

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

СОГЛАСОВАНО
Генеральный директор
ЗАО «АйТи»


Бакиев О.Р.
2011 г.

УТВЕРЖДАЮ
Ректор НИУ ИТМО


Васильев В.Н.
2011 г.

**МНОГОПРОФИЛЬНАЯ ИНСТРУМЕНТАЛЬНО-
ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА СОЗДАНИЯ
И УПРАВЛЕНИЯ РАСПРЕДЕЛЕННОЙ СРЕДОЙ
ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ CLAVIRE**

**ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА СОЗДАНИЯ ИНФОРМАЦИОННО-
ПРОГРАММНЫХ ПРЕДМЕТНО-ОРИЕНТИРОВАННЫХ ЦЕНТРОВ
КОМПЕТЕНЦИИ НА ОСНОВЕ ПРИКЛАДНЫХ СЕРВИСОВ (ПРОГРАММ) В
РАМКАХ КОНЦЕПЦИИ SAAS (МИП-Ц)**

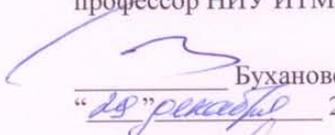
Описание применения

ЛИСТ УТВЕРЖДЕНИЯ

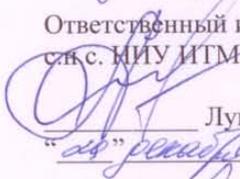
RU.СНАБ.80066-06 31 04-ЛУ

Представители
Организации-разработчика

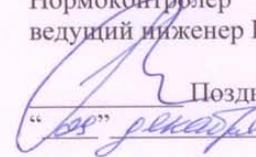
Руководитель разработки,
профессор НИУ ИТМО


Бухановский А.В.
"28" декабря 2011 г.

Ответственный исполнитель,
ст.с. НИУ ИТМО


Луценко А.Е.
"28" декабря 2011 г.

Нормоконтролер
ведущий инженер НИУ ИТМО


Позднякова Л.Г.
"28" декабря 2011 г.

2011

Име.№ подл.	Подп. и дата	Взам.име.№	Име.№ дубл.	Подп. и дата

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

УТВЕРЖДЕН
RU.СНАБ.80066-06 31 04-ЛУ

**МНОГОПРОФИЛЬНАЯ ИНСТРУМЕНТАЛЬНО-
ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА СОЗДАНИЯ
И УПРАВЛЕНИЯ РАСПРЕДЕЛЕННОЙ СРЕДОЙ
ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ CLAVIRE**

**ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА СОЗДАНИЯ ИНФОРМАЦИОННО-
ПРОГРАММНЫХ ПРЕДМЕТНО-ОРИЕНТИРОВАННЫХ ЦЕНТРОВ
КОМПЕТЕНЦИИ НА ОСНОВЕ ПРИКЛАДНЫХ СЕРВИСОВ (ПРОГРАММ) В
РАМКАХ КОНЦЕПЦИИ SAAS (МИП-Ц)**

Описание применения

RU.СНАБ.80066-06 31 04

ЛИСТОВ 33

2011

Ине.№ подл.	
Подп. и дата	
Взам.ине.№	
Ине.№ дубл.	
Подп. и дата	

RU.СНАБ.80066-06 31 04 **Ошибка! Источник ссылки не найден.**

АННОТАЦИЯ

Документ содержит описание применения технологической платформы создания информационно-программных предметно-ориентированных центров компетенции на основе прикладных сервисов (программ) в рамках концепции SaaS (МИТП-Ц) RU.СНАБ.80066-06 01 43. Технологическая платформа МИТП-Ц входит в состав многопрофильной инструментально-технологической среды (МИТП) CLAVIRE (Cloud Applications Virtual Environment) RU.СНАБ.80066-06. Она предназначена для создания программной инфраструктуры центров компетенции, предоставляющих доступ к высокопроизводительным приложениям в рамках моделей облачных вычислений SaaS и AaaS. МИТП-Ц разработана в ходе выполнения проекта «Создание распределенной вычислительной среды на базе облачной архитектуры для построения и эксплуатации высокопроизводительных композитных приложений» (Договор № 21057 от 15 июля 2010 г., шифр 2010-218-01-209) в рамках реализации постановления Правительства РФ № 218 «О мерах государственной поддержки развития кооперации российских высших учебных заведений и организаций, реализующих комплексные проекты по созданию высокотехнологичного производства».

