемный CAEBean ....................................................................66

3.1.5. Потоковый CAEBean .......................................................................68

3.1.6. Компонентный CAEBean .................................................................74

3.1.7. Интерфейс системного CAEBean ....................................................76

3.1.8. CAE-задание .....................................................................................78

3.2. Конструктор .......................................................................................... 80

3.3. Клиент .................................................................................................... 83

3.4. Сервер .................................................................................................... 84

3.5. CAE-ресурс ............................................................................................ 86

3.6. Брокер .................................................................................................... 88

3.7. Взаимодействие компонентов системы CAEBeans ............................. 92

3.8. Выводы по главе 3 ................................................................................ 95

Глава 4. Испытания системы CAEBeans ....................................................... 96

4.1. Испытание системы CAEBeans на базе DEFORM .............................. 96

4.2. Испытание системы CAEBeans на базе ANSYS Mechanical .............100

4

4.3. Испытание системы CAEBeans на базе Abaqus .................................101

4.4. Испытание системы CAEBeans на базе ANSYS CFX ........................103

4.5. Выводы по главе 4 ...............................................................................104

Заключение .....................................................................................................105

Литература .....................................................................................................110

5

**ВВЕДЕНИЕ**

**АКТУАЛЬНОСТЬ ТЕМЫ**

Системы компьютерного проектирования (CAE - Computer Aided Engineering),

ориентированные на разработку сложных технологических про-

цессов, конструкций, и материалов, являются сегодня одним из ключевых

факторов обеспечения конкурентоспособности любого высокотехнологиче-

ского производства. Использование таких систем дает возможность прово-

дить *виртуальные* эксперименты, которые в реальности выполнить затруд-

нительно или невозможно. Это позволяет значительно повысить точность

анализа вариантов проектных решений и в десятки раз сократить путь от

генерации идеи до ее воплощения в реальном промышленном

производстве [102].

Точность результатов компьютерного моделирования во многом за-

висит от степени детализации сеток, используемых для проведения вычис-

лительных экспериментов. На сегодняшний день размер сеток, используе-

мых в задачах инженерного анализа, может составлять десятки миллионов

элементов [1]. В связи с этим постоянно возрастает вычислительная слож-

ность задач, и требуются значительные вычислительные ресурсы для вы-

полнения инженерного моделирования. Решение этой проблемы заключа-

ется в использовании многопроцессорных систем. Практически все совре-

менные CAE-пакеты имеют параллельные реализации для многопроцес-

сорных систем, в том числе и для систем с кластерной архитектурой.

На сегодняшний день процесс решения задач инженерного проекти-

рования с использованием суперкомпьютерных ресурсов для рядового

пользователя сопряжен с определенными трудностями. С одной стороны,

от него требуется наличие специфических знаний, умений и навыков в об-

ласти высокопроизводительных вычислений, таких как: архитектура супер-

компьютеров, навыки работы в Unix-подобных операционных системах,

6

настройка и администрирование удаленного доступа, умение работать с

очередями приложений и др. С другой стороны, современные системы ин-

женерного проектирования представляют собой многофункциональные

программные комплексы, состоящие из множества отдельных программ-

ных подсистем со сложным пользовательским интерфейсом [32, 36]. Для

решения задач инженерного проектирования пользователю требуется изу-

чить интерфейс и особенности работы всех программных компонентов,

входящих в технологический цикл решения задачи (формирование геомет-

рии задачи, генерация вычислительной сетки, определение граничных ус-

ловий, проведение компьютерного моделирования, визуализация и анализ

результатов решения). Проблема сопряжения компонент существенно ус-

ложняется при использовании одновременно двух и более различных ин-

женерных пакетов для решения одной задачи. Все эти факторы затрудняют

широкое внедрение систем компьютерного инженерного проектирования в

практику НИОКР.

Еще одним важным фактором, препятствующим быстрому внедре-

нию систем инженерного проектирования на промышленных предприяти-

ях, является высокая стоимость приобретения, владения и поддержки су-

перкомпьютерных систем. Для создания суперкомпьютерного центра по

инженерному проектированию необходимо:

1) подготовить помещение и инфраструктуру для суперкомпьютерной

системы;

2) подготовить персонал для поддержки и администрирования супер-

компьютерной системы;

3) приобрести суперкомпьютер, на базе которого будет производиться

моделирование и анализ продукции;

4) приобрести лицензии на пакеты инженерного проектирования;

5) обучить пользователей работе с суперкомпьютером и инженерными

пакетами.

7

Каждый этап данного процесса требует значительных материальных

и людских ресурсов. По этой причине руководители предприятий часто

ставят под сомнение целесообразность внедрения систем инженерного про-

ектирования в заводских лабораториях.

Рациональной альтернативой созданию собственного суперкомпью-

терного центра является аренда вычислительных и программных ресурсов в

режиме удаленного доступа у центров коллективного пользования, функ-

ционирующих при крупных университетах, академических институтах и

других организациях. Однако при этом возникает целый комплекс проблем,

связанных с обеспечением безопасности вычислительных систем и данных.

Указанный комплекс проблем можно решить посредством примене-

ния концепции грид вычислений (Grid Computing) [60] и родственной ей

концепции облачных вычислений (Cloud Computing) [69, 123] в соответст-

вии с которыми, пользователю предоставляется конечный проблемно-

ориентированный сервис, обеспечивающий решение задач на базе ресурсов

распределенных вычислительных систем.

В соответствии с этим *актуальной* является задача разработки сер-

висно ориентированных методов использования систем инженерного про-

ектирования и анализа в распределенных вычислительных средах. В на-

стоящее время эффективные комплексные решения в этой области отсутст-

вуют.

**ЦЕЛЬ И ЗАДАЧИ ИССЛЕДОВАНИЯ**

*Цель* данной работы: на основе концепции облачных вычислений

разработать методы и алгоритмы, обеспечивающие автоматизированную

генерацию проблемно-ориентированных грид-сервисов, позволяющих ис-

пользовать программные системы для инженерного проектирования и ана-

лиза в распределенных вычислительных средах. Для достижения этой цели

необходимо было решить следующие задачи:

8

1) разработать модель проблемно-ориентированного сервиса для реше-

ния задач инженерного проектирования и анализа в грид в виде

РаВИС (Распределенного Виртуального Испытательного Стенда);

2) разработать архитектуру и принципы структурной организации Ра-

ВИС;

3) разработать методы и алгоритмы автоматизированного построения

РаВИС и реализовать их в виде *программной системы CAEBeans*,

обеспечивающей создание и использование РаВИС;

4) провести испытания системы CAEBeans путем создания и внедрения

РаВИС на промышленном предприятии.

**МЕТОДЫ ИССЛЕДОВАНИЯ**

В исследованиях, проводимых в диссертационной работе, использу-

ются методы объектно-ориентированного программирования. Для проекти-

рования систем и алгоритмов применяется аппарат UML.

**НАУЧНАЯ НОВИЗНА**

Научная новизна работы заключается в следующем:

1) предложен комплексный подход к интеграции ресурсов современных

систем инженерного проектирования и анализа в распределенные вы-

числительные среды, обеспечивающий высокую степень автоматиза-

ции разработки и исполнения распределенных виртуальных испыта-

тельных стендов на базе концепции облачных вычислений;

2) разработана модель проблемно-ориентированного сервиса для реше-

ния задач инженерного проектирования и анализа;

3) предложена концепция распределенного виртуального испытательно-

го стенда, обеспечивающая прозрачность предоставления конечному

9

пользователю ресурсов инженерных систем на базе распределенных

вычислительных сред;

4) разработана программная система CAEBeans для автоматизированно-

го создания и исполнения распределенных виртуальных испытатель-

ных стендов.

**ТЕОРЕТИЧЕСКАЯ И ПРАКТИЧЕСКАЯ ЦЕННОСТЬ**

*Теоретическая ценность* работы состоит в том, что в ней предложена

концептуальная модель распределенного виртуального испытательного

стенда, и на ее основе предложен комплекс методов и алгоритмов для пред-

ставления ресурсов систем инженерного проектирования и анализа в виде

грид-сервисов. *Практическая ценность* работы заключается в том, что про-

граммный комплекс CAEBeans, представленный в данной работе, может

быть использован для автоматизированного создания и исполнения распре-

деленных виртуальных испытательных стендов, обеспечивающих решение

различных классов задач инженерного моделирования посредством вычис-

лительных ресурсов, предоставляемых вычислительными грид-сетями.

**СТРУКТУРА И ОБЪЕМ РАБОТЫ**

Диссертация состоит из введения, четырех глав, заключения и биб-

лиографии. Объем диссертации составляет 124 страницы, объем библио-

графии – 125 наименований.

**СОДЕРЖАНИЕ РАБОТЫ**

**Первая глава, «Инженерное проектирование в распределенных**

**вычислительных средах»**, посвящена описанию и исследованию совре-

менных подходов к формированию распределенных вычислительных сред.

Исследованы основные особенности распределенных вычислительных се-

тей, построенных на базе технологий распределенных объектов; одноран-

говых вычислений; грид; облачных вычислений. Производится обзор тех-

10

нологий предоставления ресурсов в распределенных вычислительных сре-

дах и выполняется анализ их применимости к задачам инженерного проек-

тирования в качестве базовой технологии.

**Во второй главе, «Технология CAEBeans»**, производится описание

технологии CAEBeans, представляющей собой комплекс моделей, методов

и алгоритмов, направленных на автоматизированное создание иерархий

распределенных проблемно-ориентированных оболочек (Beans) над инже-

нерными (CAE) пакетами на основе сервисно ориентированного подхода и

концепции облачных вычислений. Раскрывается понятия задачи инженер-

ного моделирования и распределенного виртуального испытательного

стенда.

**В третьей главе, «Система CAEBeans»**, рассматривается архитекту-

ра программного комплекса обеспечивающего разработку и исполнение

распределенных виртуальных испытательных стендов. Приводится описа-

ние базовых сущностей системы CAEBeans. Рассматривается архитектура

предложенной программной системы и основные модули: CAEBeans

Toolbox, CAEBeans Portal, CAEBeans Server, CAEBeans Broker, CAE-ресурс.

**В четвертой главе, «Испытания системы CAEBeans»,** приводится

описание испытательных задач, на основе которых производились исследо-

вания возможности применения системы CAEBeans для решения реальных

задач инженерного моделирования на базе различных базовых инженерных

пакетов.

**В заключении** суммируются основные результаты диссертационной

работы, выносимые на защиту, приводятся данные о публикациях и апро-

бациях автора по теме диссертации, и рассматриваются направления даль-

нейших исследований в данной области.

11

**ГЛАВА 1. ИНЖЕНЕРНОЕ ПРОЕКТИРОВАНИЕ В**

**РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СРЕДАХ**

На сегодняшний день, существует большое число технологий, обес-

печивающих разработку и поддержку распределенных вычислительных

систем. Каждая такая технология ориентирована на некоторый класс задач

и обладает определенными достоинствами и недостатками. В этой главе да-

ется обзор технологий предоставления ресурсов в распределенных вычис-

лительных средах и выполняется анализ их применимости к задачам инже-

нерного проектирования в качестве базовой технологии.

***1.1. Технологии распределенных вычислений***

Проблема выбора между централизованной и децентрализованной

моделями предоставления вычислительных ресурсов является одной из

центральных проблем организации вычислительных систем [47, 81]. До се-

редины 70-х годов прошлого века по причине высокой стоимости телеком-

муникационного оборудования и относительно слабой мощности вычисли-

тельных систем доминировала централизованная модель [95, 30]. В конце

70-х годов появление систем разделения времени и удаленных терминалов,

явилось предпосылкой возникновения клиент-серверной архитектуры,

обеспечивающей предоставление ресурсов мейнфреймов конечным поль-

зователям посредством удаленного соединения [81]. Дальнейшее развитие

телекоммуникационных систем и появление персональных компьютеров

дало толчок развитию клиент-серверной парадигме обработки данных [97].

Согласно парадигме *клиент-серверной архитектуры* один или не-

сколько клиентов и один или несколько серверов совместно с базовой опе-

рационной системой и средой взаимодействия образуют единую систему,

обеспечивающую распределенные вычисления, анализ и представление

данных [109]. Использование клиент-серверного подхода позволило поль-

12

зователю персонального компьютера получить доступ к различным ресур-

сам удаленных серверов, таких как базы данных, файлы, принтеры, процес-

сорное время и др. [40]. Однако использование клиент-серверного подхода

при разработке распределенных приложений было связано со значитель-

ными трудностями. Как правило, клиент-серверные системы основывались

на закрытых протоколах информационного обмена, реализуемых посредст-

вом низкоуровневых сетевых протоколов (TCP/IP, NetBIOS), что затрудня-

ло разработку таких систем и организацию взаимодействия между систе-

мами различных производителей [109]. Появление новых сервисов требо-

вало обновление программного кода клиента и реализацию новых меха-

низмов и протоколов для их использования [44]. Особые проблемы возни-

кали при необходимости объединения приложений в гетерогенных вычис-

лительных средах, состоящих из систем, реализованных на базе различных

аппаратных и программных платформ [71].

Развитие клиент-серверной архитектуры в начале 1990-х привело к

формированию *объектно-ориентированной концепции* распределенных

систем, ориентированной на инкапсуляцию механизма распределенных

взаимодействий и уменьшение сложности разработки распределенных при-

ложений посредством методов объектно-ориентированной разработки и

удаленных вызовов методов объектов [38]. Наиболее известными предста-

вителями данного направления являются технологии CORBA и Java RMI.

Основными достоинствами данного подхода стали [39, 44]:

 простота разработки распределенных приложений по сравнению с

классическим клиент/серверным подходом;

 возможность разработки приложений для гетерогенных вычисли-

тельных сред, обеспечиваемая виртуальной машиной Java и незави-

симым описанием интерфейсов взаимодействующих компонентов;

 возможность отделения интерфейса удаленного объекта от его непо-

средственной реализации.

13

Однако использование объектного подхода было ориентировано на

создание корпоративных систем, работающих в масштабе отдельной орга-

низации или предприятия [120]. Технические ограничения значительно за-

трудняли использование этих технологий для разработки гетерогенных

распределенных вычислительных сред глобального масштаба. Развитие

технологий распределенных вычислений в конце 1990-х годов привело к

разработке концепции грид, которая позволила объединить географически-

распределенные по всему миру гетерогенные ресурсы для решения слож-

ных задач.

Архитектура *грид* ориентирована на стандартизованное совместное

использование автономных географически-распределенных компьютерных

ресурсов в зависимости от их доступности, производительности, цены и

иных характеристик, важных для конечного пользователя [33]. Само поня-

тие грид сформировалось на сопоставлении вычислительного грид (grid – с

англ. сеть) с сетями электроснабжения [112]. На основе этого сопоставле-

ния были выведены следующие три основных требования, которым долж-

ны удовлетворять грид-системы [60].

1. *Гетерогенность*. Вычислительная среда грид состоит из множества

различных ресурсов, обладающих различными характеристиками и

параметрами.

2. *Масштабируемость*. Грид может состоять из произвольного числа

ресурсов. Требуется учитывать, что при увеличении количества за-

действованных ресурсов, значительно возрастают накладные расходы

на передачу данных между компьютерами.

3. *Адаптируемость*. При работе с грид-системами необходимо учиты-

вать, что ошибки при работе с ресурсами – это не исключение, а пра-

вило. Среда Грид может состоять из сотен компьютеров, и ошибки в

работе десятка из них не должны влиять на возможность получения

результатов.

14

Грид-системы ориентированы на формирование виртуального про-

странства для прозрачного совместного использования распределенных ре-

сурсов в рамках виртуальных организаций [122]. *Виртуальная организация*

– это ряд людей и/или организаций, объединенных общими правилами кол-

лективного доступа к определенным вычислительным ресурсам [61].

В 2000-2005 гг. произошло смещение тренда разработки распреде-

ленных приложений с объектно-ориентированного подхода на *сервисно*

*ориентированную парадигму*, обеспечивающую организацию и использо-

вание распределенных сервисов, которые могут принадлежать различным

владельцам [106]. Согласно [98] *сервисами* называются открытые, самооп-

ределяющиеся программные компоненты, обеспечивающие прозрачную се-

тевую адресацию и поддерживающие быстрое построение распределенных

приложений. В последствие сервисно ориентированный подход был принят

разработчиками грид-систем [29, 52, 100] и реализован в виде архитектуры

OGSA (Open Grid Service Architecture) [62].

Параллельно с этим, в начале 2000 г., с появлением систем Napster и

gnutella, получила развитие концепция *P2P-сетей* (от англ. peer-to-peer –

равный-к-равному), обеспечивающая формирование сетей на базе принци-

пов децентрализации. P2P-сети в какой-то степени являются противопо-

ложностью тесно-связанным клиент-серверным сетевым архитектурам [23,

86, 117]. P2P – это разделение вычислительных ресурсов и сервисов по-

средством прямого обмена ресурсами между участниками сети [99]. В от-

личие от традиционной клиент-серверной архитектуры, в P2P-сетях каж-

дый узел, входящий в вычислительную сеть, может являться как клиентом,

так и сервером, предоставляя или используя ресурсы сети. Применение та-

кого подхода позволило обойти основные ограничения клиент-серверных

технологий, связанные со значительным возрастанием нагрузок на выде-

ленные сервера при увеличении количества клиентов и значительно повы-

15

сить отказоустойчивость распределенной вычислительной среды за счет

отсутствия «узких мест» в виде централизованных серверов [58].

В 2008 г. сформировалась новая концепция предоставления вычисли-

тельных ресурсов, названная «*облачными вычислениями*». До сих пор ве-

дутся дискуссии о точном определении данной концепции и ее отличий от

сервисно ориентированной архитектуры и грид [34, 69, 93, 118]. В [119] об-

лаком называется пул виртуальных ресурсов (таких как аппаратное обеспе-

чение, платформы разработки или сервисы), для которых обеспечены лег-

кий доступ и простота использования. Этот пул может быть динамически

реконфигурирован для поддержки меняющейся нагрузки, обеспечивая оп-

тимальное использование ресурсов. Тарификация предоставляемых ресур-

сов производится на базе модели «оплата по факту потребления». Провай-

дер инфраструктуры гарантирует качество сервиса посредством соглаше-

ний об уровне предоставляемых услуг. Основным отличием облачных вы-

числений от концепции грид называют ориентированность грид на совме-

стное использование ресурсов в рамках виртуальной организации, тогда

как облако предоставляет ресурсы произвольному пользователю и ориен-

тировано на коммерческое представление компьютерных ресур-

сов [34, 119].

**1.1.1. CORBA**

*CORBA (Common Object Request Broker Architecture - общая архитек-*

*тура брокера объектных запросов)* [1, 26, 96] – это технология разработки

распределенных приложений, ориентированная на интеграцию распреде-

ленных изолированных систем. Основные компоненты, составляющие ар-

хитектуру CORBA, представлены на рисунке 1.

Технологический стандарт CORBA определяет язык IDL [3, 72], при-

меняемый для унифицированного описания интерфейсов распределенных

объектов, и его отображения на языки Ada, C, C++, Java, Python, COBOL,

16

**Рис. 1.** Ядро архитектуры CORBA.

Lisp, PL/1 и Smalltalk [22]. Для преобразования описания интерфейса на

языке IDL на требуемый язык программирования используется специаль-

ный компилятор [125, 117]. В дальнейшем построенный с его помощью

программный код может быть преобразован любым стандартным компиля-

тором в исполняемый код.

Главной особенностью CORBA является использование компонента

ORB (Object Resource Broker - брокер ресурсов объектов) для создания эк-

земпляров объектов и вызова их методов. Данный компонент формирует

«мост» между приложением и инфраструктурой CORBA [94]. ORB под-

держивает удаленное взаимодействие с другими ORB, а также обеспечива-

ет управление удаленными объектами, включая учет количества ссылок и

времени жизни объекта [105]. Для обеспечения взаимодействия между ORB

используется протокол GIOP (General Inter-ORB Protocol - общий протокол

для коммуникации между ORB) [79]. Наиболее распространенной реализа-

цией данного протокола является протокол IIOP (Internet Inter-ORB Protocol

- протокол взаимодействия ORB в сети интернет), обеспечивающий ото-

бражение сообщений GIOP на стек протоколов TCP/IP [41, 72].

По сравнению с классическим клиент-серверным подходом, исполь-

зование технологии CORBA для разработки распределенных приложений

имеет следующие преимущества:

17

 использование IDL для описания интерфейсов позволяет разрабаты-

вать программные компоненты независимо от языка программирова-

ния и базовой операционной системы;

 поддержка богатой инфраструктуры распределенных объектов;

 прозрачность вызова удаленных объектов.

Однако программные решения на базе технологии CORBA редко вы-

ходят за рамки отдельных предприятий. Разработка крупномасштабных

меж-организационных систем на базе технологии CORBA сопряжена со

следующими трудностями:

 плохая совместимость различных реализаций технологии CORBA от

различных поставщиков [39];

 проблемы взаимодействия узлов CORBA через Интернет [70];

 несогласованность многих архитектурных решений CORBA и отсут-

ствие компонентной модели, которая могла бы значительно упро-

стить разработку [71].

На смену технологии CORBA, пришли стандартизованные протоколы

веб-сервисов, такие как XML, WSDL, SOAP и др. [70]. В настоящее время

CORBA используется для реализации узкого круга приложений и является

фактически нишевой технологией [71].

**1.1.2. Java RMI**

Технология *Java RMI (Remote Method Invocation –у да ле нн ыйв ыз ов м ет од а)* [94] позволяет обеспечить прозрачный доступ к методам удален-

ных объектов, обеспечивая доставку параметров вызываемого метода, со-

общение объекту о необходимости выполнения метода и передачу возвра-

щаемого значения клиенту обратно [68].

Распределенное приложение, разработанное на базе технологии Java

RMI, состоит из двух отдельных программ: клиента и сервера. Серверное

приложение создает удаленный объект, публикует ссылки на него и ожида-

18

ет, когда клиенты произведут вызов метода данного удаленного объекта.

Приложение-клиент получает с сервера ссылку на удаленный объект на

сервере, после чего может вызывать его методы. Технология RMI обеспе-

чивает механизм, при помощи которого производится обмен информацией

между клиентом и сервером [77]. Процесс публикации ссылки на удален-

ный объект может быть реализован с помощью специального регистра или

же посредством передачи удаленной объектной ссылки как части обычной

операции.

Достоинствами использования технологии Java RMI для разработки

распределенного приложения можно назвать возможность разрабатывать

систему целиком основываясь на объектно-ориентированной концепции, не

погружаясь в разработку собственных протоколов взаимодействия между

распределенными компонентами систем, а также кроссплатформенность,

предоставляемую виртуальной машиной Java. К недостаткам данного под-

хода можно отнести:

 строгую ограниченность данной технологии платформой Java;

 необходимость обработки соединений между распределенными ком-

понентами приложения ограничивает масштабируемость используе-

мого подхода.

**1.1.3. OGSA**

Термин «грид» был введен в обращение в начале 1998 года публика-

цией книги «Грид. Новая инфраструктура вычислений» [60]. Хотя в по-

следнее десятилетие базовая идея грид не претерпела существенных изме-

нений, всеобъемлющего определения грид не существует до сих пор [113].

С точки зрения архитектуры приложение грид состоит из множества

различных компонентов, отвечающих за те или иные аспекты функциони-

рования. Например, типовое приложение грид может содержать следующие

сервисы [111]:

19

1) сервис управления виртуальными организациями;

2) сервис поиска и управления ресурсами;

3) сервис управления заданиями.

Ключевым моментом в разработке грид приложений является *с та н-*

*д ар ти за ци я*, позволяющая организовать поиск, использование, размещение

и мониторинг различных компонентов, составляющих единую виртуаль-

ную систему, даже если они предоставляются различными поставщиками

услуг или управляются различными организациями [62].

В качестве базы для создания стандарта архитектуры грид приложе-

ний была выбрана сервисно ориентированная концепция и технология веб-

сервисов. Данный выбор был обусловлен двумя основными достоинствами

данной технологии. Во-первых, язык описания интерфейсов веб-сервисов

WSDL (Web Service Definition Language) поддерживает стандартные меха-

низмы для определения интерфейсов отдельно от их реализации, что в со-

вокупности со специальными механизмами связывания (транспортным

протоколом и форматом кодирования данных) обеспечивает возможность

динамического поиска и компоновки сервисов в гетерогенных средах. Во-

вторых, широко распространенная адаптация механизмов веб-сервисов оз-

начает, что инфраструктура, построенная на базе веб-сервисов, может ис-

пользовать различные утилиты и другие существующие сервисы, такие как

различные процессоры WSDL, системы планирования потоков задач и сре-

ды для размещения веб-сервисов [51].