СОДЕРЖАНИЕ

1. НАЗНАЧЕНИЕ ПРОГРАММЫ	4
1.1. Функциональное назначение	4
1.2. Область применения	5
1.3. Основные характеристики	5
1.4. Ограничения, накладываемые на область применения	7
2. УСЛОВИЯ ПРИМЕНЕНИЯ	8
2.1. Условия развертывания программы	8
2.2. Необходимые технические средства управляющей подсистемы МИТП-Ц	10
2.3. Необходимые технические средства подсистемы вычислительной инфраструктуры МИТП-Ц	10
3. ОПИСАНИЕ ЗАДАЧИ	12
3.1. Определение задачи	12
3.2. Методы решения задачи	15
3.2.1. Унифицированное описание прикладных пакетов в МИТП-Ц	16
3.2.2. Унифицированное описание композитных приложений в МИТП-Ц	18
3.2.3. Организация процесса исполнения композитного приложения в МИТП-Ц	21
3.2.4. Интеллектуальная поддержка создания композитных приложений в МИТП-Ц	23
3.2.5. Особенности создания и управления информационно-программных предметно- ориентированных центров компетенции на основе прикладных сервисов (программ) в рамках концепции SaaS	27
3.2.6. Решение типовой задачи компьютерного моделирования сложных систем с использованием МИТП-Ц	28
4. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ	30
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ	32
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ	33

1. НАЗНАЧЕНИЕ ПРОГРАММЫ

Технологическая платформа создания информационно-программных предметно-ориентированных центров компетенции на основе прикладных сервисов (программ) в рамках концепции SaaS (МИТП-Ц) RU.СНАБ.80066-06 01 43 входит в состав многопрофильной инструментально-технологической среды (МИТП) CLAVIRE (Cloud Applications Virtual Environment) RU.СНАБ.80066-06. Она предназначена для создания программной инфраструктуры центров компетенции, предоставляющих доступ к высокопроизводительным приложениям в рамках моделей облачных вычислений SaaS и AaaS.

1.1. Функциональное назначение

МИТП-Ц представляет собой комплекс программного обеспечения для разработки, настройки и эксплуатации сред распределенных вычислений центров компетенции в различных предметных областях, предназначенный для:

- 1) эффективного управления вычислительными, информационными и программными ресурсами центров компетенции, включая собственные (выделенные) вычислительные ресурсы центров и ресурсы внешних провайдеров (в том числе глобальных коллаборативных сред);
- 2) создания, исполнения, управления и предоставления сервисов доступа к предметно-ориентированным высокопроизводительным композитным приложениям в конкретной предметной области, функционирующим на основе облака прикладных сервисов в распределенной среде смешанного типа¹;
- 3) обеспечения функционирования программно-аппаратного комплекса (ПАК) центра компетенции для поддержки инфраструктуры предметно-ориентированных облачных вычислений в конкретной предметной области.

¹ Прикладным сервисом называется программа, у которой входные и выходные данные интерпретируются в терминах конкретной предметной области и которая в распределенной среде через публичную либо корпоративную сеть передачи данных доступна для выполнения без использования вычислительных и программных ресурсов на стороне пользователя.

1.2. Область применения

Технологическая платформа МИТП-Ц предназначена для создания программной инфраструктуры центров компетенции, предоставляющих доступ к высокопроизводительным приложениям в рамках моделей облачных вычислений SaaS и AaaS. Для функционирования этих приложений используются как собственные (корпоративные) ресурсы центров компетенции, так и привлекаемые ресурсы внешних провайдеров (выделенные суперкомпьютеры, Грид-среды, среды облачных вычислений первого поколения). МИТП-Ц применима для решения следующих классов задач.

1. Исследование и разработка композитных приложений для численного моделирования сложных неоднородных систем и объектов на основе исчерпывающего набора прикладных пакетов для конкретных предметных областей.
2. Обеспечение удаленного доступа к уникальным сервисам и композитным приложениям в конкретных предметных областях в рамках единого рабочего пространства центра компетенции.
3. Обеспечение интеллектуальной поддержки пользователей центров компетенции на всех этапах жизненного цикла их расчетных задач, включая выбор сервисов, разработку и настройку композитных приложений, подготовку данных, запуск задач на исполнение и интерпретацию результатов.
4. Организация динамических распределенных сред на основе ресурсов центров компетенции и внешних провайдеров для решения конкретных ресурсоемких вычислительных задач по запросу пользователя.

1.3. Основные характеристики

Технологическая платформа МИТП-Ц реализована на основе МИТП CLAVIRE в рамках концепции iPSE (Intelligent Problem Solving Environment). Она ориентирована на развитие интеллектуальных технологий поддержки жизненного цикла проблемно-ориентированных сред распределенных вычислений, обеспечивающих деятельность центров компетенции.

Технологическая платформа МИТП-Ц является специализированным средством для создания инфраструктуры проблемно-ориентированных центров компетенции в различных предметных областях. Задачей центров компетенции является, в частности, интеграция (в единый пул) прикладных пакетов (в т.ч. уникальных), покрывающих решение определенного класса задач, а также необходимых для этого вычислительных

RU.СНАБ.80066-06 31 04 **Ошибка! Источник ссылки не найден.**

ресурсов и источников данных. При этом обоснование выбора и принципов взаимосвязи пакетов является задачей специалистов-экспертов центра компетенции. Потому пользователь такого центра обеспечивается не только доступом к конкретным сервисам, но и интеллектуальной поддержкой в процессе своей работы. При этом обеспечивается возможность исполнения выбранного сервиса или приложения на необходимых для этого ресурсах: собственных выделенных системах или на привлеченных ресурсах внешних провайдеров. МИТП-Ц обеспечивает выполнение следующих функций:

- 1) Поддержка разработки и исполнения прикладных сервисов и приложений на их основе на ресурсах информационно–программных предметно-ориентированных центров компетенции – набора программно-аппаратных комплексов, принадлежащих одному владельцу, предназначенных для решения специфических классов задач предметной области, с обеспечением доступа пользователей к ним через сети общего назначения (Интернет).
- 2) Динамическое управление (мониторинг состояния, запуск приложений, передача данных, распределение нагрузки, репликация задач) в автоматическом режиме набором вычислительных ресурсов информационно–программных предметно-ориентированных центров компетенции.
- 3) Автоматическая оптимизация по времени процесса использования доступных вычислительных ресурсов и прикладных сервисов информационно–программных предметно-ориентированных центров компетенции.
- 4) Представление описания композитных приложений, создаваемых на основе ресурсов информационно–программных предметно-ориентированных центров компетенции, в форме цепочек заданий (workflow), обеспечивающих запуск, выполнение, остановку и возобновление работы цепочки заданий в ручном и автоматическом режимах.
- 5) Унифицированный доступ к вычислительным ресурсам информационно–программных предметно-ориентированных центров компетенции на основе интерфейсов доступа в составе программно-аппаратной архитектуры SMP, MPP, GPGPU, СВЕА.
- 6) Поддержка процесса установки и первоначальной конфигурации технологической платформы и ее составных частей на ресурсах информационно–программных предметно-ориентированных центров компетенции.
- 7) Поддержка многопользовательского режима доступа к ресурсам и сервисам информационно-программных предметно-ориентированных центров компетенции.