Разработанный стандарт архитектуры грид получил название *OGSA*

(Open Grid Services Architecture –Открытая Архитектура Грид

Сервисов) [53]. Он основывается на понятии грид-сервиса. *Г ри д-с ер ви со м*

называется сервис, поддерживающий предоставление полной информации

о текущем состоянии (потенциально временного) экземпляра сервиса, а

также поддерживающий возможность надежного и безопасного исполне-

ния, управления временем жизни, рассылки уведомлений об изменении со-

20

стояния экземпляра сервиса, управления политикой доступа к ресурсам,

управления сертификатами доступа и виртуализации [51]. Грид-сервис

поддерживает следующие стандартные интерфейсы.

 *П ои ск. Грид приложениям необходимы механизмы для поиска дос-*

*тупных сервисов и определения их характеристик.*

* Д ин ам ич ес ко е с оз да ни е с ер ви со в. Возможность динамического созда-*

*ния и управления службами –это один из базовых принципов OGSA,*

*требующий наличия сервисов создания новых сервисов.*

* У пр ав ле ни е в ре ме не м ж из ни. Распределенная система должна обес-*

*печивать возможность уничтожения экземпляра грид-сервиса.*

* У ве до мл ен ие. Для обеспечения работы грид приложения наборы грид*

*сервисов должны иметь возможность асинхронно уведомлять друг*

*друга о изменениях в их состоянии.*

*Первая реализация модели OGSA, разработанная в 2003 г., называ-*

*лась OGSI (Open Grid Service Infrastructure). В связи с тем, что существо-*

*вавшие тогда стандарты веб-сервисов (к которым относились WSDL,*

*SOAP, UDDI) не могли обеспечить всех требований, предъявляемых разра-*

*ботчиками к функциональным возможностям грид-сервисов, при создании*

*OGSI потребовалось модифицировать и расширить соответствующие стан-*

*дарты [55]. Это привело к тому, что совместное использование веб-*

*сервисов и грид-сервисов в одной среде стало невозможным, из-за несо-*

*вместимости базовых стандартов [19].*

*Дальнейшие совместные усилия сообщества грид и организаций по*

*разработке стандартов веб-сервисов привело к определению стандартов,*

*соответствующих требованиям грид. В предложенном стандарте WSRF*

*(Web Service Resource Framework) [43, 54, 57, 116] специфицированы уни-*

*версальные механизмы для определения, просмотра и управления состоя-*

*нием удаленного ресурса, что является критически-важным с точки зрения*

*21*

***Р ис 2.*** *Общая схема взаимодействия компонентов Globus Toolkit 4.0.*

*грид [63]. На сегодняшний день реализация модели OGSA посредством*

*стандарта WSRF (и сопутствующих стандартов, таких как WS-Notification и*

*WS-Addressing) является наиболее распространенной в среде грид.*

*В настоящее время, существуют две системы, обеспечивающие ин-*

*фраструктуру разработки грид-систем в соответствии со стандартами*

*OGSA, реализованными посредством WSRF: Globus Toolkit [56] и*

*UNICORE [90].*

*П ро ек т Globus начал разрабатываться в конце 1990-х годов для соз-*

*дания и поддержки сервисно ориентированной инфраструктуры для грид-*

*вычислений [59]. В составе системы Globus Toolkit 4.0 [56], первой системы*

*грид-вычислений, обеспечившей полноценную поддержку стандарта*

*WSRF, можно выделить следующие функциональные группы (см. рис. 2):*

* базовые сервисы (GRAM –управление ресурсами; File Transfer,*

*GridFTP - передача файлов; Trigger, Index –поиск и каталог ресурсов*

*и др.);*

* контейнеры для пользовательских сервисов, поддерживающие аутен-*

*тификацию, управление состоянием, поиск и т.п. обеспечивающие*

*поддержку стандартов WSRF, WS-Security, WS-Notification;*

*22*

* библиотеки, для обеспечения взаимодействия сторонних приложений*

*с GTK 4.0 и/или пользовательскими сервисами.*

*П ро ек т UNICORE (Uniform Interface to Computing Resources –еди-*

*ный интерфейс к вычислительным ресурсам) зародился в 1997 году, и к на-*

*стоящему моменту представляет собой комплексное решение, ориентиро-*

*ванное на обеспечение прозрачного безопасно5го доступа к ресурсам*

*грид [114].*

*Архитектура UNICORE 6 [115] формируется из клиентского, сервис-*

*ного и системного слоев (см. рис. 3). Верхним слоем в архитектуре является*

*клиентский слой. В нем располагаются различные клиенты, обеспечиваю-*

*щие взаимодействие пользователей с грид средой:*

* UCC (Unicore Command Line Client –клиент командной строки для*

*UNICORE): клиент, обеспечивающй интерфейс командной строки*

*для постановки задач и получения результатов;*

* URC (Unicore Rich Client –многофункциональный клиент*

*UNICORE): клиент, основанный на базе интерфейса среды Eclipce,*

*предоставляет в графическом виде полный набор всех функциональ-*

*ных возможностей системы UNICORE;*

* HiLA (High Level API for Grid Applications –высокоуровневый про-*

*граммный интерфейс для приложений грид): обеспечивает разработ-*

*ку клиентов к системе UNICORE;*

* Порталы: доступ пользователей к грид-ресурсам через интернет, по-*

*средством интеграции UNICORE и систем интернет-порталов.*

*Промежуточный сервисный слой содержит все сервисы и компонен-*

*ты системы UNICORE, основанные на стандартах WSRF и SOAP. Шлюз – это компонент, обеспечивающий доступ к узлу UNICORE посредством ау-*

*тентификации всех входящих сообщений [90]. Компонент XNJS обеспечи-*

*вает управление задачами и исполнение ядра UNICORE 6. Регистр сервисов*

*23*

***Р ис 3.*** *Архитектура UNICORE 6.*

*обеспечивает регистрацию и поиск ресурсов, доступных в грид-среде. Так-*

*же, на уровне сервисного слоя обеспечивается поддержка безопасных со-*

*единений, авторизации и аутентификации пользователей.*

*В основании архитектуры UNICORE лежит системный слой. Интер-*

*фейс целевой системы (TSI –Target System Interface) обеспечивает взаимо-*

*действие между UNICORE и отдельным ресурсом грид-сети. Он обеспечи-*

*вает трансляцию команд, поступающих из грид-среды локальной системе.*

*Основным достоинством использования системы UNICORE 6 для*

*разработки распределенных вычислительных систем можно считать нали-*

*чие богатого арсенала различных клиентов, обеспечивающих взаимодейст-*

*вие пользователя с ресурсами вычислительной сети, а также развитых*

*средств обеспечения безопасности при разработке грид-приложений.*

***1.1.4. P2P-т ех но ло ги и (о дн ор ан го вы е с ет и)***

*Технология P2P (анг. peer-to-peer - равный-к-равному) обеспечивает*

*формирование сетей на базе принципов децентрализации. Одноранговые*

*вычислительные сети, в какой-то степени, являются противоположностью*

*тесно-связанным клиент-серверным архитектурам (таким как CORBA,*

*24*

***Р ис 4.*** *Сравнение P2P и централизованной (клиент-серверной) архитектур.*

*RMI) [117]. P2P –это разделение вычислительных ресурсов и сервисов по-*

*средством прямого взаимодействия между участниками сети [99]. В отли-*

*чие от традиционной клиент-серверной архитектуры в P2P-сетях каждый*

*узел, входящий в вычислительную сеть, может являться как клиентом, так*

*и сервером, предоставляя или используя ресурсы сети. На рис. 4 представ-*

*лены связи в сетях с P2P и с централизованной архитектурой [92].*

*Основными преимуществами одноранговых вычислений являются:*

* высокая масштабируемость, связанная с равномерным распределени-*

*ем вычислительной нагрузки на всех участников сети;*

* стабильность работы сети, обусловленная отсутствием «узкого мес-*

*та» - выделенного сервера, обрабатывающего все сетевые запросы;*

* возможность объединения ресурсов отдельных участников сети, и их*

*предоставление другим участникам;*

* распределение совокупных затрат на предоставление ресурсов между*

*участниками сети.*

*В настоящее время наблюдается значительный скачек в развитии од-*

*норанговых сетей. По статистическим данным на конец 2006 года объем*

*трафика, генерируемого файлообменными сетями на базе P2P-сетей, соста-*

*вил более 70% всего сетевого трафика [49]. Наибольшее распространение*

*одноранговых сетей наблюдается в системах, обрабатывающих большие*

*25*

*объемы данных и обеспечивающих индивидуальный обмен информацией*

*между пользователями:*

* системы хранения и разделения информации (Gnutella [66], eDonkey,*

*BitTorrent [27]);*

* системы распределенных вычислений (проекты на платформе*

*Boinc [28]);*

* системы IP-телефонии (Skype [67,110]);*

* системы трансляции видео (Joost [24], PPLive [86]);*

* системы поддержки совместной работы (Groove [50]).*

*Структура P2P сети определяет принципы поиска новых узлов и за-*

*мены узлов вышедших из состава сети новыми. Можно выделить два ос-*

*новных типа P2P сетей: централизованные и децентрализованные [86].*

*Ц ен тр ал из ов ан на я с тр ук ту раP2P сети подразумевает наличие вы-*

*деленного индексного сервера (трекера) собирающего информацию об уз-*

*лах, входящих в P2P-сеть и обеспечивающего поиск и предоставление не-*

*обходимых сервисов одних узлов другим. Примером централизованной*

*структуры является сеть BitTorrent. Центральным узлом данной сети явля-*

*ется трекер, содержащий информацию о списке узлов, подключенных к се-*

*ти, и сервисах, предоставляемых каждым узлом (список файлов, доступных*

*для загрузки с данного узла). Для получения необходимого файла, узел по-*

*сылает трекеру запрос, содержащий уникальный идентификатор необходи-*

*мого файла. На данный запрос трекер возвращает список узлов, на которых*

*доступен требуемый файл.*

*Д ец ен тр ал из ов ан на я с тр ук ту раP2P сети предполагает отсутствие*

*выделенного сервера. Поиск и предоставление сервисов производится пу-*

*тем процедуры пошагового поиска, в которой могут участвовать все узлы,*

*входящие в сеть. Типичным примером одноранговой сети с децентрализо-*

*ванной структурой является система gnutella. В данной сети, обнаружение и*

*подключение к узлам сети происходит посредством процедуры случайного*

*26*

*обхода. Каждый узел содержит таблицу соседей, содержащую IP адрес и*

*порт известного узла gnutella. При запуске новый узел gnutella переходит в*

*режим начальной загрузки, в котором посредством одного из доступных*

*источников (список узлов на одном из узлов интернет; внутренний предус-*

*тановленный список узлов и др.) формирует начальный список соседей.*

*После чего, соседям высылается сообщение обнаружения, пересылаемое*

*далее по цепочке систем. Таким образом обеспечивается обнаружение ре-*

*сурсов, предоставляемых всеми узлами подключенными к сети.*

*Архитектурные особенности одноранговых сетей в комплексе с за-*

*ложенными в них механизмами функционирования обуславливают опреде-*

*ленные недостатки, присущие таким сетям:*

* в одноранговых сетях не может быть обеспечено гарантированное ка-*

*чество обслуживания: любой узел, предоставляющий те или иные*

*сервисы, может быть отключен от сети в любой момент;*

* индивидуальные технические характеристики узла, могут не позво-*

*лить полностью использовать ресурсы P2P сети;*

* при работе того или иного узла через брандмауэр может быть значи-*

*тельно снижена пропускная способность передачи данных в связи с*

*необходимостью использования специальных механизмов обхода;*

* участниками одноранговых сетей в основном являются индивидуаль-*

*ные пользователи а не организации, в связи с чем возникают вопросы*

*безопасности предоставления ресурсов: владельцы узлов P2P-сети,*

*сокрее всего, не знакомы друг с другом лично, предоставление ресур-*

*сов происходит без предварительной договоренности;*

* при значительном увеличении числа участников P2P сети возникает*

*большая нагрузка на сеть (как с централизованной, так и с децентра-*

*лизованной структурой).*

*Таким образом, одноранговые сети ориентированы на анонимное*

*предоставление ресурсов одних индивидуальных пользователей другим.*

*27*

*Они не ориентированы на безопасное, стандартизованное предоставление*

*вычислительных ресурсов крупных организаций и сообществ.*

***1.2. С ис те мыс ов ме ст но гоп ро ек ти ро ва ни я***

*Интеграция систем инженерного проектирования в распределенные*

*вычислительные среды посредством сервисно ориентированной концепции,*

*в настоящее время является одним из основных направления развития от-*

*расли инженерного анализа [42]. Перспективы применения сервисно ори-*

*ентированной архитектуры для построения инфраструктуры систем инже-*

*нерного проектирования хорошо согласуются с распределенным характе-*

*ром этих систем, и позволяет обеспечить «бесшовное» сопровождение раз-*

*рабатываемого продукта на всех этапах его жизненного цикла, начиная с*

*концепции дизайна, и заканчивая его утилизацией и переработкой [84].*

*В мировом научном сообществе считается перспективным направле-*

*ние, связанное с применением грид-технологий для решения ресурсоемких*

*задач инженерного анализа. В тоже время наблюдается тенденция перехода*

*от разработки специализированных грид-ориентированных систем инже-*

*нерного проектирования к интеграции существующих классических CAE-*

*систем в сервисно ориентированные грид-среды [9]. Существуют специ-*

*альные надстройки для системы UNICORE, позволяющие осуществлять*

*удаленную постановку задачи инженерным и научным системам Gaussian,*

*Nastran, Fluent, Star-CD [82].*

***1.2.1. Р ас пр ед ел ен на я о пт им из ац ияи нж ен ер ны х с ис те м***

*В последнее десятилетие было проведено множество большое число*

*исследований по разработке систем, ориентированных на различные аспек-*

*ты внедрения пакетов инженерного проектирования в распределенные вы-*

*числительные среды. В работах [73, 80, 85, 124] представлены подходы к*

*разработке грид-сервисов, обеспечивающих проведение численной оптими-*

*зации параметров инженерных систем на базе ресурсов, предоставляемых*

*28*

*гетерогенными распределенными вычислительными сетями. Принципы*

*массового параллелизма, заложенные в алгоритмах многокритериальной*

*оптимизации, позволяют достичь высоких показателей эффективности ис-*

*пользования разнородных вычислительных систем. Применение модели*

*грид-сервисов позволило обеспечить взаимодействие между различными*

*целевыми системами, независимо от языков программирования и базовых*

*операционных систем. Однако в этих работах не рассматривается возмож-*

*ность интеграции в грид CAE-пакетов, а также отсутствует комплексный*

*подход по переносу всего технологического цикла решения задачи инже-*

*нерного проектирования и анализа в распределенную вычислительную сре-*

*ду.*

***1.2.2. П от ок и з ад ачви нж ен ер но м п ро ек ти ро ва ни и***

*В настоящее время наибольшее распространение для организации*

*взаимодействия распределенных грид-сервисов получил подход, основан-*

*ный на понятии потока задач (Workflow) [76]. В работе [65] предлагается*

*следующее определения п от ок а з ад ач: автоматизация процессов, заклю-*

*чающаяся в объединении грид-сервисов для решения определенной задачи*

*или для определения нового сервиса. В работе [36] была предложена архи-*

*тектура системы, поддерживающей совместную работу нескольких групп*

*инженеров над задачами инженерного проектирования посредством грид*

*(на примере задач моделирования соударения). Основным протоколом ор-*

*ганизации взаимодействия грид-сервисов в гетерогенной вычислительной*

*среде был выбран стандартный протокол описания потоков задач*

*BPEL4WS (Business Process Execution Language for Web Services) [31]. Ос-*

*новным недостатком данного подхода является жесткая детерминирован-*

*ность потока задач, не позволяющая обеспечить динамическое предостав-*

*ление требуемых вычислительных ресурсов в зависимости от конкретных*

*входных данных.*

*29*

*В связи со специфическим характером потоков задач, возникающих в*

*процессе инженерного проектирования, в работе [78] представлена концеп-*

*ция взаимодействия компонентов комплекса инженерного проектирования,*

*основанная на потоках данных, возникающих между ними в процессе ре-*

*шения задач. Потоки задач в области CAE фактически определяются пото-*

*ками данных. Обеспечивается передача данных от постоянного хранилища*

*на вычислительный ресурс, обрабатывающий эти данные; производится*

*разбиение данных для подготовки к параллельной обработке; производится*

*трансформация данных, когда различные шаги обработки требуют различ-*

*ных форматов данных.*

***1.2.3. В не др ен иеи нж ен ер ны х с ис те м вр ас пр ед ел ен ны е с ре ды*** *В работах [45, 75] описываются системы инженерного проектирова-*

*ния и анализа, построенные на технологии CORBA, обеспечивающие со-*

*вместную распределенную работу пользователей над единым проектом.*

*В качестве основного недостатка такого подхода можно отметить, что тех-*

*нология CORBA не была ориентирована на интеграцию систем инженерно-*

*го проектирования в распределенную вычислительную среду, в связи с чем*

*требуется разработка версий инженерных систем на базе этой технологии.*

*В работах [37, 91] приводится пример разработки систем совместного*

*проектирования на базе технологии удаленного вызова методов Java RMI.*

*Достоинством данного подхода является независимость от платформы ис-*

*полнения и простота организации доступа к таким системам посредством*

*веб-доступа, что вытекает из использования базовой технологии Java. Но*

*недостатком данного подхода является отсутствие стандартной инфра-*

*структуры, которая позволила бы обеспечить унифицированные механизмы*

*интеграции ресурсов в гетерогенные вычислительные среды.*

*В работах [48, 121] раскрываются аспекты разработки системы со-*

*вместного проектирования, ориентированной на решение задач инженерно-*

*30*

*го проектирования в динамических распределенных группах разработчи-*

*ков. В связи с потребностью постоянного взаимодействия между участни-*

*ками команды инженеров в процессе разработки, в качестве базовой плат-*

*формы данной системы была выбрана концепция P2P-вычислений. Такой*

*подход позволяет обеспечить беспрепятственный обмен информацией во*

*время разработки проекта, прозрачно использовать динамические вычисли-*

*тельные ресурсы коллектива разработчиков. Но применение такого подхода*

*ориентировано не на предоставление конечному пользователю готового*

*продукта, предоставляющего ресурсы распределенной сети, а на поддержку*

*процесса совместного проектирования командой разработчиков. Такие сис-*

*темы не ориентированы на предоставление проблемно-ориентированного*

*интерфейса для решения конкретного класса задач инженерного проекти-*

*рования.*

*В работах [87, 88] представлена концепция грид-среды для совмест-*

*ного проектирования (Collaborative Manufactoring Grid) система совместной*

*разработки, основанная на стандартах грид-систем. Представлены уровни*

*взаимодействия разработчиков на уровне CAD/CAE/CAM сервисов. Взаи-*

*модействие с системой реализовано в виде грид-портала. Описание взаимо-*

*действия компонентов системы инженерного проектирования реализовано*

*посредством потоков задач. Реализованы интерфейсы постановки задач,*

*обертывающие классические системы инженерного проектирования. Реа-*

*лизована возможность решения задач инженерного моделирования посред-*

*ством указания параметров моделирования и получения конечных резуль-*

*татов через веб-портал. В качестве недостатков этой системы можно отме-*

*тить отсутствие стандартизированного подхода к созданию виртуальных*

*испытательных стендов; отсутствие возможности конкретизации проблем-*

*но-ориентированных оболочек, обеспечивающих «подгонку» задач к кон-*

*кретным требованиям пользователя; отсутствие реализации стандарта*

*WSRF. Также, отсутствует брокер ресурсов обеспечивающий прозрачное*

*31*

*для пользователя предоставление ресурсов распределенной вычислитель-*

*ной среды, наиболее соответствующих требованиям текущей задачи.*

***1.3. В ыв од ы п о г ла ве1***

*Проведенный анализ литературы показывает, что на сегодняшний*

*день отсутствуют универсальные системы, обеспечивающие прозрачное*

*внедрение ресурсов систем инженерного проектирования в распределенные*

*вычислительные среды. В соответствии с этим остается актуальной задача*

*разработки комплексного технологического подхода для интеграции сис-*

*тем инженерного проектирования и анализа в грид на базе сервисно ориен-*

*тированной концепции с поддержкой всех этапов создания и использования*

*распределенных инженерных приложений. В качестве платформы для реа-*

*лизации системы, целесообразно использовать технологию грид-*

*вычислений на базе системы UNICORE 6.*

*32*

***Г ЛА ВА2. Т ЕХ НО ЛО ГИ Я CAEBEANS***

*В данной главе описывается технология CAEBeans, представляющей*

*собой комплекс моделей, методов и алгоритмов, направленных на автома-*

*тизированное создание иерархий распределенных проблемно-*

*ориентированных оболочек (Beans) над инженерными (CAE) пакетами на*

*основе сервисно ориентированного подхода и концепции облачных вычис-*

*лений (cloud computing).*

***2.1. О сн ов ны е к он це пц иит ех но ло ги и CAEBeans***

***2.1.1. З ад ач а и нж ен ер но гом од ел ир ов ан ия*** *З ад ач а и нж ен ер но гом од ел ир ов ан иявключает \_\_\_\_\_\_\_в себя:*

* геометрическую модель объекта исследования и (или) вычислитель-*

*ную сетку, разбивающую моделируемую область на дискретные по-*

*добласти;*

* граничные условия, физические характеристики и параметры взаимо-*

*действия отдельных компонентов исследуемой области;*

* описание неизвестных величин, значения которых требуется полу-*

*чить в результате решения задачи;*

* требования к аппаратным и программным ресурсам, обеспечиваю-*

*щим процесс моделирования.*

*П ар ам ет ро м задачи инженерного моделирования назовем величину,*

*значение которой влияет на конечный результат моделирования и может*

*варьироваться в определенном диапазоне значений. Каждый параметр за-*

*дачи инженерного моделирования имеет определенную семантику, отражая*

*свойство проблемной области, либо некоторую особенность процесса ре-*

*шения задачи [14].*

*33*

*П ол ны м д ес кр ип то ро м задачи назовем множество параметров, одно-*

*значно описывающих задачу инженерного моделирования.*

*Для решения задачи инженерного моделирования, пользователь дол-*

*жен пройти сложный технологический цикл, который обычно включает в*

*себя следующие этапы:*

* подготовка \_\_\_\_\_\_\_геометрии модели;*

* построение вычислительной сетки;*

* определение физических свойств, граничных условий и протекающих*

*процессов в анализируемой области;*

* численное решение поставленной задачи моделирования;*

* визуализация и анализ полученных результатов.*

*Каждый из этих этапов может быть реализован средствами различ-*

*ных инженерных пакетов. Например, геометрия задачи может быть выпол-*

*нена в пакете SolidWorks, после чего отправлена в систему ICEM CFD для*

*генерации качественной сетки для дальнейшего решения задачи средствами*

*пакета ANSYS CFX. В соответствии с этим для решения задачи инженер-*

*ного моделирования необходимо скоординировать совместную работу*

*множества программных компонентов, отвечающих за различные этапы*

*технологического цикла. Л ог ич ес ки м п ла но м решения задачи инженерного*

*моделирования назовем ориентированный граф, в вершинах которого могут*

*находиться у зл ы двух типов:*

* д ей ст ви я, выполняемые отдельными инженерными пакетами;*

* у зл ы у пр ав ле ни я потоком решения задачи.*

*С каждым узлом логического плана связаны два подмножества пара-*

*метров полного дескриптора задачи*

* множество параметров, определяющих входные данные для узла;*

* множество параметров, в которых сохраняются результаты работы*

*узла.*

*34*

***Р ис 5.*** *Обобщенная схема распределенного виртуального испытательного*

*стенда.*

*Определим к ла ссз ад ачи нж ен ер но гом од ел ир ов ан иякак множество*

*задач, у которых совпадает структура полного дескриптора задачи и для*

*которых можно описать единый логический план, приводящий к решению.*

***2.1.2. Р ас пр ед ел ен ны й в ир ту ал ьн ыйи сп ыт ат ел ьн ыйс те нд*** *Внедрение ресурсов CAE-пакетов в распределенные вычислительные*

*среды основывается на концепции распределенного виртуального испыта-*

*тельного стенда. Р ас пр ед ел ен ны й в ир ту ал ьн ыйи сп ыт ат ел ьн ыйс те нд (Р аВ ИС –это программно-аппаратный комплекс, обеспечивающий прове-*

*дение работ инженерного моделирования в распределенной вычислитель-*

*ной среде в рамках определенного класса задач.*

*РаВИС включает в себя (см. рис. 5):*

* и нт ер фе йс, обеспечивающий постановку определенного класса задач*

*инженерного моделирования;*

* д ра йв ер: набор программных средств, обеспечивающих использова-*

*ние сервисов распределенной вычислительной среды для проведения*

*виртуального эксперимента;*

* с ер ви сыр ас пр ед ел ен но й в ыч ис ли те ль но й с ре ды: множество вычисли-*

*тельных систем, входящих в распределенную вычислительную среду,*

*в совокупности с установленными на них программными компонен-*

*тами, обеспечивающими решение задач инженерного моделирования*

*35*

*и поддерживающими безопасные стандартизованные методы удален-*

*ного взаимодействия.*

*Пользовательский интерфейс РаВИС обеспечивает возможность п ро*

*б ле мн о-о ри ен ти ро ва нн ойпостановки конкретного класса задач инженер-*

*ного моделирования. Проблемно-ориентированная постановка задачи по-*

*зволяет сделать прозрачной гетерогенную структуру распределенной вы-*

*числительной среды и абстрагировать пользователя от механизмов реше-*

*ния задачи посредством ресурсов, предоставляемых удаленными грид-*

*сервисами. При постановке и решении задачи инженерного моделирования*

*пользователю предоставляется возможность взаимодействовать с РаВИС в*

*терминах проблемной области посредством указания значений параметров,*

*описывающих интересующий его класс задач.*

*Драйвер РаВИС выполняет следующие действия:*

* автоматизирует процесс декомпозиции задачи на типовые действия;*

* обеспечивает поиск вычислительных ресурсов, обеспечивающих оп-*

*тимальное решение поставленной задачи в распределенной вычисли-*

*тельной среде;*

* анализирует и отслеживает лицензии на программное обеспечение*

*для инженерного моделирования, доступные сервисам распределен-*

*ной вычислительной среды;*

* обеспечивает безопасное соединение и обмен информацией между*

*сервисами распределенной вычислительной среды;*

* производит постановку, мониторинг и получение результатов реше-*

*ния задач на удаленных вычислительных сервисах.*

*Сервисы распределенной вычислительной среды представляют собой*

*грид-сервисы, реализованные на базе архитектуры OGSA [62], доступ к ко-*

*торым организуется посредством стандартов WSRF [43, 116]. Сервисы пре-*

*доставляют безопасный стандартизованный доступ к ресурсам CAE36*

*пакетов, установленных на вычислительных системах распределенной вы-*

*числительной среды.*

*Процессы разработки и функционирования РаВИС определяются*

*технологией CAEBeans. Т ех но ло ги я CAEBeans –это совокупность теории и*

*практической техники, на которые опирается процесс создания и использо-*

*вания распределенных виртуальных испытательных стендов. Технология*

*CAEBeans включает в себя:*

*1) к он це пт уа ль ны е с ре дс тв а, которые определяют методы разработки и*

*структуру РаВИС;*

*2) о рг ан из ац ио нн ыес ре дс тв а, которые определяют форму труда и рас-*

*пределение обязанностей в команде разработчиков и пользователей*

*РаВИС;*

*3) п ро гр ам мн ыес ре дс тв а разработки и среду исполнения РаВИС.*

***2.2. А рх ит ек ту раCAEBeans***

*О бо ло чк а CAEBean –это основная структурная единица, формирую-*

*щая РаВИС. В соответствии с технологией CAEBeans выделяются четыре*

*слоя структуры РаВИС, каждый из которых представляется своим типом*

*оболочек CAEBeans (см. рис. 6) [13]:*

*1) концептуальный слой (проблемные CAEBeans);*

*2) логический слой (потоковые CAEBeans);*

*3) физический слой (компонентные CAEBeans);*

*4) системный слой (системные CAEBeans).*

***2.2.1. К он це пт уа ль ны й с ло й***

*Концептуальный слой РаВИС формируется на основе оболочек*

*CAEBeans, которые мы будем называть п ро бл ем ны ми. Пользовательский*

*интерфейс, предоставляемый проблемным CAEBean, является основным*

*средством взаимодействия пользователя с системой CAEBeans. Посредст-*

*вом проблемного CAEBean, ориентированного на решение конкретного*

*37*

***Р ис 6.*** *Обобщенная схема слоев РаВИС.*

*класса задач инженерного моделирования, пользователь может произвести*

*постановку задачи; проследить за ходом решения поставленной задачи; по-*

*лучить результаты решения.*

*При постановке задачи посредством проблемного CAEBean пользо-*

*вателю предоставляется возможность оперировать терминами той про-*

*блемной области, в рамках которой выполняется решение задачи. Как от-*

*мечалось в п. 2.1.1, каждая задача инженерного моделирования может быть*

*описана некоторым набором значений входных параметров. В качестве*

*примера таких параметров можно привести следующие: температура жид-*

*кости, протекающей в системе труб при моделировании трубопровода; шаг*

*и профиль резьбы при моделировании резьбового соединения труб; ско-*

*рость поступательного движения и скорость вращения трубы при модели-*

*ровании процесса закалки.*

*38*

*Параметры задачи инженерного моделирования можно разделить на*

*несколько категорий:*

* п ро бл ем ны е п ар ам ет рыописывают задачу инженерного моделиро-*

*вания в терминах предметной области;*

* п ар ам ет рыв ыч ис ли те ль но й с ре дыопределяют желаемые характери-*

*стики среды исполнения виртуального эксперимента: требования к*

*аппаратным ресурсам, минимальное время ожидания отклика уда-*

*ленного сервиса, конкретные версии программного обеспечения и*

*лицензий;*

* с лу же бн ыеп ар ам ет ры значения этих параметров не устанавливают-*

*ся пользователем, а формируются автоматически в процессе испол-*

*нения РаВИС. Они используются для обмена промежуточными дан-*

*ными между различными сервисами, а также для передачи результа-*

*тов решения пользователю.*

*Интерфейс, предоставляемый пользователю проблемным CAEBean,*

*ориентирован на решение определенного класса задач инженерного моде-*

*лирования. С его помощью можно произвести постановку задачи посредст-*

*вом указания значений параметров, характеризующих данный класс за-*

*дач [16]. Таким образом обеспечивается проблемно-ориентированная по-*

*становка задачи, не зависящая от функциональных особенностей про-*

*граммных продуктов, на основе которых будет производиться ее решение.*

*Проблемный CAEBean скрывает от пользователя «лишнюю» функциональ-*

*ность САЕ-пакета, распределенный характер вычислительной среды,*

*структуру аппаратных, программных и лицензионных ресурсов [12]. Поль-*

*зователю достаточно освоить простой интерфейс проблемного CAEBean*

*для того, чтобы иметь возможность поставить и решить задачу инженерно-*

*го моделирования с помощью сервисов грид-среды.*

*Проблемно-ориентированный пользовательский интерфейс решения*

*задачи очень трудно разработать таким образом, чтобы он подходил под*

*требования всех возможных пользователей системы одновременно. Для*

*решения данной проблемы была предложена концепция д ер ев а п ро бл ем ны х*

*39*

***Р ис 7.*** *Пример иерархии проблемных оболочек.*

*о бо ло че к [13, 14]. Она позволяет адаптировать проблемно-*

*ориентированный интерфейс конкретного класса задач инженерного моде-*

*лирования под нужды определенной категории пользователей. В основе*

*данного подхода лежит процесс формирования иерархии классов задач и*

*построения дерева проблемных оболочек CAEBeans на основе этой иерар-*

*хии (см. рис. 7).*

*Корневой элемент дерева, задающего иерархию проблемных оболо-*

*чек, представляется проблемной оболочкой, соответствующей наиболее*

*широкому классу решаемых задач. С одной стороны, она обеспечивает ре-*

*шение наиболее широкого класса задач инженерного моделирования, но с*

*другой стороны, обладает очень сложным интерфейсом с большим количе-*

*ством параметров, значения которых необходимо указать пользователю для*

*того, чтобы поставить задачу.*

*Процедуру формирования дерева проблемных оболочек можно опи-*

*сать как движение от корневого элемента к листьям. Технология CAEBeans*

*позволяет дочерним проблемным оболочкам конкретизировать класс задач,*

*решаемых родительской оболочкой, путем выделения и инкапсуляции*

*40*

*групп проблемных параметров, значения которых определяются на данном*

*уровне абстракции. Скрываемый проблемный параметр дочерней оболочки*

*может быть задан в виде константы или в виде функции, зависящей от зна-*

*чений других проблемных параметров, входящих в интерфейс данной обо-*

*лочки. Пользователю не требуется указание значений таких параметров*

*вручную при постановке задачи. Таким образом обеспечивается упрощение*

*пользовательского интерфейса.*

*Для решения конкретной инженерной задачи, пользователю предла-*

*гается оболочка соответствующая листу дерева проблемных оболочек. Та-*

*кая оболочка имеет наиболее простой интерфейс и требует от пользователя*

*минимальной квалификации. В случае если пользователю необходимо за-*

*действовать большее количество параметров, он осуществляет переход к*

*родительской оболочке, тем самым переходя на более низкий уровень абст-*

*ракции. Так пользователь может дойти до корневого проблемного*

*CAEBean, который охватывает максимально широкий класс задач в данной*

*предметной области.*

*При постановке задачи листовой проблемный CAEBean формирует*

*п ол ны й дескриптор задачи: часть проблемных параметров задает пользова-*

*тель, часть определяется внутри CAEBean [14]. При этом набор параметров,*

*входящих в полный дескриптор задачи, одинаков для всех проблемных*

*CAEBean, входящих в одно дерево проблемных оболочек [12]. Если поль-*

*зователь ставит задачу средствами корневой проблемной оболочки, то все*

*значения входных параметров указываются им вручную. Если же для по-*

*становки задачи использовалась оболочка более высокого уровня абстрак-*

*ции, значения проблемных параметров, зафиксированных на данном уров-*

*не, вычисляются по формулам, заданным разработчиком РаВИС. Сформи-*

*рованный полный дескриптор задачи передается в потоковый CAEBean,*

*представляющий логический слой РаВИС.*

*41*

***2.2.2. Л ог ич ес ки й с ло й***

*Процесс решения задачи инженерного моделирования можно пред-*

*ставить в виде совокупности взаимосвязанных д ей ст ви й и нж ен ер но гом о-*

*д ел ир ов ан ия, таких как определение геометрии моделируемой области, ге-*

*нерация расчетной сетки, определение физики решаемой задачи, проведе-*

*ние компьютерного моделирования, визуализация и анализ результатов*

*компьютерного моделирования.*

*П от ок ов ыйCAEBean представляет собой л ог ич ес ки й п ла н решения*

*задач инженерного моделирования [13]. Для формирования логического*

*плана используются элементы нотации д иа гр ам мыд ея те ль но ст и стандар-*

*та UML 2.0 [4, 5, 17, 35, 104, 108]. В соответствии с определением данным в*

*п. 2.1.1 логический план решения задачи представляет собой ориентиро-*

*ванный граф, в вершинах которого могут находиться узлы двух типов:*

*действия и узлы управления, а ребра представляют собой отношения,*

*управляющие потоками управления между двумя узлами (см. рис. 8).*

*П от оку пр ав ле ни я –это ребро, задающее протекание управления, но*

*не данных. Информация, необходимая для исполнения любого узла логиче-*

*ского плана содержится в полном дескрипторе задачи. В ходе решения за-*

*дачи, любой узел может обратиться к дескриптору для получения значений*

*входных параметров и сохранения результатов в соответствующих выход-*

*ных параметрах.*

*Процесс выполнения деятельности можно представить себе как обход*

*экземпляра графа логического плана, содержащего маркеры управления.*

*М ар ке р у пр ав ле ни я указывает на наличие управления на некотором этапе*

*процесса вычисления, представленным узлом или потоком. Маркеры*

*управления не имеют внутренней структуры. Выполнение действия разре-*

*шается, если во всех его входных потоках присутствуют маркеры. Некото-*

*рые виды узлов управления могут начинать работу в момент, когда на*

*42*

***Р ис 8.*** *Потоковый CAEBean.*

*заданном подмножестве входных ребер появятся маркеры управления. Ко-*

*гда работа узла завершается, в каждом из его исходящих потоков управле-*

*ния появляется маркер управления.*

*Виды узлов логического плана решения задачи представлены в таб-*

*лице 1.*

*Начальный узел является управляющим узлом, указывающим место*

*начала исполнения логического плана. Начальный узел не может иметь ни*

*одного входа, но может иметь один выход. Когда начинается выполнение*

*логического плана, на начальный узел помещается маркер управления. Ло-*

*гический план может иметь несколько начальных узлов. В этом случае, в*

*начале исполнения маркеры управления помещаются на каждый из них.*

*Такая конструкция имитирует параллельность исполнения.*

*Выполнение конечного узла вызывает принудительное завершение*

*всех потоков в данном логическом плане и завершение выполнения логиче-*

*ского плана в целом.*

*43*

***Т аб ли ца1.*** *Виды узлов логического плана решения задач*

*Название Обозначение Описание*

*Начальный*

*узел*

*Управляющий узел, с которого начинается выпол-*

*нение логического плана.*

*Конечный*

*узел*

*Управляющий узел, обозначающий завершение*

*выполнения логического плана.*

*Действие*

*Узел, обеспечивающий выполнение определенного*

*действия инженерного моделирования.*

*Разветвление*

*Управляющий узел, обеспечивающий выбор даль-*

*нейшего маршрута исполнения логического плана,*

*в зависимости от значения булева выражения.*

*Объединение*

*Управляющий узел, в котором сходятся два или*

*более альтернативных путей управления.*

*Разделение*

*Управляющий узел, обеспечивающий копирование*

*входного маркера на каждый из выходов и иниции-*

*рующий исполнение нескольких параллельных по-*

*токов управления.*

*Слияние*

*Управляющий узел, синхронизирующий несколько*

*параллельных потоков управления.*

*Р аз ве тв ле ни е –это узел принятия решения, имеющий один входной*

*поток несколько выходных потоков. На каждом из выходных потоков по-*

*мещается ожидаемое значение условия. Маркер управления помещается в*

*тот выходной поток, условие которого совпадает со значением выражения,*

*вычисляемого на входе в узел разветвления. При этом необходимо, чтобы*

*значения всех параметров, участвующих в логической формуле, были уста-*

*новлены в дескрипторе задачи до вызова узла разветвления, иначе резуль-*

*тат вычислений будет не определен. Условия выходных потоков не должны*

*перекрываться (иначе поток управления будет неоднозначным), но при*

*этом должны учитываться все возможные варианты.*

*О бъ ед ин ен ие–это соединение двух или более маршрутов управле-*

*ния. Узел объединения имеет несколько входных потоков и один исходя-*

*щий. Если маркер приходит на узел объединения по одному из входов, он*

*немедленно копируется на выход. Допускается комбинация из разветвле-*

*ния и объединения, когда у узла есть несколько входящих и несколько по-*

*меченных исходящих переходов.*

*44*

*Р аз де ле ни е обеспечивает расщепление одного потока управления на*

*несколько параллельных. Входной маркер управления копируется на каж-*

*дый из выходов. После разделения деятельность, ассоциированная с каж-*

*дым потоком, выполняется параллельно. При этом реализация параллель-*

*ной деятельности должна быть такова, чтобы подмножество входных и вы-*

*ходных параметров узлов в каждом параллельном потоке не пересекалось с*

*подмножествами выходных параметров узлов других потоков этой парал-*

*лельной деятельности.*

*С ли ян иепредставляет синхронизацию нескольких параллельных по-*

*токов управления. Узел соединения может иметь несколько входящих по-*

*токов управления и один исходящий. Когда по каждому входному ребру*

*приходит маркер управления, они потребляются, а на выходное ребро по-*

*мещается новый маркер. В момент соединения, параллельные потоки син-*

*хронизируются, то есть каждый из них ждет, когда все остальные достиг-*

*нут точки соединения, начиная с которой продолжит выполняться один по-*

*ток управления. Между точками разделения и слияния должен поддержи-*

*ваться баланс. Это означает что число потоков, исходящих из точки разде-*

*ления, должно быть равно числу потоков, приходящих в соответствующую*

*точку слияния.*

*У зе л д ей ст ви я реализует определенное действие инженерного моде-*

*лирования. Каждый узел действия обладает списком входных и выходных*

*параметров, значения которых хранятся в полном дескрипторе задачи. При*

*исполнении узла действия, потоковый CAEBean обеспечивает взаимодей-*

*ствие с дескриптором задачи: получение значений входных параметров для*

*инициации работы действия и запись значений выходных параметров, вы-*

*численных в результате исполнения действия. Действие реализуется соот-*

*ветствующим компонентным CAEBean, находящимся на физическом слое*

*РаВИС.*

*45*

***2.2.3. Ф из ич ес ки й с ло й***

*К ом по не нт ны е CAEBeans, представляющие физический слой РаВИС,*

*отвечают за процесс постановки и решения отдельных действий инженер-*

*ного моделирования средствами конкретных инженерных пакетов. При его*

*разработке прикладной программист ориентируется на стандарты поста-*

*новки задач, поддерживаемые конкретным CAE-пакетом: форматы файлов*

*постановки задачи и результатов решения; методы автоматизации процесса*

*решения; языки описания процесса моделирования; параметры входной*

*строки и др.*

*Основная функция компонентного CAEBean –преобразование про-*

*блемно-ориентированного описания действия инженерного моделирования*

*в к ом по не нт ноо ри ен ти ро ва нн уюф ор му. На основе значений входных па-*