RU.СНАБ.80066-06 31 04 **Ошибка! Источник ссылки не найден.**

- 8) Квотирование, биллинг и тарификация использования ресурсов информационно-программных предметно-ориентированных центров компетенции.
- 9) Каталогизация входных данных пользователей на основе метаданных.
- 10) Администрирование и контроль работы с дифференцированными правами администраторов в рамках многоуровневой политики доступа к ресурсам информационно–программных предметно-ориентированных центров компетенции.
- 11) Модификация знаний, используемых системой в ручном и автоматическом режимах.
- 12) Функционирование сервисов резервирования и отката исправлений для пользовательских данных в удаленном хранилище в составе информационно-программных предметно-ориентированных центров компетенции.
- 13) Функционирование механизмов конвертирования данных между различными прикладными сервисами, по заданию пользователя.
- 14) Функционирование сервисов, обеспечивающих интерфейс с многофункциональными и предметно-ориентированными системами визуализации результатов расчетов (в составе сервисов информационно-программных предметно-ориентированных центров компетенции) с использованием прикладных сервисов и композитных приложений на их основе.

1.4. Ограничения, накладываемые на область применения

Специфика использования смешанной (корпоративно-коллаборативной) распределенной среды накладывает на применение МИТП-Ц следующие ограничения:

- 1) в состав МИТП-Ц не входят прикладные пакеты; они регистрируются в МИТП-Ц исходя из специфики деятельности конкретного центра компетенции;
- 2) идентификация и аутентификация корпоративных пользователей происходит на основе единого сертификата центра компетенции. Это обеспечивает доступ пользователя только к ресурсам центра; для использования ресурсов внешних провайдеров необходима дополнительная авторизация, например, с использованием персональных сертификатов Грид;
- 3) частично ограничены права пользователя по целевому выбору ресурса (предпочтение отдается собственным ресурсам центра компетенции);
- 4) эффективность использования средств интеллектуальной поддержки МИТП-Ц зависит от специфики предметной области (в частности, от возможности построить ее онтологическое описание).

2. УСЛОВИЯ ПРИМЕНЕНИЯ

2.1. Условия развертывания программы

Установка и настройка системных компонентов МИТП-Ц производится посредством компонента развертывания и конфигурирования RU.СНАБ.80066-06 01 36. Данный компонент предоставляет графический интерфейс для решения следующих задач:

- 1) полуавтоматическое развертывание компонентов МИТП-Ц;
- 2) конфигурирование компонентов МИТП RU.СНАБ.80066-06 01 01 при подготовке и настройке технологической платформы МИТП-Ц, а также во время эксплуатации;
- 3) автоматизированная проверка корректности развертывания компонентов за счет выполнения тестов для развернутых системных сервисов МИТП-Ц.

Развертывание МИТП-Ц производится из установочного пакета МИТП RU.СНАБ.80066-06 01, который содержит данный компонент и готовые к установке пакеты системных компонентов, включая:

- компонент хранения знаний RU.СНАБ.80066-06 01 17;
- компонент диалога поддержки принятия решений RU.СНАБ.80066-06 01 18;
- компонент разбора скрипта EasyFlow RU.СНАБ.80066-06 01 19;
- компонент интерпретации WF RU.СНАБ.80066-06 01 20;
- компонент взаимодействия с пользователем RU.СНАБ.80066-06 01 21;
- компонент серверной визуализации RU.СНАБ.80066-06 01 22;
- компонент событийного взаимодействия RU.СНАБ.80066-06 01 23;
- компонент мониторинга RU.СНАБ.80066-06 01 24;
- компонент контроля доступа RU.СНАБ.80066-06 01 26;
- компонент обеспечения доступа к инфраструктуре RU.СНАБ.80066-06 01 27;
- компонент планирования исполнения WF RU.СНАБ.80066-06 01 28;
- компонент исполнения WF RU.СНАБ.80066-06 01 29;
- компонент хранения профилей исполнения WF RU.СНАБ.80066-06 01 32;
- компонент-база ресурсов RU.СНАБ.80066-06 01 33;
- компонент учета использования ресурсов RU.СНАБ.80066-06 01 34;
- компонент-база пакетов RU.СНАБ.80066-06 01 35;
- компонент хранения данных RU.СНАБ.80066-06 01 37;
- компонент информационного портала RU.СНАБ.80066-06 01 31;
- компонент доступа к вычислительным ресурсам RU.СНАБ.80066-06 01 38.