*раметров, компонентный CAEBean формирует набор файлов, содержащих*

*описание действия в формате, поддерживаемом целевым CAE-пакетом.*

*Также, на основе информации о формате и семантике параметров команд-*

*ной строки CAE-пакета, компонентный CAEBean формирует и передает*

*команду запуска процесса решения.*

*По окончании процесса решения, компонентный CAEBean обеспечи-*

*вает преобразование компонентно-ориентированных результатов решения*

*задачи в проблемно-ориентированные. На основе информации о формате*

*результатов, компонентный CAEBean обеспечивает извлечение значений*

*выходных параметров действия из файлов решения задачи.*

*Компонентный CAEBean задает абстрактное действие, которое не*

*может быть реализовано само по себе, так как не обеспечено реальными*

*программными, аппаратными и лицензионными ресурсами. В процессе ис-*

*полнения, каждому компонентному CAEBean должен быть найден и пре-*

*доставлен физический ресурс, отвечающий всем требованиям, необходи-*

*мым для реализации специфицированной в нем деятельности. Поиск и пре-*

*46*

*доставление физических ресурсов обеспечивается брокером ресурсов [12],*

*описание которого приведено в п. 3.6. В рамках системы CAEBeans, физи-*

*ческие ресурсы реализуются посредством системных CAEBeans, описы-*

*ваемых в следующем разделе.*

***2.2.4. С ис те мн ыйс ло й***

*С ис те мн ымCAEBean называется оболочка системного слоя РаВИС,*

*предоставляющая функциональные возможности физического ресурса в*

*распределенной вычислительной среде и обеспечивающая сервисно ориен-*

*тированный подход к постановке задач и получению результатов.*

*В рамках технологии CAEBeans, для любого инженерного пакета*

*создается набор системных CAEBeans, каждый из которых представляет*

*реализацию определенного действия инженерного моделирования (см. п.*

*2.2.2). Например, для пакета ANSYS CFX может быть создано три систем-*

*ных CAEBeans:*

*1. Pre\_CAEBean (определение граничных условий и физических свойств*

*сред моделируемой области);*

*2. Solver\_CAEBean (расчет задачи вычислительной гидродинамики);*

*3. Post\_CAEBean (анализ и визуализация результатов моделирования).*

*При этом для реализации таких оболочек могут быть использованы*

*различные средства, в зависимости от программного интерфейса, предос-*

*тавляемого конкретным CAE-пакетом. В качестве примера таких средств*

*можно привести: лог-файлы и макросы, поддерживаемые CAE-пакетом*

*ANSYS Mechanical; файлы сценариев на языке Python, поддерживаемые*

*CAE-пакетом Abaqus; текстовый файл CCL (Command Language File - Файл*

*языка команд) поддерживаемый решателем ANSYS CFX-Solver [13].*

*Системный CAEBean является проблемно-независимым элементом*

*архитектуры РаВИС. С одной стороны, множество компонентных CAEBeans,*

*обеспечивающих реализацию различных действий инженерного мо-*

*47*

*делирования, могут обращаться к одному и тому же системному CAEBean,*

*если предоставляемый им физический ресурс поддерживает решение задач*

*такого типа. С другой стороны, в распределенной вычислительной среде*

*может находиться сколь угодно много однотипных системных CAEBean. В*

*этом случае выбирается тот, который свободен в данный момент.*

*Системный CAEBean предоставляет следующие возможности досту-*

*па к физическому ресурсу:*

*1) обеспечение изолированного рабочего пространства для каждого*

*действия инженерного моделирования, реализуемого на вычисли-*

*тельной системе;*

*2) загрузка исходных данных, необходимых для реализации действия*

*инженерного моделирования;*

*3) предоставление программного интерфейса для удаленного запуска*

*действия инженерного моделирования;*

*4) предоставление служебной информации и промежуточных результа-*

*тов реализации действия;*

*5) передача результатов реализации действия на требуемый узел грид-*

*сети.*

***2.2.5. В за им од ей ст ви е с ло евР аВ ИС*** *Рассмотрим процедуру постановки и решения задачи инженерного*

*моделирования в соответствии с архитектурой CAEBeans (см. рис. 9).*

*1. При постановке задачи инженерного моделирования проблемный*

*CAEBean формирует полный дескриптор задачи, занося туда значе-*

*ния проблемных параметров, указанные пользователем, и вычисляя*

*значения зафиксированных параметров.*

*2. Процесс решения задачи инженерного моделирования реализуется*

*потоковым CAEBean посредством выполнения логического плана*

*48*

***Р ис 9.*** *Взаимодействие оболочек РаВИС в процессе решения задачи*

*инженерного моделирования****.***

*решения задачи. При реализации очередного действия инженерного*

*моделирования, потоковый CAEBean получает из полного дескрип-*

*тора задачи и передает компонентному CAEBean значения требуемых*

*входных параметров. Значения всех входных параметров должны*

*быть определены до момента реализации действия.*

*3. На основе значений входных параметров компонентный CAEBean*

*формирует постановку действия для конкретного CAE-пакета, в со-*

*ответствии с поддерживаемыми форматами файлов и команд.*

*4. Посредством брокера ресурсов производится поиск и предоставление*

*системного CAEBean, обеспечивающего наилучшие возможности по*

*реализации текущего действия. На предоставленный грид-ресурс*

*производится копирование файлов постановки действия и запускает-*

*ся исполнение действия инженерного моделирования.*

*49*

*5. Инженерный пакет производит решение поставленной задачи и фор-*

*мирует набор файлов с результатами.*

*6. Компонентный CAEBean получает от удаленного сервиса необходи-*

*мые ему файлы с результатами решения.*

*7. Компонентный CAEBean производит разбор результатов действия и*

*извлекает значения требуемых выходных параметров из файлов, по-*

*лученных в результате решения. Значения данных параметров пере-*

*даются потоковому CAEBean.*

*8. Потоковый CAEBean помещает полученные значения выходных па-*

*раметров в полный дескриптор задачи и завершает исполнение ком-*

*понентного CAEBean. Производится вызов следующего узла логиче-*

*ского плана решения задачи.*

*9. При завершении исполнения логического плана результаты модели-*

*рования в виде значений выходных параметров, сохраненных в пол-*

*ном дескрипторе задачи, передаются проблемному CAEBean.*

***2.3. П ро гр ам мн ыес ре дс тв а CAEBeans***

*Технологию разработки и исполнения распределенных испытатель-*

*ных стендов невозможно эффективно реализовать без использования спе-*

*циализированных программных средств, ориентированных на поддержку*

*данной технологии. Программные средства CAEBeans можно разбить на*

*два типа: средства исполнения и средства разработки РаВИС (см. рис. 10).*

***2.3.1. Р аз ра бо тк а Р аВ ИС*** *Разработка проблемно-ориентированной части распределенного вир-*

*туального испытательного стенда ведется в системе К он ст ру кт ор. Посред-*

*ством Конструктора прикладной программист формирует оболочки кон-*

*цептуального, логического и физического слоя РаВИС, отвечающие за ре-*

*шение конкретного класса задач инженерного моделирования.*

*50*

***Р ис 10.*** *Программные средства CAEBeans.*

*На концептуальном слое поддерживается создание корневого про-*

*блемного CAEBean, описание параметров задачи инженерного моделиро-*

*вания и полного дескриптора задачи. Также обеспечивается возможность*

*формирования дерева проблемных оболочек посредством наследования и*

*конкретизации наборов параметров родительских проблемных оболочек.*

*На логическом слое поддерживается визуальное проектирование ло-*

*гического плана решения задачи в соответствии с ограничениями и семан-*

*тикой узлов, определенных в пункте 2.2.2. Поддерживается подключение*

*физических CAEBean к узлам действий.*

*На физическом слое поддерживается описание интерфейса физиче-*

*ского CAEBean, выбор входных и выходных параметров данного CAEBean,*

*выбор необходимых файлов, обеспечивающих реализацию действия. Более*

*детально реализация конструктора рассмотрена в пункте 3.2.*

*Физический*

*слой*

*Системный*

*слой*

*П от ок ов ый CAEBean*

*К ом по не нт ны е*

*CAEBeans*

*С ис те мн ыеCAEBeans*

*П ро бл ем ны е*

*CAEBeans*

***К ли ен т*** *Концептуальный*

*слой*

*Логический*

*слой*

***Б ро ке р р ес ур со в***

***Ц ел ев ыес ис те мы Г ри д***

***И сп ол не ни е***

*П от ок ов ый CAEBean*

*К ом по не нт ны е*

*CAEBeans*

*П ро бл ем ны е*

*CAEBeans*

***Р аз ра бо тк а***

*51*

***2.3.2. И сп ол не ни е Р аВ ИС*** *В соответствии с технологией CAEBeans работа РаВИС осуществля-*

*ется с помощью следующих четырех программных систем:*

*1) клиент;*

*2) сервер;*

*3) брокер ресурсов;*

*4) набор целевых систем.*

*К ли ен т предоставляет унифицированный интерфейс конкретного*

*РаВИС, который был описан в п. 2.1.2. Клиент обеспечивает постановку и*

*получение результатов решения задачи инженерного моделирования. Под-*

*держивается возможность автоматической генерации пользовательского*

*интерфейса РаВИС на основе описания параметров требуемого проблемно-*

*го CAEBean. Подробное описание клиента будет дано в п. 3.3.*

*С ер ве р отвечает за хранение и исполнение РаВИС. На сервере произ-*

*водится анализ введенных пользователем значений параметров РаВИС и*

*запускается процесс исполнения логического плана решения задачи. Сер-*

*вер обеспечивает исполнение логического плана решения задачи. Более*

*подробно архитектура сервера будет описана в п. 3.4.*

*Б ро ке р р ес ур со в –это автоматизированная система регистрации, ана-*

*лиза и предоставления ресурсов распределенной вычислительной среды.*

*Брокер представляет собой промежуточное программное обеспечение,*

*обеспечивающее оптимальный выбор и виртуализацию доступа к наборам*

*ресурсов в гетерогенной вычислительной среде. Архитектура брокера ре-*

*сурсов и алгоритмы предоставления физических ресурсов описаны в п. 3.6.*

*В совокупности, сервер и брокер ресурсов реализуют драйвер распределен-*

*ного виртуального испытательного стенда, определенный в п. 2.1.2.*

*Ц ел ев аяс ис те ма–это совокупность грид-сервисов, которые имеют*

*доступ к пространству программных, аппаратных и лицензионных ресурсов*

*52*

***Р ис 11.*** *Распределение компетенции пользователей РаВИС.*

*некоторого узла грид, и поддерживает аутентификацию и авторизацию*

*пользователей. Сервисы, предоставляемые каждой отдельной целевой сис-*

*темой можно разделить на две категории:*

*1. системные CAEBeans;*

*2. служебные сервисы.*

*С ис те мн ыеCAEBeans инкапсулируют ресурсы, предоставляемые от-*

*дельными инженерными пакетами. Системные CAEBeans обеспечивают*

*удаленную постановку действия инженерного моделирования и получение*

*результатов его исполнения.*

*С лу же бн ыес ер ви сы, входящие в целевую систему, поддерживают*

*работу вычислительной системы в грид-среде совместно с программным*

*обеспечением системы CAEBeans. Более подробно структура целевой сис-*

*темы будет описана в п. 3.5.*

***2.4. О рг ан из ац ияр аб отвс ис те меCAEBeans***

*Разработка РаВИС –это сложный процесс, который невозможен без*

*участия специалистов в различных областях информационных технологий*

*и инженерного проектирования и анализа. Можно выделить три основные*

*роли в разработке и использовании РаВИС (см. рис. 11):*

*1) инженер;*

*2) прикладной программист;*

*3) системный программист.*

***К ом пе те нц ия И нж ен ер но е***

***п ро ек ти ро ва ни е***

***И нф ор ма ци он ны е***

***т ех но ло ги и***

*Инженер*

*Прикладной*

*программист*

*Системный*

*программист*

*53*

***2.4.1. И нж ен ер*** *И нж ен ер–это пользователь системы CAEBeans, решающий задачу*

*инженерного моделирования на основе созданного для этой цели распреде-*

*ленного виртуального испытательного стенда. Доступ к возможностям*

*РаВИС инженер получает посредством Клиента системы CAEBeans, пре-*

*доставляющего пользовательский интерфейс соответствующего проблем-*

*ного CAEBean. В связи с этим, инженер должен обладать минимальной*

*компетенцией в области информационных технологий, которой достаточно*

*чтобы запустить клиента системы, произвести постановку задачи инженер-*

*ного моделирования и получить результаты решения.*

*При разработке РаВИС инженер представляет собой постановщика*

*задачи инженерного моделирования, обладающего информацией о физиче-*

*ской сущности решаемой задачи. Таким образом, инженер обладает макси-*

*мальной компетенцией в сфере инженерного моделирования поставленной*

*задачи среди всех пользователей системы CAEBeans.*

***2.4.2. П ри кл ад но й п ро гр ам ми ст*** *П ри кл ад но й п ро гр ам ми ст–это специалист в области разработки*

*РаВИС посредством технологии CAEBeans. Это пользователь системы, ко-*

*торый производит разработку РаВИС в Конструкторе CAEBeans на основе*

*информации, полученной от инженера. Прикладной программист должен*

*обладать широкими знаниями как в области инженерного моделирования,*

*так и в области информационных технологий.*

*С точки зрения инженерного моделирования прикладной програм-*

*мист должен достаточно глубоко разбираться в поставленной задаче, чтобы*

*выявить возможные пути ее параметризации и создать дерево проблемных*

*CAEBean. Прикладной программист должен уметь описать оптимальный*

*технологический цикл решения данной задачи при формировании потоко-*

*54*

*вого CAEBean и выбрать CAE-пакеты, обеспечивающие решение постав-*

*ленной задачи.*

*С точки зрения информационных технологий, прикладной програм-*

*мист должен обладать глубокими знаниями о методах взаимодействия с*

*CAE-пакетами, методах автоматизированной постановки задач инженерно-*

*го моделирования, форматах входных и выходных файлов, методах их ав-*

*томатического формирования и извлечения результатов решения. Также*

*прикладной программист должен иметь хорошее представление о техноло-*

*гиях распределенных вычислений, чтобы адекватно оценить и по возмож-*

*ности минимизировать время исполнения логического плана решения зада-*

*чи.*

***2.4.3. С ис те мн ыйп ро гр ам ми ст*** *С ис те мн ыйп ро гр ам ми ст–это специалист в области информацион-*

*ных технологий, отвечающий за формирование распределенной вычисли-*

*тельной среды для исполнения РаВИС. Он обеспечивает установку и на-*

*стройку программных средств системы CAEBeans, а также CAE-пакетов,*

*средствами которых производится решение задач инженерного моделиро-*

*вания.*

*При разработке РаВИС системный программист отвечает за создание*

*и установку системных CAEBeans, обеспечивающих удаленную постановку*

*и решение задач средствами инженерных пакетов. Программный интер-*

*фейс разработанного системного CAEBean передается прикладному про-*

*граммисту для возможности дальнейшей разработки компонентного*

*CAEBean на базе ресурсов, предоставляемых данным системным CAEBean.*

***2.5. П ар ам ет ри че ск ием од ел и п ро из во ди те ль но ст и Г ри д***

*Принцип работы и функциональность грид приложений значительно*

*отличаются от обычных последовательных и параллельных систем. Основ-*

*ное отличие –это возможность агрегирования и совместного использова-*

*55*

*ния больших наборов гетерогенных ресурсов, распределенных между гео-*

*графически-разделенными областями. Во многих случаях это приносит*

*большие выгоды, например, когда приложение требует ресурсов, недоступ-*

*ных в рамках одного узла, оно может затребовать ресурсы у других узлов,*

*подключенных к грид [46].*

*Но такое сложное поведение несет в себе и определенные проблемы.*

*К высоко-гетерогенной, динамически-формируемой распределенной среде*

*очень трудно напрямую применить такие традиционные метрики произво-*

*дительности, как скорость вычислений, пропускная способность канала*

*и др. В связи с этим, для оценки качества предоставляемого сервиса требу-*

*ется использование специализированных метрик.*

*Предположим, что в грид-среде доступно . ресурсов и существует*

*система распределения заданий ., обеспечивающая распределение постав-*

*ленных задач . ∈ . на доступные ресурсы. В рамках данной системы, каж-*

*дое задание может быть разбито на действия . ∈ .. Количество заданий в*

*системе | .|; количество действий в задаче | .|. При постановке задачи, ука-*

*зывается время . ., до которого пользователь желает получить результаты*

*решения.*

*Каждая задача . и все ее действия . ∈ . поступают в грид в момент*

*времени . .. В связи с тем, что грид работает в “online”-режиме, значение . .*

*не известно заранее для большинства задач. Как только появляется опреде-*

*ленная задача, производится планирование ее работы, после чего произво-*

*дится поиск и выделение ресурсов необходимых для запуска.*

***2.5.1. М ет ри ки з ав ис ящ иео т в ре ме ни*** *Предположим, что в результате финального распределения ., каждое*

*действие . ∈ . будет исполнено за время . .( .). Таким образом, задача .*

*может быть решена не раньше, чем за время*

*56*

*. .( .) = max*

*.∈ . . .( .).*

*Определим время реализации действия . ∈ . как . . . Таким образом, время*

*решения . . задачи . может быть вычислено следующим образом:*

*. . = . .( .) −min*

*.∈ . ( . .( .) − . .).*

*Полученные величины позволяют оценить интегральные свойства*

*грид-среды. Для анализа качества сервиса, предоставляемого грид-средой*

*можно использовать показатель максимального опоздания задач:*

*. . . . = max*

*.∈ . . . .( .) − . . ..*

*При оптимизации работы распределенной среды необходимо стремиться к*

*минимизации значения данного показателя. Также, можно использовать*

*показатель . ., определяемый как количество опоздавших задач: . ∈ . ∧ . . > . .. Такой показатель предоставляет информацию о количестве невы-*

*полненных пользовательских запросов.*

*Потребление ресурсов . . . определенной подзадачей определим как*

*произведение соответствующего времени решения на количество исполь-*

*зуемых ресурсов:*

*. . . = . . ∙ . ..*

*Следовательно, мы можем определить потребление ресурсов опреде-*

*ленным заданием (1), и всего перечня задач планировщика соответствен-*

*но (2).*

*. . . = . . . .*

*.∈ . ; (1)*

*. .( .) = . . . .*

*.∈ . . (2)*

*Используя определение суммарного потребления ресурсов, можно*

*определить величину использования . (3) доступных ресурсов.*

*57*

*. =*

*. .( .)*

*. ⋅ .max*

*.∈ . . .( .) −min*

*.∈ . . . .( .) − . . . .*

*. (3)*

*Данная величина характеризует насколько оптимально используются ре-*

*сурсы, доступные в распределенной сети.*

*В процессе работы грид, очень часто возникают ситуации, когда в*

*процессе исполнения задачи происходит сбой. Тогда задание должно быть*

*запущено несколько раз для того, чтобы успешно выполниться. Вследствие*

*этого, мы можем определить полное потребление ресурсов . .{ ., .} . . . . и*

*полную величину использования ресурсов . . . . ., как соответствующую ве-*

*личину плюс затраты на исполнение сбойных заданий. В связи с этим,*

*можно определить метрику растрат:*

*. . . . . = . . . . . − .,*

*которая определяет динамическую величину ошибок грид системы и долж-*

*на быть минимизирована владельцем вычислительных ресурсов.*

*Так как пользователи и администраторы часто выдвигают различные*

*(и даже конфликтующие) требования к грид системе, очень трудно подоб-*

*рать метрику, которая удовлетворяла бы всех. С точки зрения пользователя,*

*возможно выделить метрики среднего времени ответа (4) (Average Response*

*Time – . . .) и среднего времени ожидания (5) (Average Wait Time – . . .)*

*[107]:*

*. . . =*

*1*

*│ . ¦*

*. . . .( .) .,*

*.∈ . (4)*

*. . . =*

*1*

*│ . ¦*

*. . . .( .) − . . ..*

*.∈ . (5)*

*Значение параметра . . . характеризует, насколько быстро происхо-*

*дит решение задач пользователей. С другой стороны, значение параметра*

*. . . интересно пользователям, которые производят постановку относи-*

*тельно небольших заданий.*

*58*

*Относительно хороший и простой метод измерения справедливости*

*использования ресурсов это расчет девиации среднего времени ожидания:*

*. . . . =*

*1*

*| .|*

*. . . . . . . . − . .*

*. . .*

*| .|*

*.∈ . .*

*.*

*.∈ . ,*

*где . . . = (С .( .) − . .).*

*Для достижения оптимальных результатов работы грид, необходимо*

*добиться минимизации параметра . . . . каждым владельцем ресурсов.*

*Также, важна обработка результатов мониторинга работы грид сис-*

*темы. В этом случае, можно использовать метрику эффективности грид*

*(Grid Efficiency –GE) [107]:*

*. . =*

*Σ(( . . . . . . . . − . . . . . . . . .∈ . . .) × . . . . . × . . . . . . . . .)*

*( . . . . . . . . . . . . . . − . . . . . . . . . . . . . . . . . .) ×Σ .∈ .( . . . . . × . . . . . . . . .)*

*×100% , (6)*

*где ( . . . . . . . . . . . . . . − . . . . . . . . . . . . . . . . . .) –это время работы системы;*

*. . . . . и . . . . . . . . . –это количество процессоров использованных*

*задачей . и их производительность;*

*. . . . . и . . . . . . . . . –это количество процессоров в машине . и их*

*производительность.*

***2.5.2. М ет ри ки з ав ис ящ иео т о бъ ем а р аб от ы***

*В современных грид системах, возможность завершить исполнение*

*данного объема работы может быть даже более важным, чем ускорение,*

*полученное посредством такого исполнения (требуется отметить, что зада-*

*чи, исполняемые в грид средах, могут быть значительно сложнее тех зада-*

*ний, которые исполняются в традиционных параллельных системах, на-*

*пример потоки заданий обладают значительно более сложной логической*

*структурой, чем пакеты задач). Грид требует переопределения понятия*

*ошибки приложения: грид приложение, которое не смогло успешно выпол-*

*ниться в рамках отведенного ей бюджета, генерирует сообщение об ошиб-*

*59*

*ке, как только обнаружится невозможность успешного исполнения. Напри-*

*мер, сбой может произойти вследствие того, что не могут быть найдены ре-*

*сурсы для выполнения вычислений или в связи с наступлением крайнего*

*срока работы приложения. Используя это понятие, отказоустойчивость*

*можно определить как возможность на как можно больший срок перено-*

*сить время появления ошибки, пока есть хоть какие-то шансы того, что*

*приложение завершится успешно [74].*

*Определим метрику з ав ер ше нн ог о о бъ ем а р аб от ы (Workload*

*Completion) как отношение успешно завершенных задач к объему всех за-*

*дач, поставленных планировщику грид-среды.*

*. . =*

*Σ .∈ . .( . . . . . . . . . .) 1*

*| .|*

*.*

*Данная метрика позволяет определить ограничения грид системы, и*

*ее максимизация может быть основной целью как пользователей, так и вла-*

*дельцев ресурсов. С другой стороны, . . имеет некоторые ограничения с*

*токи зрения владельцев ресурсов, так как задачи с меньшим количеством*

*действий имеют большее влияние на данную величину.*

*В качестве дополнительной метрики предлагается ввести метрику з а-*

*в ер ше ни я д ей ст ви й (Task Completion), которую можно определить как ко-*

*личество завершенных действий к общему количеству действий, исполнен-*

*ных в рамках системы распределения заданий:*

*. . =*

*Σ .∈ . . .∈ . .( . . . . . . . . . .) 1*

*Σ .∈ .| .|*

*.*

*Также, можно ввести понятие завершения разблокированных дейст-*

*вий (Enabled Task Completion), где под понятием разблокированного дейст-*

*вия мы будем понимать те действия, которые могут быть выполнены толь-*

*ко после того, как все зависимости для данного действия будут выполнены:*

*. . . =*

*Σ .∈ . . .∈ . .( . . . . . . . . . .) 1*

*Σ .∈ . . .∈ . .( . . . . . . . .) 1*

*.*

*60*

*Таким образом, владельцы ресурсов должны стремиться к максими-*

*зации . . .. Если метрики . . и . . . значительно разнятся, то необходимо*

*принять специальные меры для обеспечения выполнения критичных дейст-*

*вий (таких действий, от которых зависит исполнение большого количества*

*других действий).*

***2.5.3. А да пт ац иям од ел ейп ро из во ди те ль но ст и***

*При использовании моделей производительности в грид для базовых*

*вычислительных приложений применительно к проектируемой системе*

*CAEBeans необходимо учитывать нюансы функционирования грид-*

*сервисов данной системы.*

*Потоковый CAEBean****,*** *в соответствии с логическим планом решения*

*поставленной задачи, производит последовательную реализацию действий*

*посредством соответствующих компонентных оболочек CAEBeans. Следо-*

*вательно, обращение к потоковой оболочке инициирует формирование за-*

*дачи инженерного моделирования, которая разбивается на атомарные дей-*

*ствия, соответствующие компонентным CAEBean, участвующим в решении*

*задачи. Таким образом, любая задача CAEBeans разворачивается в после-*

*довательность действий, которые, скорее всего, зависят друг от друга по*

*данным и решаются, в общем случае, на различных вычислителях.*

*Таким образом, возможно определить базовые параметры исполнения*

*системы CAEBeans следующим образом. В рамках системы распределения*

*заданий ., каждое задание . ∈ . представлено отдельной задачей, постав-*

*ленной потоковому CAEBean. Каждая задача разбита на действия . ∈ ., в*

*простейшем случае, соответственно количеству вызовов компонентных*

*CAEBeans. Особенность процесса решения типовых задач инженерного*

*моделирования состоит в том, что отдельные действия, на которые раскла-*

*дывается задача, в большинстве случаев зависят друг от друга по данным, в*

*61*

*связи с чем высока степень последовательно-исполняемых участков логи-*

*ческого плана.*

*Одним из наиболее важных показателей является показатель среднего*

*времени ответа . . . (4). Предположив, что последовательные участки ло-*

*гического плана являются доминирующими при решении задачи инженер-*

*ного моделирования, показатель ART можно расписать следующим обра-*

*зом:*

*. . . =*

*1*

*│ . ¦*

*. . . . . . . . . . − . . . . . . . . . . . . =*

*1*

*│ . ¦*

*. . . . . . . . . . . . .*

*.∈ . .*

*.∈ . ,*

*.∈ . (7)*

*где . . . . . . . . . . –это, суммарные временные затраты на реализацию дейст-*

*вия ..*

*Применительно к модели CAEBeans мы можем расписать значение*

*величины . . . . . . . . . . следующим образом:*

*. . . . . . . . . . = . . . . . . . . . . . . . . . . + . . . . . . . . . . . +*

*. . . . . . . . . . . . . . . + . . . . . . . . . . . . . ,*

*(8)*

*где . . . . . . . . . . . . . . . . –время подготовки к решению и поиска необходи-*

*мых вычислительных ресурсов; . . . . . . . . . . . - время загрузки начальных*

*данных для реализации действия на найденный вычислительный узел;*

*. . . . . . . . . . . . . . . –время непосредственной реализации действия;*

*. . . . . . . . . . . . . –время загрузки результатов действия.*

***2.5.4. О це нк а п ро из во ди те ль но ст и т ех но ло ги и CAEBeans***

*Для оценки приемлемых величин параметра ART был произведен*

*анализ текущих задач, решаемых суперкомпьютерным центром*

*ЮУрГУ [18]. В результате, исходя из статистики использования вычисли-*

*тельных ресурсов суперкомпьютерного центра за последние 6 месяцев, бы-*

*ли получены следующие значения параметров их использования:*

*62*

* средний объем затрат на решение расчетной задачи: .=1500 машино-*

*минут;*

* среднее количество процессоров, выделяемых на одну задачу: . = 10*

*процессоров (из 50 доступных);*

* среднее время получения результата вычисления: .р еш= .*

*. = 150 ми-*

*нут;*

* среднее время подготовки к запуску задания: .з ап= 2 минуты*

* среднее количество одновременно выполняемых заданий: | .| = 4.*

*При оценке параметра ART, были использованы следующие допуще-*

*ния:*

* среднее количество действий соответствует количеству шагов типо-*

*вого технологического цикла решения задач численного моделирова-*

*ния: формирование геометрии задачи, построение расчетной сетки,*

*формирование физики задачи, решение задачи, анализ результатов*

*решения. Таким образом, | .| = 5;*

* объем данных, передаваемых для решения задачи составляет порядка*

*100Mb. Таким образом, пересылка данных для постановки задачи и*

*для получения результатов решения на скорости 1,5MB/с составляет*

*порядка .п ер ед ач и= 2 минуты.*

*Также, при оценке общих затрат необходимо учитывать, что в на-*

*стоящее время посредством вычислительных ресурсов суперкомпьютерно-*

*го центра ЮУрГУ в основном решаются только наиболее ресурсоемкие*

*этапы задач компьютерного моделирования. Временные затраты на реше-*

*ние других этапов составляют порядка .*

*. . . от временных затрат на наиболее*

*ресурсоемкий этап. Таким образом, учитывая один ресурсоемкий этап и че-*

*тыре относительно нересурсоемких этапа вычислений, мы получаем сле-*

*дующую оценку параметра . . .:*

*63*

*. . . = 4 ⋅ . .з ап+ 2 ⋅ .п ер ед ач и +*

*.р еш 100*

*. + 1 ⋅ . .з ап+ 2 ⋅ .п ер ед ач и + .р еш . =*

*= 4 ⋅(2 + 2 ⋅2 + 15) + 1 ⋅(2 + 2 ⋅2 + 150) = 84 + 156 = 240 (м ин.*

*(9)*

*Таким образом, полученная оценка параметра ART для системы*

*CAEBeans позволяет судить о потенциально хорошей эффективности ис-*

*пользования данной системы для автоматизации решения типовых задач*

*инженерного моделирования. Также полученная оценка позволяет опреде-*

*лить потенциальные возможности повышения эффективности использова-*

*ния разрабатываемой системы посредством кэширования результатов пре-*

*дыдущих этапов вычислений для их возможного повторного использования*

*на последующих этапах моделирования определенных классов задач и по-*

*вышения степени параллельности реализуемых действий.*

***2.6. В ыв од ы п о г ла ве2***

*В главе 2 была рассмотрена архитектура CAEBeans, определяющая*

*структуру и принципы работы РаВИС. В основе архитектуры CAEBeans*

*лежит понятие задачи инженерного моделирования, базирующееся на по-*

*нятиях параметра задачи инженерного моделирования, полного дескрипто-*

*ра задачи и логического плана решения задачи. Архитектура CAEBeans оп-*

*ределяет четыре слоя РаВИС: концептуальный, логический, физический и*

*системный. Данным слоям соответствуют четыре типа оболочек CAEBeans.*

*Разработка и исполнение РаВИС поддерживается программным комплек-*

*сом «система CAEBeans». Выделено три основных роли пользователей, ра-*

*ботающих с РаВИС: инженер, прикладной программист, системный про-*

*граммист. Произведена оценка показателя среднего времени ответа для ре-*

*шения задачи инженерного моделирования средствами суперкомпьютер-*

*ных ресурсов Южно-Уральского государственного университета.*

*64*

***Г ЛА ВА3. С ИС ТЕ МАCAEBEANS***

*Для поддержки технологии CAEBeans, описанной в главе 2, был разра-*

*ботан программный комплекс с ис те маCAEBeans, обеспечивающий разра-*

*ботку и исполнение РаВИС.*

***3.1. С тр ук ту рас ис те мыCAEBeans***

***3.1.1. С ос та в с ис те мыCAEBeans***

*В систему CAEBeans [11] входят следующие компоненты (см.*

*рис. 12):*

*1. CAEBeans Constructor –интегрированная среда разработки распреде-*

*ленных виртуальных испытательных стендов для грид;*

*2. CAEBeans Portal –веб-приложение, обеспечивающее выбор, запуск и*

*получение результатов моделирования РаВИС;*

*3. CAEBeans Server –хранилище и среда исполнения РаВИС;*

*4. CAE-ресурсы –грид-сервисы, обеспечивающие удаленную постанов-*

*ку и решение задач инженерного моделирования;*

*5. CAEBeans Broker –автоматизированная система регистрации, анали-*

*за и предоставления CAE-ресурсов.*

***3.1.2. CAE-п ро ек т***

*В основе системы CAEBeans лежит понятие CAE-проекта.*

*CAEProject –это класс, объединяющий в себе взаимосвязанные сущно-*

*сти проблемных, потоковых и компонентных оболочек CAEBeans, ориен-*

*тированные на решение конкретного класса задач инженерного моделиро-*

*вания (см. рис. 13).*

*Класс CAEProject обладает следующими атрибутами:*

*65*

***Р ис 12.*** *Общая схема взаимодействия компонентов системы CAEBeans.*

***Р ис 13.*** *Сущности, составляющие CAE-проект.*

* GUID projectId: уникальный идентификатор [83] CAE-проекта,*

*обеспечивающий однозначную идентификацию [89] всех проектов*

*рамках системы CAEBeans;*

* String name: имя CAE-проекта, заданное прикладным програм-*

*мистом в процессе разработки.*

*В класс CAEProject входят экземпляры следующих классов:*

* ProblemCAEBeansTree: контейнер, содержащий дерево про-*

*блемных оболочек CAEBeans входящих в текущий CAE-проект;*

* WorkflowCAEBean: потоковый CAEBean.*

*Хотя класс CAEProject не реализует методов, обеспечивающих*

*решение задачи инженерного моделирования, он обеспечивает взаимодей-*

*ствие между входящими в него компонентами и объединяет их в рамках*

*единого CAE-проекта.*

*66*

***3.1.3. CAE-п ар ам ет р***

*CAEParameter –это класс, содержащий информацию об отдельном*

*параметре задачи инженерного моделирования. Можно выделить следую-*

*щие основные атрибуты класса CAEParameter:*

* String name: имя параметра, является его уникальным идентифи-*

*катором и не может дублироваться в рамках одного CAE-проекта;*

* CAEParameterType parameterType: тип CAE-параметра (цело-*

*численный, число с плавающей запятой, множество, строка и др.);*

* String default: значение параметра, заданное по умолчанию;*

* String units: единицы измерения параметра;*

* String comment: комментарии, описывающие особенности и об-*

*ласть применения CAE-параметра в терминах проблемной области*

*CAE-проекта;*

* String value: значение параметра (поддерживается использова-*

*ние стандартных арифметических операций, использующих в качест-*

*ве параметров имена других параметров задачи);*

* Bool visible: видимость параметра пользователю.*

*Унифицированное описание CAE-параметра позволяет автоматиче-*

*ски формировать пользовательский интерфейс для постановки любой зада-*

*чи инженерного моделирования на основе множества проблемных CAE-*

*параметров, значения которых должен указать инженер.*

***3.1.4. П ро бл ем ны й CAEBean***

*Класс ProblemCAEBean, реализующий сущность проблемного*

*CAEBean, объединяет CAE-параметры задачи инженерного моделирования*

*и формирует полный дескриптор задачи, на основе которого осуществляет-*

*ся постановка и решение задачи.*

*67*

*Можно выделить следующие основные атрибуты класса*

*ProblemCAEBean:*

* GUID problemCAEBeanId: идентификатор проблемного*

*CAEBean;*

* CAEProject\* parentProject: указатель на базовый CAE-*

*проект, в который входит текущий проблемный CAEBean;*

* String author: имя разработчика проблемного CAEBean;*

* String version: версия проблемного CAEBean;*

* CAEParamterCategory[] categories: контейнер, содержащий*

*в себе описание категорий, на которые разбиты параметры текущего*

*проблемного CAEBean;*

* CAEParameter[] parameters: множество параметров, объеди-*

*ненных данным проблемным CAEBean, включающее в себя парамет-*

*ры как с предопределенными значениями, так и со значениями опре-*

*деляемыми пользователем при постановке задачи.*

*Можно выделить следующий основной метод ProblemCAEBean:*

* void execute(ParameterValues[] userDefinedValues): вы-*

*зов данного метода инициирует процесс решения задачи инженерно-*

*го моделирования:*

*1) производится вычисление значений всех входных параметров*

*задачи инженерного моделирования и формирование полного*

*дескриптора задачи;*

*2) полный дескриптор задачи передается потоковому CAEBean*

*как параметр при вызове его метода execute().*

*В хо дн ыеп ар ам ет ры множество значений CAE-параметров, указан-*

*ных инженером при постановке задачи инженерного моделирования.*

*68*

***3.1.5. П от ок ов ыйCAEBean***

*Класс WorkflowCAEBean, реализующий сущность потокового*

*CAEBean, содержит информацию о логическом плане решения задачи ин-*

*женерного моделирования и обеспечивает его исполнение. Типы и семан-*

*тика узлов логического плана решения задачи основаны на нотации диа-*

*граммы деятельности стандарта UML 2.0.*

*Класс WorkflowCAEBean содержит следующие атрибуты:*

* ProblemDescriptor mainProblemDescriptor: полный де-*

*скриптор задачи, содержащий все CAE-параметры, необходимые для*

*решения задачи;*

* Workflow workflow: граф деятельности.*

*Интерфейс WorkflowCAEBean определяет методы, обеспечиваю-*

*щую исполнение логического плана решения задачи:*

* void execute(ProblemDescriptor problemDescriptor):*

*запуск исполнения логического плана решения задачи.*

*В хо дн ыеп ар ам ет ры полный дескриптор поставленной задачи.*

* void monitorLoop(): процедура исполнения м он ит ор а л ог ич е-*

*с ко гоп ла на.*

*Перед тем как описать алгоритм монитора логического плана, необ-*

*ходимо рассмотреть основные методы, предоставляемые узлами логическо-*

*го плана.*

*Граф деятельности, реализуемый классом Workflow, представляет*

*собой связный список, элементы которого представляют собой узлы логи-*

*ческого плана а связи между ними устанавливают очередность исполнения*

*данных узлов в процессе решения задачи инженерного моделирования. Уз-*

*лы логического плана потокового CAEBean, являются наследниками абст-*

*рактного класса AbstractWorkflowNode (см. рис. 14), что обеспечивает*

*69*

***Р ис 14.*** *Диаграмма классов, реализующих виды узлов логического плана.*

*универсальный механизм исполнения любого узла логического плана, не-*

*зависимо от его типа.*

*Рассмотрим атрибуты класса AbstractWorkflowNode:*

* AbstractWorkflowNode[] sourceNodes: набор узлов логиче-*

*ского плана, из которых выходят потоки управления, входящие в те-*

*кущий узел;*

* AbstractWorkflowNode[] targetNodes: набор узлов логиче-*

*ского плана, в которые ведут исходящие потоки управления из теку-*

*щего узла;*

* ProblemDescriptor\* descriptor: указатель на структуру, в*

*которой хранится полный дескриптор решаемой задачи инженерного*

*моделирования;*

*AbstractWorkflowNode определяет следующие виртуальные ме-*

*тоды:*

* Boolean ready(): проверка, готов ли узел к исполнению;*

* void sendToken(sourceNode\* AbstractWorkflowNode):*

*получить маркер управления от узла sourceNode;*

*70*

*// п ок а п ро ис хо ди т и сп ол не ни е л ог ич ес ко гоп ла на* ***while*** *(running) {*

*// ц ик л п о в се м у зл амл ог ич ес ко гоп ла на* ***foreach*** *(AbstractWorkflowNode node in workflow) {*

***if*** *node.ready(){ //Е сл и у зе л г от овки сп ол не ни ю*

*node.reset(); //С бр оси нф ор ма ци и оп ос ту пи вш их //м ар ке ра х у пр ав ле ни я*