RU.СНАБ.80066-06 31 04Ошибка! Источник ссылки не найден.

Для развертывания компонентов МИТП-Ц необходима вычислительная система под управлением ОС Windows (XP и выше), с установленной средой Silverlight 4.0, или Linux (с ядром 2.6.22 и выше), с установленной средой Mono Framework с поддержкой библиотек .NET 2.0 и выше (рекомендуется версия Mono Framework 2.6 или выше). Для корректного функционирования необходимо наличие установленного web-сервера с поддержкой технологии ASP .NET WebServices, WCF, Silverlight и удаленного развертывания сервисов (с использованием технологии WebDeploy). Примером web-сервера, соответствующего предъявленным требованиям, может служить Microsoft IIS версии 7.0 или выше.

Дополнительно для функционирования МИТП-Ц должен быть установлен сервер баз данных: MongoDB версии 1.6.5. В ходе установки и настройки используются стандартные конфигурации указанных программных средств, не требующие отдельной настройки. После установки необходимо осуществить запуск сервера баз данных для локального использования (localhost).

СУБД MongoDB используется компонентами CLAVIRE/Ginger RU.СНАБ.80066-06 01 21 – для хранения данных о пользовательских проектах; CLAVIRE/Eventing RU.СНАБ.80066-06 01 23 – для журналирования произошедших в системе событий; CLAVIRE/Monitoring RU.СНАБ.80066-06 01 24 – в качестве хранилища актуальных данных о платформе; CLAVIRE/GateKeeper RU.СНАБ.80066-06 01 26 – для хранения учетных данных пользователей; CLAVIRE/InfraAccess RU.СНАБ.80066-06 01 27 – для хранения данных о зарегистрированных компонентах; CLAVIRE/Provenance RU.СНАБ.80066-06 01 32 – для хранения профилей исполнения композитных приложений; CLAVIRE/Billing RU.СНАБ.80066-06 01 34 – для хранения пользовательских счетов, тарифов и истории операций; CLAVIRE/Storage RU.СНАБ.80066-06 01 37 – для хранения сервисной информации, используемой центральным модулем хранения данных, а также для хранения метаданных, соответствующей объектам хранения.

Для работы компонента информационного портала RU.СНАБ.80066-06 01 31 требуется установка СУБД MySQL (версии 5.0 или выше) и поддержка web-сервером интерпретатора языка PHP (версии 5.2 или выше). Для работы компонента хранения знаний RU.СНАБ.80066-06 01 17 требуется установка СУБД Microsoft SQLServer Compact Edition (версии 3.5 или выше). Также должен быть установлен web-сервер Glassfish версии 3.0.1, обеспечивающий поддержку технологии WebServices, необходимой для функционирования варианта реализации хранилища онтологической структуры RunLib. Кроме того, на ту же вычислительную систему должен быть установлен интерпретатор

RU.СНАБ.80066-06 31 04 **Ошибка! Источник ссылки не найден.**

онтологической структуры Pellet версии 2.2.2, необходимый для функционирования хранилища знаний.

2.2. Необходимые технические средства управляющей подсистемы МИТП-Ц

Компоненты МИТП-Ц функционируют на вычислительной системе – серверной ЭВМ со следующими минимальными характеристиками:

- тип процессоров: Intel-совместимый;
- количество ядер – не менее 4;
- количество процессоров – не менее 2;
- тактовая частота каждого процессора – не ниже 2.0 ГГц;
- оперативная память (на ядро) – не менее 2.0 ГБ;
- дисковая подсистема – не менее 5×250 ГБ RAID5;
- пропускная способность сетевых интерфейсов – не менее 1 Гбит/с.

Для взаимодействия с другими модулями системы требуется наличие выхода в Интернет или локальную сеть (если web-сервисы других подсистем доступны из локальной сети) с соответствующей поддержкой со стороны оборудования.

Для функционирования компонента развертывания и конфигурирования необходима рабочая станция с видеоадаптером и дисплеем, способным отображать WPF-приложение с размером окна 800×600 пикселей, со следующими минимальными характеристиками:

- архитектура процессора – x86, x86_64, IA64;
- объем оперативной памяти – 1 ГБ;
- объем свободного пространства на жестком диске – 1 ГБ;
- тактовая частота процессора – 1 ГГц.

В целях повышения производительности и реактивности МИТП-Ц отдельные компоненты могут функционировать на разных вычислительных системах в рамках общей локальной сети центра компетенции.

2.3. Необходимые технические средства подсистемы вычислительной инфраструктуры МИТП-Ц

В состав комплекса технических средств подсистемы вычислительной инфраструктуры МИТП-Ц входят как собственные ресурсы центра компетенции, так и ресурсы внешних провайдеров, например, в составе глобальных сред распределенных вычислений.

RU.СНАБ.80066-06 31 04 **Ошибка! Источник ссылки не найден.**

1. *Вычислительные кластеры.* Доступные высокопроизводительные ресурсы центра компетенции и внешних провайдеров; предназначены для установки (в т.ч. автоматической) и последующего использования прикладных сервисов МИТП-Ц. Ввиду того что МИТП-Ц предоставляет возможность унифицированной работы с ресурсами, обладающими различными классами архитектур, ключевыми требованиями к таким ресурсам являются возможность доступа по стандартным сетевым протоколам в рамках корпоративной сети, программная совместимость с прикладными пакетами, а также возможность использования стандартных средств управления ресурсами.
2. *Корпоративная или глобальная облачная инфраструктура.* Виртуальная вычислительная инфраструктура, конфигурируемая по запросу со стороны МИТП-Ц или пользователя системы (при организации доступа к уже сконфигурированным виртуальным ресурсам). Со стороны МИТП-Ц работа с облачной инфраструктурой реализуется не только на уровне абстрактных прикладных сервисов (реализуются подбор и конфигурация существующих статических ресурсов), но и на уровне абстрактных вычислительных ресурсов (осуществляется динамическая конфигурация ресурса в соответствии с предъявляемыми требованиями).
3. *Грид-инфраструктура первого поколения.* Данный класс ресурсов реализует концепцию виртуальных организаций. При этом специфика доступа к таким ресурсам определяется технологическими особенностями сервисной среды Грид, а также высокой изменчивостью структуры и характеристик этих ресурсов. Тем не менее распределенный характер и высокая суммарная производительность позволяют эффективно задействовать данный класс ресурсов при решении ряда вычислительных задач.
4. *Прочие виды корпоративных ресурсов центра компетенции* (рабочие станции, серверные ЭВМ и пр.). Интеграция широкого спектра разнородных ресурсов позволяет сформировать инфраструктуру, обеспечивающую исполнение заданий, оптимизированных для различных архитектур. Минимальные требования к корпоративным ресурсам для использования в МИТП-Ц:
 - архитектура: SMP, MPP, GPGPU, СВЕА;
 - тип процессоров: Intel-совместимый;
 - количество ядер – не менее 4;
 - количество процессоров – не менее 1;
 - количество вычислительных узлов – не менее 1;

RU.СНАБ.80066-06 31 04 **Ошибка! Источник ссылки не найден.**

- тактовая частота каждого процессора – не ниже 2.0 ГГц;
- оперативная память (на ядро) – не менее 1.0 ГБ;
- дисковая подсистема – не менее 250 ГБ на узел;
- системы управления Torque, Ganglia;
- операционные системы: Windows, Linux.