*new Thread(node).start(); //И сп ол не ни е у зл а*

*} //if //во тд ел ьн омп от ок е*

*}//foreach*

*}//while*

***Р ис 15.*** *Алгоритм работы монитора потокового CAEBean****.***

* void run(): исполнить текущий узел логического плана;*

* void reset(): сбросить информацию о всех полученных маркерах*

*управления.*

*На рис. 15 приведен алгоритм работы монитора логического плана,*

*реализованный в методе WorkflowCAEBean.monitorLoop().*

*Процесс исполнения логического плана поддерживается посредством*

*монитора логического плана. Он постоянно производит проверку состояния*

*всех узлов логического плана. Готовность узла к исполнению зависит от*

*наличия маркеров управления в потоках управления, входящих в узел. В*

*зависимости от типа узлов, может требоваться наличие одного или не-*

*скольких маркеров управления на входных потоках управления. В связи с*

*этим, проверка готовности узла к исполнению инкапсулируется методом*

*ready().*

*Как только появляется узел, готовый к исполнению, монитор форми-*

*рует отдельную нить, в которой запускает исполнение метода run() дан-*

*ного узла. При завершении исполнения данного метода узел логического*

*плана обязан сформировать маркеры управления на потоках управления,*

*выходящих из него.*

*В зависимости от подхода к обработке поступающих маркеров*

*управления и формирования выходных маркеров, выделены три класса уз-*

*71*

*лов логического плана: DecisionMergeNode, InitialNode и абст-*

*рактный класс ParallelNode.*

*Класс InitialNode реализует начальный узел логического плана*

*решения задачи. Исполнение данного узла инициируется один раз за все*

*время исполнения логического плана решения задачи.*

*Единственный атрибут данного класса Boolean executed являет-*

*\_\_\_\_\_\_\_ся флагом, отмечающим то, что данный узел уже был исполнен.*

*Метод sendToken() является заглушкой виртуального метода,*

*описанного в родительском классе AbstractWorkflowNode, и не несет*

*никакой функциональной нагрузки, в связи с отсутствием входных потоков*

*в начальный узел.*

*Рассмотрим остальные методы, реализуемые данным классом:*

* void run(): вызывает метод putTokens();*

* Boolean ready(): возвращает значение флага executed;*

* void reset(): устанавливает значение флага executed равным*

*«и ст ин а».*

* void putTokens(): устанавливает маркеры деятельности во все*

*потоки деятельности исходящие из текущего узла.*

*Класс DecisionMergeNode объединяет в себе семантику узлов*

*разветвления и объединения (особенности узлов разветвления и объедине-*

*ния описаны в п. 2.2.2). Это позволяет формировать единый узел разветв-*

*ления-объединения, имеющий несколько входных и выходных потоков*

*управления. В состав класса входят следующие атрибуты:*

* String[] guardConditions: охранные условия, количество ко-*

*торых соответствует количеству потоков управления targetNodes,*

*исходящих из текущего узла;*

*72*

* String criteria: выражение, значение которого сравнивается с*

*охранными условиями для выбора дальнейшего пути исполнения ло-*

*гического плана;*

* CAEParameter[] inCAEParameters: множество CAE-*

*параметров участвующих в вычислении criteria;*

* Boolean token: флаг, отмечающий, что данному узлу пришел*

*маркер управления.*

*Класс DecisionMergeNode реализует абстрактные методы*

*sendToken(), run(), ready() и reset() определенные в роди-*

*тельском классе AbstractWorkflowNode, а также собственный метод*

*putToken():*

* void sendToken(): определить значение флага token равным*

*«и ст ин а»;*

* void putToken(AbstractWorkflowNode targetNode): пе-*

*редать маркер управления последующему узлу логического плана*

*targetNode;*

* void reset(): установить значение флага token равным «л ож ь»;*

* Boolean ready(): возвращает значение флага token.*

* void run(): исполнить текущий узел логического плана;*

*Метод DecisionMergeNode.run() обеспечивает интерпретацию*

*выражения, содержащегося в атрибуте criteria. Результат вычисления*

*выражения сравнивается с граничными условиями guardConditions.*

*После сравнения маркер управления передается по тому потоку управле-*

*ния, охранные условия которого удовлетворяют значению вычисленного*

*выражения. Если значение атрибута criteria не определено, считается,*

*что объект класса DecisionMergeNode реализует исключительно узел*

*слияния, и значения аргументов guardConditions, criteria и*

*73*

*inCAEParameters не учитываются при его исполнении. В этом случае*

*маркер управления передается единственному последующему узлу логиче-*

*ского плана.*

*Абстрактный класс ParallelNode обеспечивает возможность син-*

*хронизации потоков управления на входе и формирования множества пото-*

*ков управления на выходе узла логического плана.*

*Атрибут NodesDictionary tokens формирует словарь, веду-*

*щий учет входящих маркеров управления. В tokens каждому входящему*

*потоку управления поставлен в соответствие элемент типа Boolean, зна-*

*чение которого показывает, был ли получен по данному потоку маркер*

*управления. При инициации узла, словарь tokens заполняется значения-*

*ми «л ож ь».*

*В данном классе реализуются абстрактные методы sendToken(),*

*ready(), reset(), описанные в классе AbstractWorkflowNode, а*

*также метод putTokens():*

* void sendToken(): получение входного маркера управления от*

*предшествующего узла логического плана. При вызове данного мето-*

*да, текущий узел получает указатель на узел-источник*

*sourceNode\* AbstractWorkflowNode из параметра метода.*

*Элементу словаря tokens, соответствующему данному узлу, при-*

*сваивается значение «и ст ин а».*

* Boolean ready(): возвращает «и ст ин а», если значение всех*

*элементов в словаре tokens равно «и ст ин а». Иначе возвращает*

*значение «л ож ь».*

* void reset(): установить значения всех элементов словаря*

*tokens равными «л ож ь».*

*74*

*Класс FinalNode описывает конечный узел логического плана. Ис-*

*полнение метода run() этого узла инициирует процесс завершения реше-*

*ния задачи.*

*Класс ForkJoinNode объединяет в себе семантику узлов разделе-*

*ния и слияния. Такой подход позволяет формировать единый узел слияния-*

*разделения, поддерживаемый стандартом UML 2.0. Метод run() данного*

*класса вызывает метод putTokens().*

*Класс ActionNode является реализацией узла действия. Основная*

*задача данного класса –подготовить входные параметры для компонентно-*

*го CAEBean, содержащегося в атрибуте action, и экспортировать резуль-*

*таты его работы обратно в полный дескриптор задачи. Информация о тре-*

*буемых параметрах доступна непосредственно из атрибута Component-*

*CAEBean action. Метод ActionNode.run() вызывает метод*

*action.execute(ParameterValues[] inParameterValues).*

*ActionNode передает значения входных параметров компонентно-*

*му CAEBean и инициирует исполнение действия. После исполнения дейст-*

*вия, ActionNode записывает значения выходных параметров в полный де-*

*скриптор задачи, ссылка на который хранится в атрибуте descriptor, и*

*вызывает метод putTokens().*

***3.1.6. К ом по не нт ны й CAEBean***

*Класс ComponentCAEBean реализует функциональные возможно-*

*сти, предоставляемые компонентным CAEBean. Можно выделить следую-*

*щие основные атрибуты класса ComponentCAEBean:*

* String name: уникальное имя компонентного CAEBean, однознач-*

*но определяющее его в рамках текущего CAE-проекта.*

*75*

* SystemCAEBeanInterface\* systemCAEBeanInterface:*

*указатель на описание интерфейса системного CAEBean, средствами*

*которого реализуется действие компонентного CAEBean;*

* CAEParameter[] inCAEParameters: набор CAE-параметров,*

*значения которых требуются для постановки и реализации действия*

*компонентного CAEBean.*

* CAEParameter[] outCAEParameters: набор CAE-параметров,*

*значения которых устанавливаются в результате реализации текуще-*

*го действия.*

*Класс ComponentCAEBean поддерживает следующие методы:*

* void generateActionDefinition(ParameterValues[]*

*inParameterValues): формирование системно-ориентированной*

*постановки действия в формате, соответствующем интерфейсу базо-*

*вого системного CAEBean systemCAEBeanInterface;*

*в хо дн ыеп ар ам ет ры массив значений входных параметров;*

* URI findCAEResource(): данный метод инкапсулирует процесс*

*взаимодействия с Брокером Ресурсов, обеспечивая формирование за-*

*проса, поиск и предоставление CAE-ресурса, соответствующего тре-*

*бованиям данного компонентного CAEBean;*

*р ез ул ьт ат адрес предоставляемого CAE-ресурса;*

* GUID setAction(URI CAEResource): обеспечивает постанов-*

*ку действия на выбранный CAE-ресурс, включая пересылку файлов*

*постановки задачи и запуск процесса решения в соответствии с ин-*

*терфейсом базового системного CAEBean;*

*в хо дн ыеп ар ам ет ры адрес CAE-ресурса;*

*р ез ул ьт ат идентификатор, однозначно определяющий контекст ис-*

*полняемого действия на удаленной вычислительной системе;*

*76*

* ActionStatus checkActionStatus(GUID action, URI*

*CAEResource): метод обеспечивает проверку состояния текущего*

*действия на удаленном CAE-ресурсе;*

*в хо дн ыеп ар ам ет ры action –идентификатор действия,*

*CAEResource - адрес CAE-ресурса;*

*pе зу ль та т: статус действия;*

* void getResults(GUID action, URI CAEResource): обес-*

*печивает получение результатов (промежуточных или конечных, в*

*зависимости от статуса действия) с удаленного CAE-ресурса;*

*в хо дн ыеп ар ам ет ры action –идентификатор действия,*

*CAEResource - адрес CAE-ресурса;*

* ParameterValues[] translateResults(): метод анализирует*

*результаты решения задачи и извлекает значения выходных парамет-*

*ров;*

*pе зу ль та т: массив значений выходных параметров компонентного*

*CAEBean;*

* ParameterValues[] execute(ParameterValues[] inParameterValues):*

*метод обеспечивает автоматизированную генерацию,*

*постановку и получение результатов действия;*

*в хо дн ыеп ар ам ет ры массив значений входных параметров;*

*р ез ул ьт ат массив значений выходных параметров.*

***3.1.7. И нт ер фе йсс ис те мн ог о CAEBean***

*Системный CAEBean - это грид-сервис, обеспечивающий предостав-*

*ление функциональных возможности конкретной инженерной системы в*

*грид-среде. Класс SystemCAEBeanInterface представляет интерфейс*

*системного CAEBean. Он включает в себя описание формата входных и*

*77*

*выходных файлов, а также параметры интерфейса командной строки для*

*запуска процесса исполнения.*

* String name: имя, однозначно идентифицирующее системный*

*CAEBean;*

* CLIParameter[] parameters: описание параметров командной*

*строки, необходимых для запуска процесса решения задачи инженер-*

*ного моделирования средствами данного системного CAEBean;*

* FileFormat[] inFiles: описание формата входных файлов, не-*

*обходимых для постановки задачи;*

* FileFormat[] outFiles: описание формата выходных файлов,*

*формирующихся в результате решения задачи.*

*Методы класса SystemCAEBeanInterface обеспечивают реали-*

*зацию системно-зависимых операций при постановке и реализации дейст-*

*вия инженерного моделирования, инкапсулируя процесс взаимодействия с*

*сервисом системного CAEBean:*

* GUID createAction(URI CAEResource): формирует контекст*

*исполнения для определенного действия на удаленном системном*

*CAEBean;*

*в хо дн ыеп ар ам ет ры адрес CAE-ресурса;*

*р ез ул ьт ат массив значений выходных параметров.*

* void putFiles(GUID action, URI CAEResource, File[]*

*inFiles): копирует файлы постановки задачи, необходимые для*

*решения задачи средствами данного системного CAEBean;*

*в хо дн ыеп ар ам ет ры action –идентификатор действия,*

*CAEResource - адрес CAE-ресурса, inFiles –массив файлов,*

*содержащих постановку задачи, в формате, соответствующем базо-*

*вой CAE-системе;*

*78*

* void executeCLICommand(GUID action, URI targetSystem,*

*CLIParameter[] parameters): передает на удаленный*

*системный CAEBean значения параметров командной строки и ини-*

*циирует процесс реализации действия инженерного моделирования*

*средствами инженерного пакета;*

*в хо дн ыеп ар ам ет ры action –идентификатор действия,*

*CAEResource - адрес CAE-ресурса, inFiles –массив файлов,*

*содержащих постановку задачи, в формате, соответствующем базо-*

*вой CAE-системе;*

* void getFiles(GUID action, URI CAEResource): осуще-*

*ствляет получение файлов с результатами решения с системного*

*CAEBean;*

*в хо дн ыеп ар ам ет ры action –идентификатор действия,*

*CAEResource - адрес CAE-ресурса;*

* ActionStatus checkActionStatus(GUID action, URI*

*CAEResource): метод обеспечивает проверку состояния текущего*

*действия;*

*в хо дн ыеп ар ам ет ры action –идентификатор действия,*

*CAEResource - адрес CAE-ресурса;*

* void destroyAction(GUID action, URI CAEResource):*

*остановка исполнения действия;*

*в хо дн ыеп ар ам ет ры action –идентификатор действия,*

*CAEResource - адрес CAE-ресурса.*

***3.1.8. CAE-з ад ан ие*** *Класс CAEJob реализует сущность CAE-з ад ан ия****.*** *CAE-задание со-*

*держит информацию об определенной инженерной задаче, поставленной*

*пользователем. Когда пользователь заполняет значения всех необходимых*

*79*

*параметров проблемного CAEBean и отправляет CAE-проект на решение,*

*система CAEBeans формирует CAE-задание, которое поддерживает весь*

*дальнейший процесс решения задачи инженерного моделирования. Можно*

*выделить следующие основные атрибуты класса CAEJob:*

* GUID jobId: уникальный идентификатор CAE-задания;*

* CAEProject project: CAE-проект, содержащий полную инфор-*

*мацию о поставленной задаче;*

* PorblemCAEBean\* problemBean: указатель на проблемный*

*CAEBean, содержащий информацию о значениях параметров постав-*

*ленного CAE-задания;*

* DateTime submitTime: время постановки CAE-задания;*

* DateTime destroyTime: время принудительной остановки ис-*

*полнения CAE-задания;*

* String status: текущий статус CAE-задания.*

*CAEJob поддерживает следующие методы:*

* GUID createJob(CAEProject\* project,*

*PorblemCAEBean\* problemBean, ParameterValues[]*

*userDefinedValues): метод обеспечивает формирование задания на*

*основе определенного проблемного CAEBean существующего CAE-*

*проекта с учетом значений входных параметров userDefinedValues*

*установленных пользователем; метод возвращает уникальный идентифика-*

*тор CAE-задания;*

*в хо дн ыеп ар ам ет ры project –указатель на базовый CAE-проект,*

*problemBean –указатель на базовый проблемный CAEBean,*

*userDefinedValues –значения входных параметров, указанные*

*инженером при постановке задачи;*

*р ез ул ьт ат уникальный идентификатор сформированного CAE-*

*задания;*

*80*

***Р ис 16.*** *Общая схема CAEBeans Constructor.*

* String updateStatus(): обновить информацию о статусе CAE-*

*задания;*

* void terminateJob(): остановить \_\_\_\_\_\_\_исполнение задания;*

* void fetchOutcome(): метод обеспечивает предоставление ре-*

*зультатов исполнения задания.*

***3.2. К он ст ру кт ор*** *CAEBeans Constructor –это интегрированная среда разработки Ра-*

*ВИС на основе CAE-проектов. CAEBeans Constructor предоставляет при-*

*кладному программисту пользовательский интерфейс для разработки обо-*

*лочек CAEBeans концептуального, логического и физического слоев. В со-*

*ответствии с этим, пользовательский интерфейс, обеспечивающий разра-*

*ботку CAE-проектов в среде CAEBeans Constructor, разделен на 3 секции,*

*обеспечивающих разработку проблемных, логических и физических оболо-*

*чек CAEBean соответственно (см. рис. 16).*

*Разработка CAE-проекта начинается с формирования концептуально-*

*го слоя и корневого проблемного CAEBean в дереве проблемных оболочек.*

*Пользовательский интерфейс концептуального уровня позволяет приклад-*

*ному программисту сформировать список категорий и указать свойства*

*CAE-параметров входящих в каждую из категорий (см. рис. 17).*

*81*

***Р ис 17.*** *Пользовательский интерфейс для редактирования*

*концептуального слоя CAE-проекта.*

*Также, CAEBeans Constructor предусматривает возможность форми-*

*рования иерархии проблемных CAEBean, посредством наследования до-*

*черних проблемных CAEBean от родительского. CAE-параметры в насле-*

*дуемом проблемном CAEBean можно зафиксировать определенными зна-*

*чениями (константой, либо выражением, значение которого зависит от дру-*

*гих CAE-параметров).*

*Пользовательский интерфейс логического слоя CAEBeans Constructor*

*обеспечивает визуальное редактирование графа логического плана потоко-*

*вого CAEBean (см. рис. 18). Разработчик РаВИС может создавать, редакти-*

*ровать и удалять узлы логического плана. В зависимости от типа добавляе-*

*мого узла, может потребоваться указать его дополнительные свойства: при*

*создании узла ветвления требуется указать условие ветвления, при созда-*

*нии узла ветвления управления требуется указать количество параллельных*

*ветвей. При создании нового узла деятельности, пользователю предостав-*

*ляется возможность выбора одного из существующих или создание нового*

*компонентного CAEBean.*

*82*

***Р ис 18.*** *Пользовательский интерфейс для редактирования*

*логического слоя CAE-проекта.*

*Для создания нового или редактирования существующего компо-*

*нентного CAEBean программисту предоставляется пользовательский ин-*

*терфейс физического слоя. Программисту необходимо выбрать базовый*

*системный CAEBean, интерфейсу которого будет соответствовать новый*

*компонентный CAEBean. Система CAEBeans Constructor поддерживает*

*возможность импорта описания новых системных CAEBeans, разработан-*

*ных системным программистом. Далее, разработчик указывает входные и*

*выходные параметры для созданного компонентного CAEBean, включая*

*параметры-файлы, обеспечивающие постановку задачи инженерного моде-*

*лирования и анализ полученных результатов.*

*Разработанный CAE-проект можно сохранить на компьютере при-*

*кладного программиста, или внедрить в систему CAEBeans, экспортировав*

*CAE-проект в CAEBeans Server, посредством метода putCAEProject().*

*83*

***Р ис 19.*** *Схема CAEBeans Portal.*

***3.3. К ли ен т***

*CAEBeans Portal –это веб-приложение, доступное через интернет,*

*обеспечивающее пользовательский интерфейс для постановки и решения*

*задач инженерного моделирования средствами системы CAEBeans.*

*В соответствии с функциональной нагрузкой, можно выделить три*

*части CAEBeans Portal: пользовательский веб-интерфейс, генератор веб-*

*форм и клиент CAEBeans Server (см. рис. 19).*

*При входе в систему, производится авторизация пользователя, после*

*чего ему предоставляется список РаВИС, доступных для моделирования, а*

*также список задач, запущенных пользователем ранее с указанием их те-*

*кущего статуса. Для решения новой задачи инженерного моделирования,*

*пользователь должен выбрать РаВИС, на основе которого будет произведе-*

*на постановка задачи. После выбора РаВИС, пользователю предлагается*

*интерфейс проблемного CAEBean, соответствующего выбранной задаче.*

*Генератор веб-форм обеспечивает хранение проблемных оболочек*

*CAEBeans, импортированных из CAEBeans Server и автоматическую гене-*

*рацию веб-форм для постановки CAE-заданий на основе описания CAE-*

*параметров соответствующих проблемных CAEBean.*

*Графический интерфейс постановки задачи инженерного моделиро-*

*вания соответствует и полностью определяется описанием параметров ин-*

*84*

*женерного моделирования базового проблемного CAEBean. Параметры*

*разделены на группы, соответствующие группам параметров проблемного*

*CAEBean. На основе информации о типе параметра инженерного модели-*

*рования, выбирается метод ввода параметра: выпадающий список, пере-*

*ключатель, или же поле ввода. Атрибуты класса CAEParameter units и*

*comment позволяют привести описание сущности указанного CAE-*

*параметра в пользовательском интерфейсе. Значение параметра по умолча-*

*нию выставляется в соответствии со значением атрибута default. Таким*

*образом, прикладному программисту не требуется дополнительно форми-*

*ровать интерфейс для инженера для каждого нового проблемного*

*CAEBean.*

*Клиент CAEBeans Server обеспечивает постановку и получение ре-*

*зультатов решения CAE-заданий от CAEBeans Server, а также импорт су-*

*ществующих проблемных оболочек CAEBeans для дальнейшей генерации*

*веб-форм для новых РаВИС.*

***3.4. С ер ве р***

*CAEBeans Server –это грид-сервис, обеспечивающий хранение и ин-*

*терпретацию CAE-проектов. Стуктурно, CAEBeans Server состоит из сле-*

*дующих подсистем (см. рис. 20):*

* и нт ер фе йспредоставляет возможности по взаимодействию с систе-*

*мой CAEBeans Server со стороны CAEBeans Portal и импорт CAE-*

*проектов из CAEBeans Constructor;*

* х ра ни ли щеобеспечивает хранение CAE-проектов, библиотеки сис-*

*темных CAEBean, промежуточных и конечных результатов решения*

*задач инженерного моделирования;*

* м ен ед же р CAE-з ад ачотвечает за создание, управление работой и*

*уничтожение CAE-заданий;*

*85*

***Р ис 20.*** *Структура CAEBeans Server.*

* и сп ол ни те льобеспечивает исполнение отдельного CAE-задания на*

*базе определенного CAE-проекта с уникальным набором значений*

*входных параметров;*

* к ли ен т CAEBeans Broker отвечает за взаимодействие с системой*

*CAEBeans Broker для поиска и предоставления CAE-ресурсов.*

*Можно выделить следующие основные методы, предоставляемые ин-*

*терфейсом CAEBeans Server:*

* GUID createJob(GUID projectId, GUID problemCAEBeanId,*

*ParameterValues[] userDefinedValues): создать*

*CAE-задание.*

*в хо дн ыеп ар ам ет ры projectId –идентификатор CAE-проекта,*

*problemCAEBeanId –идентификатор проблемного CAEBean,*

*userDefinedValues –значения входных параметров, указанные*

*инженером при постановке задачи;*

*р ез ул ьт ат уникальный идентификатор сформированного CAE-*

*задания;*

***CAEBeans Server***

***CAEBeans***

***Broker***

***CAEBeans***

***Portal***

***Х ра ни ли ще И сп ол ни те ль CAE-***

***Р ес ур с***

***CAEBeans***

***Constructor***

***М ен ед же р***

***CAE-з ад ач*** *86*

* CAEJobProperties getJobProperties(GUID jobId): полу-*

*чить свойства указанного CAE-задания;*

*в хо дн ыеп ар ам ет ры jobId –уникальный идентификатор CAE-*

*задания;*

*р ез ул ьт ат структура, содержащая свойства CAE-задания, такие, как*

*текущий статус, время постановки задачи, время окончания решения,*

*ссылки на файлы, содержащие текущие результаты решения;*

* void terminateJob(): остановить исполнение задания;*

* void putCAEProject(CAEProject project): импортировать*

*CAE-проект, разработанный в CAEBeans Constructor;*

* ProblemCAEBean getProblemCAEBean(GUID problemCAEBeanId):*

*экспортировать проблемный CAEBean в CAEBeans Portal*

*для формирования пользовательского интерфейса.*

*Функционально, CAEBeans Server представляет собой грид-сервис на*

*базе стандарта WSRF:*

* при решении задачи инженерного моделирования, каждый запрос к*

*CAEBeans Server происходит в контексте CAE-задания, уникальный*

*идентификатор которого обеспечивает формирование контекста ис-*

*полнения запроса по стандарту WS-Addressing;*

* свойствами ресурса, предоставляемые CAEBeans Server по стандарту*

*WS-ResouceProperties, являются свойства текущего CAE-задания;*

* обеспечивается подписка на базе стандарта WS-Notification об изме-*

*нении состояния свойств CAE-задания;*

* время жизни CAE-задания управляется на базе стандарта WSResourceLifetime.*

***3.5. CAE-р ес ур с***

*CAE-р ес ур со м является экземпляр системного CAEBean, предостав-*

*ляющий ресурсы некоторого инженерного пакета на базе конкретной целе-*

*87*

*вой системы. Интерфейс, предоставляемый CAE-ресурсом, определяется*

*классом SystemCAEBeanInterface, описание которого приведено в п.*

*3.1.7, и обеспечивает:*

* получение данных для решения задачи средствами базового инже-*

*нерного пакета из CAEBeans Server или внешнего источника данных;*

* запуск и автоматизированное решение задачи инженерного модели-*

*рования;*

* передачу результатов решения CAEBeans Server или во внешнее хра-*

*нилище данных.*

***Ц ел ев аяс ис те ма****–это совокупность грид-сервисов, которые имеют*

*доступ к пространству программных, аппаратных и лицензионных ресурсов*

*некоторого узла грид, и поддерживает аутентификацию и авторизацию*

*пользователей. Целевая система CAEBeans:*

* формируется на базе целевой системы UNICORE;*

* обеспечивает авторизацию и аутентификацию пользователей по про-*

*токолам безопасности грид;*

* является контейнером CAE-ресурсов и обеспечивает их динамиче-*

*скую конфигурацию;*

* обеспечивает взаимодействие грид-узла с CAEBeans Server (поста-*

*новка и получение результатов решения подзадач, оповещение об из-*

*менении статуса подзадач и др.);*

* обеспечивает взаимодействие грид-узла с CAEBeans Broker (предос-*

*тавление информации о текущем состоянии (статических и динами-*

*ческих характеристиках) целевой системы, резервирование и освобо-*

*ждение вычислительных ресурсов и др.).*

*Целевая система UNICORE предоставляет набор стандартных серви-*

*сов UNICORE Atomic Services [25, 103], поддерживающих управление за-*

*дачами и обмен данными:*

*88*

* TargetSystemFactory: обеспечивает создание новых сервисов целевых*

*систем;*

* TargetSystemService: сервис, предоставляющий интерфейс для поста-*

*новки задач на удаленную вычислительную систему;*

* JobManagementService: обеспечивает управление и мониторинг ис-*

*полнения задач на целевой системе;*

* StorageManagementService, FileTransferService: обеспечивают обмен*

*файлами постановки при постановке задачи и получении результатов*

*решения.*

*Системные CAEBeans реализуются в виде специальных сервисов це-*

*левой системы, обеспечивающих исполнение действий инженерного моде-*

*лирования. Комплект системных CAEBeans разрабатывается отдельно для*

*каждого пакета инженерного моделирования. Также, реализация системных*

*CAEBeans одного вида может различаться в зависимости от версии инже-*

*нерного пакета, или от базовой операционной системы.*

***3.6. Б ро ке р***

*CAEBeans Broker обеспечивает автоматизированную регистрацию,*

*поиск и выделение CAE-ресурсов для реализации действий инженерного*

*проектирования. В основе архитектуры CAEBeans Broker лежит понятие*

*CAE-ресурса, обеспечивающее виртуализацию физических ресурсов, дос-*

*тупных в грид. При решении задач инженерного проектирования в грид не-*

*обходимо учитывать значительные ограничения, возлагаемые на вычисли-*

*тельные ресурсы, реализующие отдельные действия. Они могут быть вы-*

*званы различными причинами: различного типа лицензионные ограниче-*

*ния; низкий уровень масштабируемости конкретного класса задач на су-*

*перкомпьютерные системы определенной архитектуры и др. В связи с этим,*

*становится неоптимальным предоставление ресурсов всей целевой системы*

*под нужды определенной задачи. Данная проблема решается посредством*

*89*

***Р ис 21.*** *Структура очередей CAEBeans Broker.*

*реализации виртуальных CAE-ресурсов, которые инкапсулируют опреде-*

*ленную часть физических ресурсов (вычислительных и лицензионных) от-*

*дельного узла грид.*

*Система CAEBeans Broker обеспечивает распределение действий ин-*

*женерного моделирования по виртуальным CAE-ресурсам. В структуре*

*системы CAEBeans Broker можно выделить следующие основные блоки*

*(см. рис. 21):*

* в хо дн аяо че ре дьз ад ан ий(inputQueue) обеспечивает получение от*

*сервера запросов на предоставление CAE-ресурсов;*

* р ас пр ед ел ит ел ь з ад ан ийобеспечивает анализ поступающих запросов*

*и их распределение по очередям заданий;*

* к ат ал огр ес ур со в хранит информацию о характеристиках всех физи-*

*ческих вычислительных узлов, входящих в грид, и виртуальных CAE-*

*ресурсах находящихся на данных узлах;*

*90*

* б ло к п ер ер ас пр ед ел ен ияр ес ур со в обеспечивает возможность удале-*

*ния старых и формирования новых CAE-ресурсов на основе доступ-*

*ных физических ресурсов;*

* о че ре дьо жи да ни я (waitQueue), в которую попадают задания, которые*

*не могут быть исполнены на существующем наборе CAE-ресурсов,*

*но можно использовать перераспределение для получения CAE-*

*ресурсов, удовлетворяющих требованиям.*

*Интерфейс CAEBeans Broker предоставляет следующие методы:*

* GUID resourceQuery(Query query): найти ресурс соответст-*

*вующий требованиям действия;*

*в хо дн ыеп ар ам ет ры запрос, содержащий требования к предостав-*

*ляемому ресурсу, такие как тип необходимого системного CAEBean,*

*базовая операционная система, максимальное время исполнения за-*

*дачи, требования к аппаратной платформе;*

*р ез ул ьт ат идентификатор задачи в брокере ресурсов;*

* void removeTask(GUID taskId): удалить задание из всех оче-*

*редей;*

*в хо дн ыеп ар ам ет ры уникальный идентификатор задания брокера;*

* void lockResource(GUID taskId, GUID CAEResource-*

*Id): закрепить CAE-ресурс для выполнения определенного действия;*

*в хо дн ыеп ар ам ет ры taskId - уникальный идентификатор задания*

*брокера; CAEResourceId - уникальный идентификатор CAE-*

*ресурса;*

* void unlockResource(GUID taskId, GUID CAEResource-*

*Id): освободить CAE-ресурс после исполнения действия;*

*91*

*в хо дн ыеп ар ам ет ры taskId - уникальный идентификатор задания*

*брокера; CAEResourceId - уникальный идентификатор CAE-*

*ресурса.*

*Рассмотрим а лг ор ит м п ре до ст ав ле ни я CAE-р ес ур садля реализации*

*действия.*

*1. При исполнении очередного действия, физический CAEBean форми-*

*рует запрос query на предоставление CAE-ресурса, и CAEBeans*

*Server передает его CAEBeans Broker при вызове метода*

*resourceQuery();*

*2. CAEBeans Broker анализирует доступные виртуальные и физические*

*ресурсы. Если существует возможность решить задачу существую-*

*щими CAE-ресурсами, он ставит задачу ко всем ресурсам, подходя-*

*щим по запросу и возвращает CAEBeans Server уникальный иденти-*

*фикатор задания брокера taskId. Если существует возможность пе-*

*рераспределить ресурсы для формирования нового CAE-ресурса, от-*

*вечающего требованиям запроса, CAEBeans Broker ставит задачу в*

*очередь waitQueue и возвращает CAEBeans Server уникальный иден-*

*тификатор задания taskId. Если нет физических ресурсов, обеспе-*

*чивающих решение данной задачи, CAEBeans Broker возвращает*

*ошибку «Н етр ес ур со в, с оо тв ет ст ву ющ ихз ап ро су».*

*3. Если задача занимает первое место в очереди к CAE-ресурсу,*

*CAEBeans Broker производит оповещение CAEBeans Server о том,*

*что существует возможность решить задачу на выделенном ресурсе,*

*указывая при этом место текущей задачи в очередях ко всем осталь-*

*ным ресурсам.*

*4. Если CAEBeans Server решает захватить требуемый ресурс, он вызы-*

*вает метод lockResource(), захватывая предложенный ресурс.*