5. *Специализированные хранилища данных.* МИТП-Ц обеспечивает унифицированный доступ как к локальным, так и к распределенным хранилищам и источникам данных при условии их нахождения в локальной сети центра компетенции совместно с управляющей подсистемой МИТП-Ц. Использование инфраструктуры внешних провайдеров для хранения данных МИТП-Ц нецелесообразно по соображениям: (а) безопасности и (б) надежности.

3. ОПИСАНИЕ ЗАДАЧИ

3.1. Определение задачи

Основной задачей МИТП-Ц является обеспечение и интеллектуальная поддержка работы пользователя с облачными сервисами и композитными приложениями в распределенной предметно-ориентированной среде центра компетенции. Организация процесса создания и исполнения композитного приложения под управлением МИТП-Ц в рамках концепции iPSE сводится к последовательной формализации наборов описаний в терминах потоков заданий (workflow, WF). На первом этапе процесса проектирования композитного приложения создается мета-WF (MWF). Пользователь может осуществлять выбор классов сервисов, которые доступны в облачной среде, и уточнять их по мере ввода дополнительной информации. Указанные пользователем классы сервисов будут использоваться на следующем этапе для подбора конкретных сервисов. На основе MWF создается поток заданий, в котором уже зафиксированы конкретные реализации вычислительных сервисов, однако еще ничего не известно об условиях их выполнения (AWF). Следующим этапом процесса проектирования является построение расписания и создание сценария выполнения в терминах конкретного WF (CWF), который представляет собой поток заданий с полностью определенными блоками. Для блоков действий указаны сервисы и узлы для исполнения, а для блоков данных – конкретные местоположения необходимых данных. Более полное описание процесса создания и исполнения композитного приложения, общего для всех технологических платформ в составе МИТП CLAVIRE приведено в документе RU.СНАБ.80066-06 31 01.

RU.СНАБ.80066-06 31 04 **Ошибка! Источник ссылки не найден.**

Применительно к МИТП-Ц процесс создания распределенных корпоративных программных приложений определяется следующим алгоритмом (рис. 3.1).

1. Пользователь авторизуется в МИТП-Ц через предметно-ориентированный портал центра компетенции (используя персональный сертификат МИТП-Ц), что дает ему возможность доступа к соответствующим ресурсам и сервисам.
2. Пользователь МИТП-Ц через соответствующий web-интерфейс может выбрать конкретные сервисы или шаблоны композитных приложений в форме WF, а также получить (при необходимости) доступ к технической и эксплуатационной документации. За подбор, каталогизацию и аннотирование сервисов отвечают эксперты центра компетенции.
3. Выбрав необходимые сервисы, пользователь средствами МИТП-Ц конструирует соответствующее композитное приложение в форме описания WF. При этом может использоваться как графическая, так и текстовая форма представления композитных приложений.
4. В ряде случаев п. 2 и 3 могут быть выполнены в автоматическом (полуавтоматическом) режиме с использованием системы интеллектуальной поддержки выбора сервисов и конструирования WF в предметной области центра компетенции. Использование такой системы не является обязательным и определяется общим уровнем предметных знаний и задачами, решаемыми пользователем МИТП-Ц.
5. Для подготовленного описания композитного приложения пользователь конфигурирует условия вычислений: определяет требуемые параметры WF, редактирует (при необходимости) его описание, готовит и загружает в хранилище МИТП-Ц входные данные для расчетов. В ряде случаев такие данные могут предоставляться самим центром компетенции (через соответствующий раздел хранилища).
6. Пользователь определяет режим исполнения задачи в МИТП-Ц (утверждает предлагаемые варианты) в соответствии с требованиями к временным характеристикам расчета. При этом пользователю предлагаются различные тарифные варианты, в том числе связанные с привлечением ресурсов внешних провайдеров сред распределенных вычислений.