*92*

*При этом задача автоматически снимается со всех очередей к альтер-*

*нативным CAE-ресурсам. Дальнейшая постановка и решение задачи*

*на захваченном ресурсе производится без участия CAEBeans Broker.*

*По окончании решения, CAEBeans Server вызывает метод*

*unlockResource(), сигнализируя CAEBeans Broker о том, что ре-*

*сурс освободился.*

*5. Во время работы CAEBeans Broker, периодически производится про-*

*цедура перераспределения CAE-ресурсов для обеспечения оптималь-*

*ного времени решения задач. На основе анализа задач в очереди*

*waitQueue может быть принято решение на формирование более*

*крупного CAE-ресурса. В этом случае:*

*a) производится поиск ресурсов, суммарная мощность которых*

*может обеспечить решение поставленной задачи;*

*b) они маркируются, после чего в очереди к ним не производится*

*постановка новых задач;*

*c) как только очереди ко всем помеченным ресурсам очищены от*

*задач, производится удаление этих CAE-ресурсов и формиро-*

*вание нового более крупного ресурса. К нему в очередь ставят-*

*ся соответствующие задачи из очереди waitQueue.*

***3.7. В за им од ей ст ви е к ом по не нт овс ис те мыCAEBeans***

*Рассмотрим, каким образом компоненты системы CAEBeans обеспе-*

*чивают процесс постановки и решения задачи инженерного моделирования*

*(см. рис. 22).*

*1. Инженеру предоставляется доступ к веб-страницам CAEBeans Portal.*

*Он авторизуется на данном портале и выбирает РаВИС. Посредством*

*пользовательского интерфейса, предоставляемого CAEBeans Portal,*

*инженер указывает значения CAE-параметров и интересующие его*

*результаты решения, после чего запускает процесс решения задачи.*

*93*

***Р ис 22.*** *Диаграмма последовательности постановки и решения задачи инженерно-*

*го моделирования в системе CAEBeans.*

*2. CAEBeans Portal формирует проблемный CAEBean, устанавливая*

*значения всех CAE-параметров, введенные пользователем, и отправ-*

*ляет запрос CAEBeans Server на создание нового CAE-задания с соот-*

*ветствующими входными значениями. В ответ на данный запрос,*

*CAEBeans Server формирует новый контекст исполнения задачи, и*

*передает уникальный идентификатор CAE-задания (в терминах*

*WSRF –идентификатор ресурса) системе CAEBeans Portal. В даль-*

*нейшем, любую информацию о ходе решения задачи можно полу-*

*чить, указав данный идентификатор CAE-задания в запросе*

*CAEBeans Server.*

*94*

*3. В CAE-задание загружаются данные о соответствующем CAE-*

*проекте, включая дерево проблемных оболочек CAEBeans, и произ-*

*водится инициализация всех CAE-параметров задачи. После этого*

*начинается последовательное исполнение логического плана решения*

*задачи. Дальнейшие шаги последовательно повторяются для каждого*

*узла деятельности в составе логического плана вплоть до окончания*

*решения задачи:*

*3.1. Компоненту CAEBeans Broker отправляется запрос на выбор*

*оптимального вычислительного ресурса. Он производит анализ*

*состояния доступных узлов грид и соответствующих целевых*

*систем, после чего резервирует и передает адрес наиболее под-*

*ходящего CAE-ресурса системе CAEBeans Server.*

*3.2. CAEBeans Server обновляет статус задачи и оповещает об этом*

*CAEBeans Portal.*

*3.3. CAEBeans Server производит загрузку начальных данных на*

*выделенный CAE-ресурс, после чего, инициирует процесс ре-*

*шения подзадачи.*

*3.4. По окончании решения подзадачи, CAE-ресурс производит вы-*

*грузку результатов решения на выделенный сервер хранения*

*результатов (в зависимости от задачи это может быть CAEBeans*

*Server; узел грид, ответственный за следующий этап ре-*

*шения CAE-задания или же какой-либо другой узел грид).*

*3.5. CAEBeans Server отправляет запрос CAEBeans Broker на осво-*

*бождение текущего CAE-ресурса.*

*4. CAEBeans Server обновляет статус задачи на «Успешно решена» и*

*останавливает исполнение CAE-задания.*

*95*

*5. Инженер заходит на CAEBeans Portal и выбирает данные, которые*

*хочет загрузить на локальный компьютер. CAEBeans Portal обращает-*

*ся к CAEBeans Server и получает ссылки на запрошенные данные, по-*

*сле чего, производится загрузка результатов решения пользователю.*

***3.8. В ыв од ы п о г ла ве3***

*В рамках диссертационного исследования был разработан программ-*

*ный комплекс система CAEBeans, обеспечивающий поддержку всех этапов*

*разработки и исполнения РаВИС. Была разработана архитектура программ-*

*ного комплекса, описаны основные интерфейсы, предоставляемые компо-*

*нентами системы CAEBeans. Также, были исследованы процедуры взаимо-*

*действия между компонентами системы CAEBeans в процессе постановки и*

*решения задачи инженерного моделирования средствами РаВИС.*

*96*

***Г ЛА ВА4. И СП ЫТ АН ИЯС ИС ТЕ МЫCAEBEANS***

*Для проверки возможностей, предоставляемых системой CAEBeans в*

*области создания и исполнения виртуальных испытательных стендов, был*

*разработан набор РаВИС, обеспечивающих решение различных задач ин-*

*женерного моделирования средствами различных CAE-пакетов. Были про-*

*изведены исследования методов взаимодействия с современными инженер-*

*ными пакетами, и проанализированы API, предоставляющие методы авто-*

*матизированного решения задач инженерного моделирования. Для испыта-*

*ния системы CAEBeans были разработаны оболочки для решения задач*

*средствами наиболее распространенных пакетов инженерного моделирова-*

*ния: ANSYS CFX [15], ANSYS Mechanical [8], ABAQUS [10],*

*DEFORM [6, 7].*

*В качестве основной испытательной задачи была выбрана задача мо-*

*делирования процесса закалки и охлаждения труб и анализа влияния раз-*

*личных аспектов процесса закалки на качество производимой продукции*

*(«Распределенный виртуальный испытательный стенд «Термообработка»),*

*решаемая по заказу ОАО «Челябинский трубопрокатный завод».*

***4.1. И сп ыт ан иес ис те мыCAEBeans н а б аз е DEFORM***

*Основные испытания системы CAEBeans производились на базе вир-*

*туального испытательного стенда, моделирующего процесс овализации*

*труб при их закалке и последующем отпуске на ОАО «Челябинский трубо-*

*прокатный завод» [6]. Процесс изготовления цельнокатаных труб является*

*технологически сложным и недетерминированным по такому параметру,*

*как разнотолщинность стенок изготавливаемой трубы. После изготовления*

*труба поступает в цех термической обработки, где производится ее закалка*

*и отпуск на индукционной установке. При разогреве трубы и последующем*

*охлаждении в ряде случаев возникает эффект овализации концов трубы.*

*97*

*Трубы с подобным дефектом технически невозможно сваривать в трубо-*

*провод. Челябинский трубопрокатный завод использует прессовые уста-*

*новки, для устранения брака продукции. Однако подобный способ решения*

*проблемы брака является устаревшим и экономически нецелесообразным,*

*так как приводит к большим финансовым потерям [6]. В соответствии с*

*этим актуальной является задача создания виртуального испытательного*

*стенда как средства проведения вычислительных экспериментов, с целью*

*поиска оптимального решения по изменению существующей технологии*

*производства труб.*

*Перед созданием компьютерной модели процесса термической обра-*

*ботки были произведены тепловизионные исследования процесса закалки*

*труб непосредственно на производстве. Был произведен сбор информации*

*о геометрии заготовок и температурных полях, величине разностенности с*

*разверткой по длине и окружности заготовки, величине начальной овально-*

*сти с разверткой по длине и кривизне оси заготовки.*

*Для создания компьютерной модели процесса термической обработки*

*труб на индукционных установках был выбран инженерный пакет*

*DEFORM. В рамках поставленной задачи пакет DEFORM имеет ряд пре-*

*имуществ, так как предназначен для анализа процессов обработки металлов*

*давлением, термической и механической обработки. Кроме того, в пакете*

*DEFORM есть модуль Microstructure 3D, который предназначен для моде-*

*лирования трехмерных процессов термической обработки и позволяет мо-*

*делировать микроструктурные превращения в заготовке при ее деформиро-*

*вании.*

*Пакет DEFORM предоставляет возможность производить постановку*

*задачи, ее решение и обработку результатов без запуска графического ин-*

*терфейса, путем использования пакетного режима препроцессора, решателя*

*и постпроцессора [7]. Пакетный режим препроцессора позволяет произво-*

*дить весь цикл постановки задачи, на основе текстового \*.key файла с ее*

*98*

***Р ис 23.*** *Интерфейс постановки задачи инженерного моделирования предостав-*

*ляемый CAEBeans Portal.*

*описанием. Решатель позволяет произвести запуск решения задачи в пакет-*

*ном режиме, указав имя базы данных задачи. Постпроцессор в пакетном*

*режиме позволяет сохранять изображения модели на разных шагах расчета,*

*с выбором требуемых данных для отображения: температура, фазовый со-*

*став, деформации и т.д.*

*При разработке проблемного CAEBean была предусмотрена возмож-*

*ность изменения следующих технических параметров индукционной уста-*

*новки: количество индукторов, частота и сила тока, длина индукторов, ко-*

*личество и конфигурация водяных струй, давление и расход воды, скорость*

*движения трубы через индукционную установку, частота вращения труб*

*(см. рис. 23). Также, была предусмотрена возможность моделирования тер-*

*мообработки труб из различных марок сталей (путем указания физических*

*характеристик материала).*

*В результате моделирования из файла базы данных пакета DEFORM*

*сохраняются все необходимые результаты: данные о распределении темпе-*

*ратур в трубе, данные о фазовом составе вещества трубы, данные о дефор-*

*мациях трубы, графики смещения выбранной точки трубы (см. рис. 24).*

*99*

***Р ис 24.*** *Пример результата моделирования задачи овализации трубы при закалке.*

*Результаты предоставляются инженеру в виде изображений формата PNG и*

*табличных данных.*

*Система CAEBeans была развернута на базе суперкомпьютерных ре-*

*сурсов СКЦ ЮУрГУ [18]. Системы CAEBeans Portal, CAEBeans Server и*

*CAEBeans Broker были установлены на вычислительных узлах суперком-*

*пьютера Infinity. В системе CAEBeans Constructor был разработан CAE-*

*проект РаВИС. В CAEBeans Portal был загружен проблемный CAEBean*

*созданного виртуального испытательного стенда. На CAEBeans Portal была*

*создана тестовая учетная запись и для нее предоставлены права на разрабо-*

*танный виртуальный испытательный стенд.*

*На узлах суперкомпьютера СКИФ Урал была установлена и настрое-*

*на грид-система Unicore [25]. В Unicore была создана целевая система, реа-*

*лизующая системный CAEBean для взаимодействия с пакетом DEFORM.*

*На вычислительных узлах суперкомпьютера СКИФ Урал был установлен*

*пакет DEFORM.*

*100*

*Доступ инженера к CAEBeans Portal осуществляется посредством ин-*

*тернет-обозревателя. После авторизации менеджер задач отображает спи-*

*сок доступных испытательных стендов и список запущенных расчетов. В*

*менеджере задач можно создать и запустить новую задачу моделирования*

*индукционного нагрева. По окончанию расчетов изображения с результа-*

*тами моделирования доступны в разделе результатов задачи.*

***4.2. И сп ыт ан иес ис те мыCAEBeans н а б аз е ANSYS Mechanical***

*Для тестирования возможности формирования РаВИС на базе пакета*

*ANSYS Mechanical был разработан виртуальный испытательный стенд для*

*моделирования резьбового соединения обсадных и насосно-компрессорных*

*труб для нефтяных скважин [8].*

*Обсадные трубы используются для защиты оборудования, опущенно-*

*го в скважину, от внешних воздействий, таких как загрязнение, механиче-*

*ское воздействие. Обсадные трубы соединяются с помощью муфт. При уг-*

*лублении скважины на обсадную колонну навинчивают очередную трубу и*

*опускают глубже. Данный вид труб свинчивается однократно и подвергает-*

*ся изгибающим и растягивающим нагрузкам во время эксплуатации.*

*Насосно-компрессорные трубы имеют диаметр от 80 до 140 мм, рас-*

*полагаются внутри обсадных и используются для подачи напора воды и*

*получения нефти. Эти трубы подвергаются многократному свинчиванию и*

*развинчиванию, что приводит к сильному износу резьбы и невозможности*

*эксплуатации труб.*

*Поэтому, одной из задач, возникающих на практике в нефтяной про-*

*мышленности, является задача построения параметризованной модели*

*резьбового соединения двух труб с помощью муфты. Модель предназначе-*

*на для разработки новых видов резьбовых соединений для обсадных и на-*

*сосно-компрессорных труб, которые могли бы обеспечить более совершен-*

*ные характеристики соединения.*

*101*

***Р ис 25.*** *Пример результатов моделирования резьбы обсадной трубы.*

*Моделирование резьбового соединения было произведено в пакете*

*ANSYS Mechanical. Для создания системных CAEBeans, использовался ре-*

*жим пакетной обработки, поддерживаемый ANSYS Mechanical. В качестве*

*входных параметров постановки задачи используется log-файл. Log-файл*

*представляет собой набор команд, выполняя которые пошагово в консоль-*

*ной строке ANSYS Mechanical, можно в воспроизвести действия, произво-*

*димые в GUI за время работы с ANSYS Mechanical.*

*В разработанном проблемном CAEBean выделены три группы про-*

*блемных параметров:*

*1. параметры трубы (такие как наружный диаметр, толщина стенки,*

*длина конусной части);*

*2. параметры резьбы (шаг резьбы, высота профиля);*

*3. нагрузка (растяжение, свинчивание, коэффициент трения).*

*Результаты моделирования предоставляются пользователю в графиче-*

*ском (см. рис. 25) и табличном виде.*

***4.3. И сп ыт ан иес ис те мыCAEBeans н а б аз е Abaqus***

*Для испытания системы CAEBeans на базе конечно-элементного CAE-*

*пакета ABAQUS, был разработан РаВИС для моделирования напряженно-*

*102*

***Р ис 26.*** *Пример результатов моделирования грунтового массива.*

*деформированного состояния грунтового массива под массивным зданием*

*или сооружением [10].*

*Взаимодействие с САЕ-пакетом ABAQUS было реализовано посредст-*

*вом Интерфейса Сценариев ABAQUS. Интерфейс Сценариев ABAQUS яв-*

*ляется расширением объектно-ориентированного языка Python [101], и*

*обеспечивает взаимодействие с функциональными возможностями, предос-*

*тавляемыми компонентами, входящими в пакет ABAQUS. Интерфейс Сце-*

*нариев ABAQUS обеспечивает:*

* создание и модификацию компонентов модели ABAQUS, таких как*

*детали, материалы, нагрузки, шаги расчета;*

* создание, изменение и запуск вычислительного процесса в ABAQUS;*

* чтение и запись базы данных модели;*

* просмотр и анализ результатов расчета.*

*Проблемный CAEBean обеспечивает изменение таких параметров мо-*

*дели как размеры грунтового массива, нагрузка, размеры нагружаемой об-*

*ласти. Результаты моделирования предоставляются пользователю в графи-*

*ческом виде (см. рис. 26).*

*103*

***4.4. И сп ыт ан иес ис те мыCAEBeans н а б аз е ANSYS CFX***

*Для испытания взаимодействия системы CAEBeans и комплекса*

*ANSYS CFX был разработан тестовый РаВИС, обеспечивающий моделиро-*

*вание обдувания дымовой трубы.*

*В ходе анализа пакета ANSYS CFX, было выявлено несколько спосо-*

*бов взаимодействия внешних систем с компонентами, составляющими дан-*

*ный пакет. Наиболее приемлемый способ автоматизации взаимодействия с*

*компонентами пакета ANSYS CFX –это запуск и исполнение его компо-*

*нентов в командном режиме [15].*

*Все подсистемы пакета ANSYS CFX (ANSYS CFX-Pre, ANSYS CFXSolver,*

*ANSYS CFX-Post) поддерживают работу в пакетном режиме по-*

*средством указания специального флага в командной строке. Также, каж-*

*дый компонент пакета ANSYS CFX поддерживает возможность автомати-*

*зированной постановки действий инженерного моделирования посредством*

*специальных форматов импортируемых файлов:*

* файлы сессии ANSYS CFX-Pre (\*.pre);*

* файл CCL («Command Language File»), обеспечивающий отображение*

*параметров задачи инженерного моделирования в текстовом форма-*

*те;*

* файлы сессии ANSYS CFX-Post (\*.cse).*

*Таким образом, указав необходимые параметры в командной строке*

*или входных файлах, можно поставить требуемое действие инженерного*

*моделирования компоненту пакета ANSYS CFX и получить результаты ее*

*решения.*

*Проблемный CAEBean обеспечивает изменять скорость и температу-*

*ру воздушных потоков (как ветра, так и потока, исходящего из трубы). Ре-*

*зультаты моделирования предоставляются пользователю в графическом*

*виде (см. рис. 27).*

*104*

***Р ис 27.*** *Пример результатов моделирования*

*обтекания трубы воздушным потоком.*

***4.5. В ыв од ы п о г ла ве4***

*Для тестирования технологии использования компонентных*

*CAEBeans были созданы оболочки для решения следующих задач инже-*

*нерного моделирования:*

*1) моделирование резьбовых соединений труб для нефтяных скважин*

*(на базе инженерного пакета ANSYS Mechanical);*

*2) моделирование эффекта овализации труб при термической обработке*

*(на базе инженерного пакета DEFORM);*

*3) моделирование напряженно-деформированного состояния грунтового*

*массива (на базе инженерного пакета ABAQUS);*

*4) моделирование обтекания дымовой трубы воздушным потоком (на*

*базе инженерного пакета ANSYS CFX).*

*Разработка распределенных испытательных стендов для решения*

*данных задач показала возможность применения системы CAEBeans для*

*решения задач из различных областей инженерного моделирования.*

*105*

***З АК ЛЮ ЧЕ НИ Е***

*В диссертационной работе были рассмотрены вопросы, связанные с*

*внедрением систем инженерного проектирования и анализа в распределен-*

*ные вычислительные среды. Было произведено исследование современных*

*подходов по организации распределенных вычислительных систем. Были*

*исследованы основные аспекты внедрения систем инженерного проектиро-*

*вания и анализа в распределенную вычислительную среду. На основе про-*

*веденных исследований была предложена концепция распределенного вир-*

*туального испытательного стенда, обеспечивающая проблемно-*

*ориентированный подход к решению конкретных классов задач инженер-*

*ного проектирования посредством ресурсов, предоставляемых вычисли-*

*тельными грид-средами. Была предложена технология CAEBeans, в соот-*

*ветствии с которой, выделяются четыре слоя архитектуры РаВИС, каждый*

*из которых представляется своей оболочкой CAEBeans: Концептуальный*

*слой (проблемный CAEBean), Логический слой (потоковый CAEBean), Фи-*

*зический слой (компонентный CAEBean), Системный слой (системный*

*CAEBean). Разработан прототип комплекса программных средств «Система*

*CAEBeans», обеспечивающий поддержку разработки и исполнения РаВИС.*

*В состав системы входят компоненты: CAEBeans Constructor, CAEBeans*

*Portal, CAEBeans Server, CAEBeans Broker, CAE-ресурсы. На основе разра-*

*ботанного прототипа произведено испытание системы посредством разра-*

*ботки виртуальных испытательных стендов, ориентированных на задач ин-*

*женерного моделирования на базе ряда наиболее распространенных паке-*

*тов инженерного моделирования.*

*Работа выполнялась при поддержке Р ос на ук и (гос. контракт*

*2007-4-1.4-20-01-026) и н ау чн о-т ех ни че ск ойп ро гр ам мыС ою зн ог о г ос уд ар*

*с тв а Р ос си я-Б ел ор ус си я "С КИ Ф-Г РИ Д" (контракты 2007-СГ-04/4 и*

*2009-СГ-03).*

*В заключение перечислим основные полученные результаты диссер-*

*тационной работы, приведем данные о публикациях и апробациях, и рас-*

*смотрим направления дальнейших исследований в данной области.*

*106*

***О СН ОВ НЫ Е Р ЕЗ УЛ ЬТ АТ Ы Д ИС СЕ РТ АЦ ИО НН ОЙР АБ ОТ Ы***

*На защиту выносятся следующие новые научные результаты.*

*1. Разработана модель проблемно-ориентированного сервиса для реше-*

*ния задач инженерного проектирования и анализа в распределенных*

*вычислительных средах.*

*2. Разработаны архитектура и принципы структурной организации рас-*

*пределенного виртуального испытательного стенда (РаВИС), предос-*

*тавляющего сервис для решения задач инженерного анализа на осно-*

*ве грид-технологий.*

*3. Разработан комплекс методов и алгоритмов, позволяющих автомати-*

*зировать процесс построения специализированных РаВИС для реше-*

*ния прикладных задач с использованием различных CAE-пакетов.*

*4. Разработан прототип программной системы CAEBeans, включающий*

*в себя средства автоматического создания и исполнения РаВИС.*

*Произведены испытания системы CAEBeans путем создания РаВИС*

*на базе инженерных пакетов ANSYS CFX, ANSYS Mechanical,*

*ABAQUS, DEFORM. Распределенный виртуальный испытательный*

*стенд «Термообработка», внедрен в опытную эксплуатацию на пред-*

*приятии ОАО «Челябинский трубопрокатный завод».*

***П УБ ЛИ КА ЦИ И П О Т ЕМ Е Д ИС СЕ РТ АЦ ИИ*** *Основные результаты диссертации полностью опубликованы в сле-*

*дующих работах автора.*

*1. Радченко Г.И. Технология построения проблемно-ориентированных*

*иерархических оболочек над инженерными пакетами в грид-средах*

*// Системы управления и информационные технологии. № 4(34).*

*2008. С. 57-61.*

*107*

*2. Радченко Г.И., Соколинский Л.Б. Технология построения*

*виртуальных испытательных стендов в распределенных*

*вычислительных средах // Науч.-техн. вест. СПбГУ ИТМО. № 54.*

*2008. С. 134-139.*

*3. Радченко Г.И. Методы организации грид-оболочек системного слоя*

*в технологии CAEBeans // Вестник ЮУрГУ. Серия "Математическое*

*моделирование и программирование" № 15 (115). Вып. 1. 2008.*

*С. 69-78.*

*4. Радченко Г.И. Грид-система CAEBeans: интеграция ресурсов*

*инженерных пакетов в распределенные вычислительные среды //*

*Параллельные вычислительные технологии (ПаВТ’2009): Тр.*

*междунар. науч. конф. (Н.Н., 30 марта –3 апреля 2009 г.).*

*Челябинск: Изд-во ЮУрГУ, 2009. C. 281-292.*

*5. Радченко Г.И., Дорохов В.А., Насибулина Р.С., Соколинский Л.Б.,*

*Шамакина А.В. Технология создания виртуальных испытательных*

*стендов в грид-средах // Вторая Международная научная*

*конференция "Суперкомпьютерные системы и их применение"*

*(SSA'2008): доклады конференции (27-29 октября 2008 года, Минск)*

*Минск: ОИПИ НАН Беларуси, 2008. С. 194-198.*

*6. Радченко Г.И., Соколинский Л.Б., Шамакина А.В. Разработка*

*компонентно-ориентированных CAEBean-оболочек для пакета*

*ANSYS CFX // Параллельные вычислительные технологии*

*(ПаВТ'2008): Труды международной научной конференции (28*

*января - 1 февраля 2008 г., г. Санкт-Петербург). Челябинск: Изд-во*

*ЮУрГУ, 2008. C. 438-443.*

*108*

*7. Радченко Г.И., Соколинский Л.Б., Кутепов И.С. BeanShells:*

*интеграция CAE-пакетов в GPE // Параллельные вычислительные*

*технологии (ПаВТ'2007): Труды международной научной*

*конференции (Челябинск, 29 января –2 февраля 2007 г.). Челябинск:*

*Изд. ЮУрГУ, 2007. Т.2., C. 15*

*8. Свидетельство Роспатента об официальной регистрации программы*

*для ЭВМ № 2008612879. «CAEBeans Toolbox: программная среда*

*для разработки проблемно-ориентированных оболочек для грид» /*

*Юрков В.В., Дорохов В.А., Радченко Г.И., Насибулина Р.С.,*

*Шамакина А.В; Заяв. 03.10.2008.*

*9. Свидетельство Роспатента об официальной регистрации программы*

*для ЭВМ № 2008614998. «CAEBeans Sphere: программное средство*

*для поддержки распределенных вычислительных сред на базе*

*платформы Microsoft.NET» / Юрков В.В., Дорохов В.А., Радченко*

*Г.И., Насибулина Р.С., Шамакина А.В.; Заяв. 03.10.2008.*

*10. Свидетельство Роспатента об официальной регистрации программы*

*для ЭВМ № 2008612879. «Пакет проблемно-ориентированных*

*оболочек CAEBeans для решения типовых инженерных задач» /*

*Радченко Г.И., Насибулина Р.С., Шамакина А.В., Юрков В.В.,*

*Федянин О.Н., Дорохов В.А.; Заяв. 04.05.2008.*

*Работы 1, 2 опубликованы в журналах, включенных ВАК в перечень*

*журналов, в которых должны быть опубликованы основные результаты*

*диссертаций на соискание ученой степени доктора наук.*

***А ПР ОБ АЦ ИЯР АБ ОТ Ы***

*Основные положения диссертационной работы, разработанные моде-*

*ли, методы, алгоритмы и результаты вычислительных экспериментов док-*

*ладывались автором на следующих международных и всероссийских науч-*

*ных конференциях:*

*109*

* на VI Всероссийской межвузовской конференции молодых ученых*

*(14 - 17 апреля 2009г., Санкт-Петербург);*

* на V Всероссийской межвузовской конференции молодых ученых*

*(15 - 18 апреля 2008 г., Санкт-Петербург);*

* на Международной научной конференции «Параллельные вычисли-*

*тельные технологии» (30 марта –3 апреля 2009 г., Нижний Новго-*

*род);*

* на Международной научной конференции «Параллельные вычисли-*

*тельные технологии» (29 января –2 февраля 2007 г., Челябинск);*

* на Всероссийской научной конференции «Научный сервис в сети Ин-*

*тернет: технологии параллельного программирования» (24-29 сен-*

*тября 2007 г., Новороссийск).*

***Н АП РА ВЛ ЕН ИЯД АЛ ЬН ЕЙ ШИ Х И СС ЛЕ ДО ВА НИ Й***

*Теоретические исследования и практические разработки, выполнен-*

*ные в рамках этой диссертационной работы, предполагается продолжить по*

*следующим направлениям.*

*1. Исследование методов и алгоритмов внедрения систем многокрите-*

*риальной оптимизации, обеспечивающих работу в грид, в архитекту-*

*ру распределенного виртуального испытательного стенда.*

*2. Разработка методов и алгоритмов автоматической генерации исход-*

*ных кодов РаВИС для определенных классов задач инженерного про-*

*ектирования и анализа.*

*110*

***Л ИТ ЕР АТ УР А***

*1. Афанасьев А.П., Волошинов В.В., Кривцов В.Е. О возможных прин-*

*ципах организации доступа к удаленным вычислительным ресурсам на*

*основе CORBA. Тез. докл. 1-й Московской конференции «Декомпози-*

*ционные методы в математическом моделировании». ВЦ РАН, Моск-*

*ва, 2001. C. 11-14.*

*2. Бегунов А.А. Применение результатов моделирования для оптимиза-*

*ции и управления технологическими процессами // Параллельные вы-*

*числительные технологии: тр. Междунар. науч. конф. (28 янв. –1*

*февр. 2008 г., г. Санкт-Петербург). Челябинск: Изд. ЮУрГУ. 2008.*

*C. 31-38.*

*3. Бурмакин Е.М., Tuominen J.O. Интеграция технологии CORBA и объ-*

*ектных баз данных // XXIX Неделя науки СПбГТУ. Материалы межву-*

*зовской научной конференции. Ч.V. CПб.: Изд-во СПбГТУ, 2001.*

*С. 62-63.*

*4. Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя.*

*2-е изд. М.: ДМК Пресс, 2007. 496 c.*

*5. Буч Г., Якобсон А., Рамбо Дж. UML. Классика CS. 2-е изд. Спб.: Пи-*

*тер, 2006. 736 с.*

*6. Дорохов В.А., Маковецкий А.Н., Соколинский Л.Б. Разработка про-*

*блемно-ориентированной GRID-оболочки для решения задачи овали-*

*зации труб при закалке // Параллельные вычислительные технологии:*

*Труды международной научной конференции (28 января - 1 февраля*

*2008 г., г. Санкт-Петербург). -Челябинск: Изд-во ЮУрГУ. -2008. C.*

*520.*

*111*

*7. Дорохов В.А. Разработка виртуального испытательного грид-стенда*

*для исследования эффекта овализации труб при термической обработ-*

*ке // Параллельные вычислительные технологии (ПаВТ'2009): Труды*

*международной научной конференции (Нижний Новгород, 30 марта - 3*

*апреля 2009 г.). Челябинск: Изд-во ЮУрГУ, 2009. С. 457-462.*

*8. Лёушкин В.В., Соколинский Л.Б., Чайко К.А., Юрков В.В. Разработка*

*проблемно-ориентированной Grid-оболочки для моделирования резь-*

*бовых соединений труб для нефтяных скважин в распределенных вы-*

*числительных средах // Параллельные вычислительные технологии:*

*Труды международной научной конференции (28 января - 1 февраля*

*2008 г., г. Санкт-Петербург). Челябинск: Изд-во ЮУрГУ. 2008. C. 534.*

*9. Лукичев А.С. Интеграция SOA- и классических высокопроизводитель-*

*ных приложений // Научный сервис в сети Интернет: технологии рас-*

*пределенных вычислений: Труды Всероссийск. науч. конф. (18-23 сен-*

*тября 2006 г., г. Новороссийск). М.: Изд-во МГУ. 2006. C. 42-44.*

*10. Насибулина Р.С. и др. Методы организации программных интерфейсов*

*к инженерным пакетам в среде GPE // Параллельные вычислительные*

*технологии: Труды международной научной конференции (28 января -*

*1 февраля 2008 г., г. Санкт-Петербург). Челябинск: Изд-во ЮУрГУ,*

*2008. C. 537.*

*11. Радченко Г.И. Грид-система CAEBeans: интеграция ресурсов инже-*

*нерных пакетов в распределенные вычислительные среды // Парал-*

*лельные вычислительные технологии (ПаВТ'2009): Труды междуна-*

*родной научной конференции (Нижний Новгород, 30 марта - 3 апреля*

*2009 г.). Челябинск: Изд-во ЮУрГУ, 2009. С. 281-292.*

*112*

*12. Радченко Г.И. Методы организации грид-оболочек системного слоя в*

*технологии CAEBeans // Вестник ЮУрГУ. Серия "Математическое*

*моделирование и программирование". 2008. № 15 (115). Вып. 1.*

*С. 69-80.*

*13. Радченко Г.И. Технология построения проблемно-ориентированных*

*иерархических оболочек над инженерными пакетами в грид-средах //*

*Системы управления и информационные технологии. № 4(34). 2008.*

*С. 57-61.*

*14. Радченко Г.И., Соколинский Л.Б. Технология построения виртуальных*

*испытательных стендов в распределенных вычислительных средах //*

*Научно-технический вестник Санкт-Петербургского государственного*

*университета информационных технологий, механики и оптики. № 54.*

*2008. С. 134-139.*

*15. Радченко Г.И., Соколинский Л.Б., Шамакина А.В. Разработка компо-*

*нентно-ориентированных CAEBean-оболочек для пакета ANSYS CFX*

*// Параллельные вычислительные технологии (ПаВТ'2008): Труды ме-*

*ждународной научной конференции (28 января - 1 февраля 2008 г.,*

*г. Санкт-Петербург). Челябинск: Изд-во ЮУрГУ, 2008. C. 438-443.*

*16. Радченко Г.И., Соколинский Л.Б. CAEBeans: иерархические системы*

*структурированных проблемно-ориентированных оболочек над инже-*

*нерными пакетами// Научный сервис в сети Интернет: многоядерный*

*компьютерный мир. 15 лет РФФИ: Труды Всероссийск. науч. конф.*

*(24-29 сентября 2007 г., г. Новороссийск). М.: Изд-во МГУ, 2007.*

*C. 54-57.*

*17. Рамбо Дж., Блаха М. UML 2.0. Объектно-ориентированное моделиро-*

*вание и разработка. 2-е изд. СПб.: Питер, 2007. 544 с.*

*113*

*18. Суперкомпьютерный центр Южно-Уральского государственного уни-*

*верситета. URL: http://supercomputer.susu.ru/ (дата обращения:*

*19.05.2009)*

*19. Черняк Л. Web-сервисы, grid-сервисы и другие // Открытые системы.*

*СУБД. №12. 2004. С. 20-27.*

*20. Шамакина А.В. Организация брокера ресурсов в системе CAEBEANS*

*// Научный сервис в сети Интернет: решение больших задач: Труды*

*Всероссийск. науч. конф. (22-27 сентября 2008 г., Новороссийск).*

*М.: Изд-во МГУ, 2007. С. 326-327.*

*21. Шамакина А.В. Организация брокера ресурсов в системе CAEBeans //*

*Вестник Южно-Уральского государственного университета. Серия*

*"Математическое моделирование и программирование". 2008.*

*№ 27(127). Вып. 2. C. 110-116.*

*22. Aleksy M., Gitzel R. Relationship between the standard compliance of IDLcompilers*

*and interoperability of CORBA-based applications // Proceedings*

*of the 1st international symposium on Information and communication*

*technologies. Dublin, Ireland. Dublin: Trinity College, 2003. P. 512-517.*

*23. Aberer K., Hauswirth M. Peer-to-peer information systems: concepts and*

*models, state-of-the-art, and future systems // Proceedings of the 8th European*

*software engineering conference held jointly with 9th ACM SIGSOFT*

*international symposium on Foundations of software engineering. New*

*York: ACM, 2001. P. 326-327.*

*24. Alhaisoni M., Liotta A. Characterization of signaling and traffic in Joost //*

*Peer-to-Peer Networking and Applications. 2008. Vol. 2, No. 1. P. 75-83.*

*25. Architectural Overview of OGSA-BES Adoption in UNICORE 6 // Forschungszentrum*

*Jüich, 2008. 18 p.*

*114*

*26. Bastide Remi et al. Formal specification of CORBA services: experience*

*and lessons learned // Proceedings of the 15th ACM SIGPLAN conference*

*on Object-oriented programming, systems, languages, and applications.*

*New York: ACM. 2000. P. 105-117.*

*27. BitTorrent. URL: http://www.bittorrent.com/ (дата обращения:*

*19.05.2009).*

*28. BOINC - Berkeley Open Infrastructure for Network Computing. URL:*

*http://boinc.berkeley.edu/ (дата обращения: 19.05.2009).*

*29. Brebner P., Emmerich W. Two Ways to Grid: The Contribution of Open*

*Grid Services Architecture (OGSA) Mechanisms to Service-Centric and*

*Resource-Centric Lifecycles // Journal of Grid Computing. Vol. 5, No. 1.*

*2006. P. 151-131.*

*30. Bucci G., Streeter D.N. A methodology for the design of distributed information*

*systems // Communications of the ACM. Vol. 22, Issue 4. 1979.*

*P. 233-245.*

*31. Business Process Execution Language for Web Services Version 1.1, URL:*

*http://www-106.ibm.com/developerworks/webservices/library/ws-bpel*

*(дата обращения: 19.05.2009)*

*32. Buriola T.M., Scheer S. CAD and CAE Integration Through Scientific Visualization*

*Techniques for Illumination Design // Tsinghua Science &*

*Technology. Vol. 13, No. 1. 2008. P. 26-33.*

*33. Buyya R., Abramson D., Venugopal S. The Grid Economy. Proceedings of*

*the IEEE. Vol. 93, Issue 3. 2005 P. 698-714.*

*34. Buyya R. et al. Cloud computing and emerging IT platforms: Vision, hype,*

*and reality for delivering computing as the 5th utility // Future Generation*

*Computer Systems. 2009. № 25. P. 599-616.*

*115*

*35. Canevet C. et. al. Analysing UML 2.0 activity diagrams in the software performance*

*engineering process // Proceedings of the 4th international workshop*

*on Software and performance. New York: ACM, 2004. P. 74-78.*

*36. Chao K.-M., Younas M., Griffiths N. BPEL4WS-based coordination of*

*Grid Services in design // Computers in Industry. Vol. 57. 2006. P. 778-786.*

*37. Cheng H.-C., Fen C.-S. A web-based distributed problem-solving environment*

*for engineering applications // Advances in Engineering Software.*

*Vol. 37, Issue 2. 2006. P. 112-128.*

*38. Chin R.S., Chanson S.T. Distributed, object-based programming systems //*

*ACM Computing Surveys (CSUR). Vol. 23, Issue 1. 1991. P. 91-124.*

*39. Coatta T. Corba: Gone but (Hopefully) Not Forgotten // Queue. Vol. 5, Issue*

*4. 2007. P. 3.*

*40. Cols D.R. Business-structured client/server: an architecture for distributed*

*applications // Proceedings of the 1993 conference of the Centre for Advanced*

*Studies on Collaborative research: software engineering. IBM Press,*

*1993. Vol. 1. P. 41-53.*

*41. Coulson G., Baichoo S. Implementing the CORBA GIOP in a highperformance*

*object request broker environment // Distributed Computing.*

*Vol. 14, No. 2. 2001. P. 113-126.*

*42. Credle R. et al. SOA Approach to Enterprise Integration for Product*

*Lifecycle Management. IBM, 2008. 506 p.*

*43. Czajkowski K., Ferguson D., Foster I. et al. From Open Grid Services*

*Infrastructure to WS-Resource Framework: Refactoring & Evolution. –The*

*Globus Project Whitepaper. 2004. URL: http://www.globus.org/wsrf/specs/*

*ogsi\_to\_wsrf\_1.0.pdf (дата обращения: 02.06.2009).*

*44. Dogac A., Dengi C., Оszu M.Т. Distributed Object Computing Platforms //*

*Communications of the ACM. Vol. 41, Issue 9. 1998. P. 95-103.*

*116*

*45. Domazet D. et al. An infrastructure for inter-organizational collaborative*

*product development // Proceedings of the 33rd Annual Hawaii*

*International Conference on System Sciences. 2000. 10 pp.*

*46. Ernemann C. et al. On advantages of grid computing for parallel job*

*scheduling. // Proc. of the 2nd IEEE Int’l. Symp. on Cluster Computing and*

*the Grid (CCGrid). Washington, DC: IEEE Computer Society, 2002.*

*P 39-49.*

*47. Evaristo J.R., Desouza K.C., Hollister K. Centralization momentum: the*

*pendulum swings back again // Communications of the ACM. Vol. 48,*

*Issue 2. 2005. P. 66-71.*

*48. Fan L.Q. et al. Development of a distributed collaborative design*

*framework within peer-to-peer environment. // Computer-Aided Design.*

*Vol. 40. 2008. P. 891–04.*

*49. Ferguson D. Trends and statistics in peer-to-peer. Workshop on technical*

*and legal aspects of peer-to-peer television. Amsterdam, Netherlands, 2006,*

*Mar. 17. 2006. URL:*

*http://www2.noticiasdot.com/publicaciones/2006/0406/1804*

*/noticias/images/CacheLogic\_AmsterdamWorkshop\_Presentation\_v1.0.ppt*

*(дата обращения: 02.06.2009).*

*50. Finlayson N., Morrison J. P2P and Client-Server Hybrids: Groove-enabling*

*a J2EE portal using web services // Networking and Electronic Commerce*

*Research Conference (NAEC 2005), October 6-9, 2005, Riva Del Garda, Italy.*

*URL: http://www.atlanticshack.com/resources/P2P\_and\_Client-*

*Server\_NAEC2005\_final.pdf (дата обращения: 15.05.2009).*

*51. Foster I.T., Kesselman C., Nick J., Tuecke S. The Physiology of the Grid:*

*An Open Grid Service Architecture for Distributed Systems Integration. /*

*Global Grid Forum. 2002. URL: http://www.globus.org/ogsa/ (дата*

*обращения: 14.05.2009).*

*117*

*52. Foster I.T. Service-Oriented Science // Science. 2005. Vol. 308, No. 5723.*

*P. 814–17.*

*53. Foster I. et al. Grid Services for Distributed System Integration // Computer.*

*Vol. 35, Issue 6. 2002. P. 37-46.*

*54. Foster I. et al. How Do I Model State? Let Me Count the Ways // Queue.*

*Vol. 7, Issue 2. 2009. P. 54-64.*

*55. Foster I. et al. Modeling and Managing State in Distributed Systems: The*

*Role of OGSI and WSRF // Proceedings of the IEEE. Vol. 93, Issue 3.*

*2005. P. 604-612.*

*56. Foster I. Globus Toolkit Version 4: Software for Service-Oriented Systems*

*// IFIP International Conference on Network and Parallel Computing.*

*Springer, 2005. P. 2-13.*

*57. Foster I., Frey J., Graham S. et al. Modeling Stateful Resources with Web*

*Services / The Globus Project Whitepaper, 2004 URL:*

*http://www.ibm.com/developerworks/library/ws-resource/wsmodelingresources.*

*pdf (дата обращения: 16.05.2009).*

*58. Foster I., Iamnitchi A. On death, taxes, and the convergence of peer-to-peer*

*and grid computing // In 2nd International Workshop on Peer-to-Peer*

*Systems (IPTPS’03). Springer, 2003. P. 118-128.*

*59. Foster I., Kesselman C. Globus: A Metacomputing Infrastructure Toolkit //*

*International Journal of Supercomputer Applications Vol. 11, Issue 2. 1997.*

*P. 115-128.*

*60. Foster I., Kesselman C. The Grid. Blueprint for a new computing*

*infrastructure. San Francisco: Morgan Kaufman, 1999. 677 p.*

*118*

*61. Foster I., Kesselman C., Tuecke S. The Anatomy of the Grid: Enabling*

*Scalable Virtual Organizations // International Journal of Supercomputer*

*Applications and High Performance Computing. 2001. Vol. 15, No 3.*

*P. 200-222.*

*62. Foster I., Kishimoto H., Savva A. et al. The Open Grid Services*

*Architecture. 2005. 62 p. URL: http://forge.gridforum.org/projects/ogsa-wg*

*pdf (дата обращения: 16.05.2009).*

*63. Foster I., Jennings N.R., Kesselman C. Brain Meets Brawn: Why Grid and*

*Agents Need Each Other // Proceedings of the Third International Joint*

*Conference on Autonomous Agents and Multiagent Systems. Washington,*

*DC: IEEE Computer Society, 2004. Vol. 1. P. 8-15.*

*64. Foster I. What is the Grid? A Three Point Checklist. 2002. URL:*

*http://www.fp.mcs.anl.gov/~foster/Articles/WhatIsTheGrid.pdf (дата об-*

*ращения: 17.05.2009).*

*65. Fox G. C., Gannon D. Workflow in Grid Systems // Concurrency and*

*Computation: Practice & Experience. Vol. 18. No. 10. 2006. P. 1009-1019.*

*66. Gnutella. URL: http://www.gnutella.com/ (дата обращения: 25.05.2009)*

*67. Guha S., Daswani N., Jain R. An experimental study of the Skype peer-topeer*

*VoIP system. In: Proceedings of the IPTPS’06. Santa Barbara, CA,*

*Feb. URL: http://iptps06.cs.ucsb.edu/talks/guha-skype-talk.pdf (дата обра-*

*щения: 25.05.2009).*

*68. Gray N.A.B. Comparison of Web Services, Java-RMI, and CORBA service*

*implementations // Australian Software Engineering Conference (ASWEC*

*2004). Melbourne, Australia (April 13 and 14, 2004). Melburne: Swinburne*

*Univercity Of Technology. 2004. P. 1-12.*

*69. Hayes B. Cloud computing // Communication of the ACM. Vol. 51. Issue 7.*

*2008. P. 9-11.*

*119*

*70. Haynos M. Perspectives on grid: Grid computing –Next-generation*

*distributed computing / IBM Whitepaper. 2004. 11 p. URL: http://www-*

*128.ibm.com/developerworks/grid/library/gr-heritage/ (дата обращения:*

*19.06.2009).*

*71. Henning M. The Rise and Fall of CORBA // ACM Queue. Vol. 4, Num. 5.*

*2006. P. 28-34.*

*72. Henning M, Vinoski S. Advanced CORBA programming with C++.*

*New York: Addison-Wesley, 1999. 1120 p.*

*73. Hollingsworth J., Powell D. The Implementation of an Evolutionary-Based*

*Engineering Optimization Framework for the Grid / Elon University. 2007.*

*URL: http://www.cs.uncc.edu/~abw/ITCS4146S07/uncc.pdf (дата обраще-*

*ния: 19.05.2009).*

*74. Iosup A., Epema D. et al. On Grid Performance Evaluation using Synthetic*

*Workloads // CoreGRID Technical Report Number TR-0039. 2006. 18 p.*

*75. Jagannathan V., Almasi G., Suvaiala A. Collaborative infrastructures using*

*the WWW and CORBA-based environments // Proceedings of the IEEE*

*Workshops on Enabling Technologies Infrastructure for Collaborative*

*Enterprises (WETICE’96). 1996. P. 292–97.*

*76. Jalonski S., Bussker C. Workflow management: modeling concepts,*

*architecture and implementation. London: International Thomson Computer*

*Press, 1996. 351 p.*

*77. Java RMI Tutorial. URL: http://java.sun.com/docs/books/tutorial/rmi/*

*index.html (дата обращения: 02.06.2009).*

*78. Joncheere N., Deridder D., Van Der Straeten R., Jonckers V. A Framework*

*for Advanced Modularization and Data Flow in Workflow Systems //*

*Service-Oriented Computing –ICSOC 2008. Vol. 5364. 2008. P. 592-598.*

*120*

*79. Kamel M., Leue S. Formalization and validation of the General Inter-ORB*

*Protocol (GIOP) using PROMELA and SPIN // International Journal on*

*Software Tools for Technology Transfer (STTT). Vol. 2, No. 4. 2000.*

*P. 394-409.*

*80. Kim J.-H. et al. A Problem Solving Environment Portal for Multidisciplinary*

*Design Optimization. // Advances in Engineering Software. Vol. 40.*

*Issue 8. 2009. P. 623-629.*

*81. King J.L. Centralized versus decentralized computing: organizational considerations*

*and management options // ACM Computing Surveys. Vol. 15,*

*Issue 4. 1983. P. 319-349.*

*82. Krüer М. Grid Computing & UNICORE. 2006. URL:*

*http://www.be.itu.edu.tr/news/grid\_atolye/UNICOREIntroduction.pdf (дата*

*обращения: 16.05.2009).*

*83. Leach P., Mealling M., Salz R. A Universally Unique IDentifier (UUID)*

*URN Namespace. RFC 4122. July 2005.*

*84. Lee B.-E., Suh S.-H. An architecture for ubiquitous product life cycle*

*support system and its extension to machine tools with product data model*

*// The International Journal of Advanced Manufacturing Technology.*

*Vol. 1. 2009. -P. 15.*

*85. Lee H.-J., Lee J.-W., Lee J.-O. Development of Web services-based*

*Multidisciplinary Design Optimization framework. // Advances in*

*Engineering Software. Vol. 40. No. 3. 2009. P. 176-183.*

*86. Li J. On peer-to-peer (P2P) content delivery // Peer-to-Peer Networking and*

*Applications. 2008. Vol. 1, No. 1. P. 45-63.*

*87. Li Z. et al. Architecture of collaborative design grid and its application*

*based on LAN // Advances in Engineering Software. Vol. 38, № 2. 2007.*

*P. 121-132.*

*121*

*88. Li Z. et al. Conception and implementation of a collaborative manufacturing*

*grid // The International Journal of Advanced Manufacturing Technology.*

*Vol. 34, № 11-12. 2007. P. 1224-1235.*

*89. Lutteroth C., Weber G. Efficient Use of GUIDs // Proceedings of the 2008*

*Ninth international Conference on Parallel and Distributed Computing, Applications*

*and Technologies (December 01 - 04, 2008). Washington, DC:*

*IEEE Computer Society, 2008. P.115-120.*

*90. Memon M. S. et al. Enhanced resource management capabilities using standardized*

*job management and data access interfaces within UNICORE Grids*

*// 13th International Conference on Parallel and Distributed Systems. Julich:*

*Central Inst. of Appl. Math, 2007. Vol. 2. P. 1-6.*

*91. Mervyn F., Senthil Kumar A., Bok S.H., Nee A.Y.C. Developing distributed*

*applications for integrated product and process design // Computer-*

*Aided Design. Vol 36, Issue 8. 2004. P. 679–89.*

*92. Milojicic D. S. et al. Peer-to-Peer Computing, Hewlett-Packard, Tech. Rep.*

*HPL-2002-57R1. 2003. URL: http:// www.hpl.hp.com/ techreports/*

*2002/HPL-2002-57R1.html (дата обращения: 19.05.2009).*

*93. Milojicic D. Cloud computing: Interview with Russ Daniels and Franco*

*Travostino // IEEE Internet Computing. Vol. 15, Issue 5. 2008. P. 7–.*

*94. Munoz C., Zalewski J. Archictecture and Performance of Java-Based*

*Distributed Object Models: CORBA vs RMI // Real-Time Systems. Vol. 21,*

*No. 1-2. 2001. P. 43-75.*

*95. Nolan R.L. Managing the computer resource: a stage hypothesis //*

*Communications of the ACM. Vol. 16, Issue 7. 1973. P. 399-405.*

*96. Object Management Group (OMG), The Common Object Request Broker*

*Architecture (CORBA). URL: http://www.corba.org (дата обращения:*

*19.05.2009).*

*122*

*97. Olson M.H. Remote office work: changing work patterns in space and time*

*// Communications of the ACM. Vol. 26, Issue 3. 1983. P. 182-187.*

*98. Papazoglou M.P., Georgakopoulos D. Service-Oriented Computing //*

*Communications of the ACM. Vol. 46, No. 10. 2003. P. 25-28.*

*99. Peer-to-peer Working Group. URL: http://www.p2pwg.org (дата обраще-*

*ния: 19.05.2009).*

*100. Prodan R., Fahringer T. From Web Services to OGSA: Experiences in Implementing*

*an OGSA-based Grid Application // Proceedings of the 4th International*

*Workshop on Grid Computing. Washington, DC: IEEE Computer*

*Society, 2003. P. 2-10.*

*101. Python Programming Language - Official Website. URL: www.python.org*

*(дата обращения: 19.05.2009)*

*102. Raphael B., Smith I.F.C. Fundamentals of computer aided engineering. John*

*Wiley, 2003. 306 p.*

*103. Riedel M., Mallmann. D. Standardization Processes of the UNICORE Grid*

*System // Proceedings of 1st Austrian Grid Symposium 2005, Schloss Hagenberg,*

*Austria. Austrian Computer Society, 2005. P: 191–03.*

*104. Russell N. et. al. On the suitability of UML 2.0 activity diagrams for business*

*process modelling // Proceedings of the 3rd Asia-Pacific conference on*

*Conceptual modelling. Hobart: Australian Computer Society, Inc, 2006.*

*Vol. 53. P. 95-104.*

*105. Schnekenburger T. Load Balancing in CORBA: A Survey of Concepts, Patterns,*

*and Techniques // The Journal of Supercomputing. Vol. 15, No. 2.*

*2000. P. 141-161.*

*106. Service Oriented Architecture (SOA) Reference Model Public Review Draft*

*1.0 (Feb) / Organization for the Advancement of Structured Information*

*Standards (OASIS), 2006. 28 p.*

*123*

*107. Shan H., Oliker L., Biswas R. Job superscheduler architecture and performance*

*in computational grid environments // Proceedings of the ACM/IEEE*

*conference on Supercomputing. Washington, DC: IEEE Computer Society,*

*2003. P. 44–4.*

*108. Silva V.T., Noya R.C., Lucena C.J.P. Using the UML 2.0 activity diagram*

*to model agent plans and actions // Proceedings of the fourth international*

*joint conference on Autonomous agents and multiagent systems.*

*New York: ACM, 2005. P. 594-600.*

*109. Sinha A. Client-server computing // Communications of the ACM. Vol. 35,*

*Issue 7. 1992. P. 77-98.*

*110. Skype. URL: http://www.skype.com (дата обращения: 02.06.2009).*

*111. Sotomayor B. The Globus Toolkit 4 Programmer’s Tutorial / University of*

*Chicago, Department of Computer Science, 2005. URL:*

*http://gdp.globus.org/gt4-tutorial/download/progtutorial-pdf\_0.2.1.tar.gz*

*(дата обращения: 02.06.2009).*

*112. Stevens R. et. al. From the I-WAY to the National Technology Grid //*

*Communications of the ACM. Vol. 40, Issue 11. 1997. P. 50-60.*

*113. Stockinger H. Defining the Grid: A Snapshot on the Current View // The*

*Journal of Super-computing. 2007. № 42(1). P. 3-17.*

*114. A. Streit. UNICORE: Getting to the heart of Grid technologies // eStrategies.*

*Vol. 3. 2009. P. 8-9.*

*115. A. Streit. UNICORE - What lies beneath Grid functionality? // eStrategies.*

*Vol. 7. 2008. P. 38-39.*

*116. Sundaram B. Understanding WSRF, Part 1: Using WS-ResourceProperties.*

*2005. URL: http://www.ibm.com/developerworks/edu/gr-dw-gr-wsrf1-*

*i.html (дата обращения: 02.06.2009).*

*124*

*117. Taylor I., Harrison A. From P2P and Grids to Services on the Web. Springer.*

*2008. 462 p.*

*118. Twenty-One Experts Define Cloud Computing. URL:*

*http://cloudcomputing.sys-con.com/read/612375\_p.htm (дата обращения:*

*18.05.2009).*

*119. Vaquero L. M. et al. A break in the clouds: towards a cloud definition.*

*ACM SIGCOMM Computer Communication Review. Vol. 39, Issue 1.*

*2009. P. 50-55.*

*120. Vogels W. Web Services Are Not Distributed Objects // IEEE Internet*

*Computing. Vol. 7, No. 6. 2003. P. 59-66.*

*121. Yin J.W., Zhang W.Y., Li Y., Chen H.W. A peer-to-peer-based multi-agent*

*framework for decentralized grid workflow management in collaborative*

*design. // The International Journal of Advanced Manufacturing Technology.*

*Vol. 41, № 3-4. 2009. P. 407-420.*

*122. Wasson G., Humphrey M. Policy and Enforcement in Virtual Organizations*

*// Proceedings of the 4th International Workshop on Grid Computing.*

*Washington, DC: IEEE Computer Society, 2003. P. 125-133.*

*123. Weiss A. Computing in the Clouds // netWorker. № 11(4). 2007. P. 16-25.*

*124. Xue G., Song W., Cox S.J., Keane An. Numerical Optimisation as Grid*

*Services for Engineering Design // Journal of Grid Computing. 2004. Vol.*

*2. P. 223-238.*

*125. Zalila B., Hugues J., Pautet L. An improved IDL compiler for optimizing*

*CORBA applications // Proceedings of the 2006 annual ACM SIGAda international*

*conference on Ada. Albuquerque, New Mexico, USA. New*

*York: ACM, 2006. P. 21-28.\_\_*