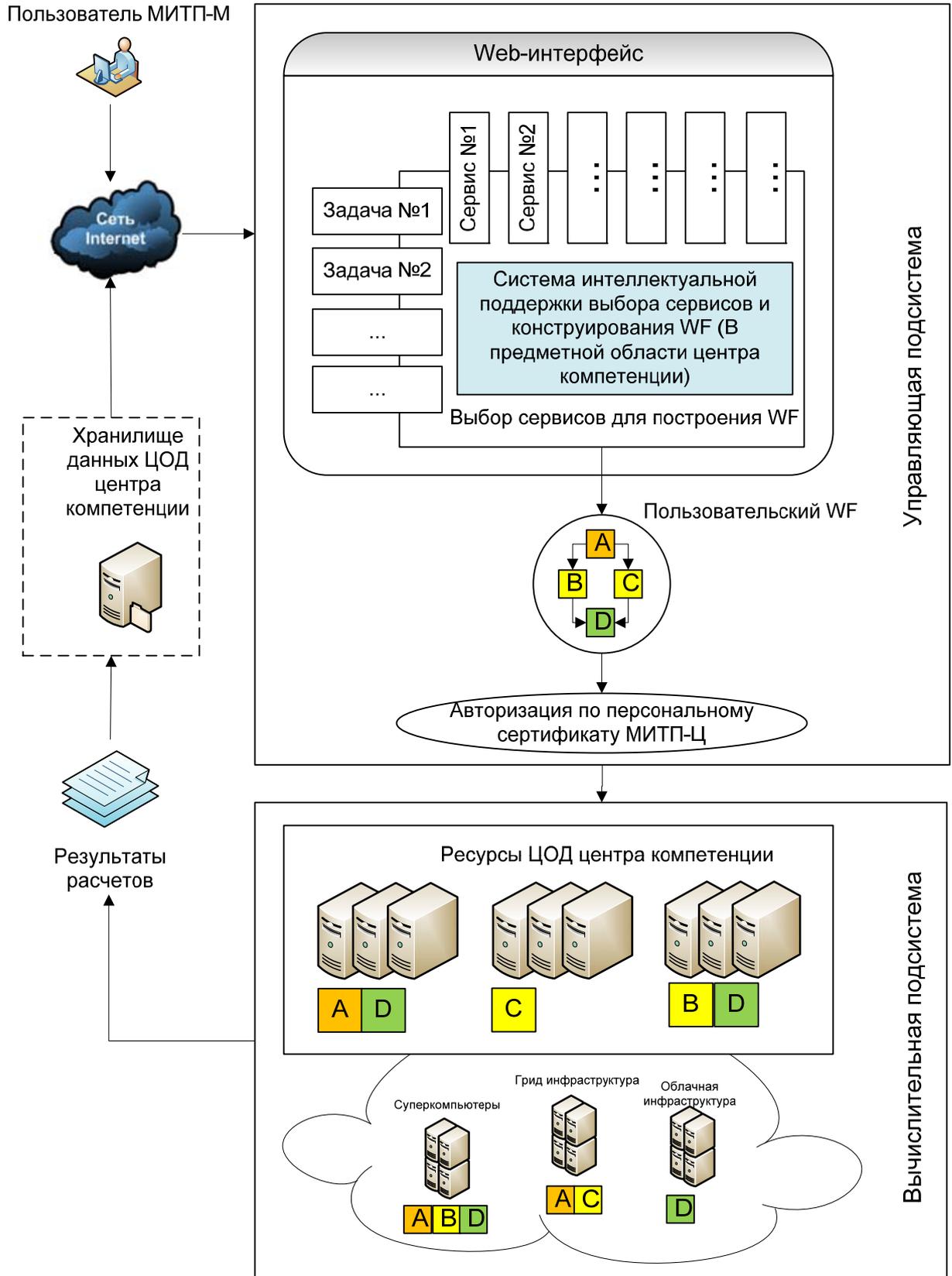


Рисунок 3.1 Принцип функционирования МИТП-Ц

RU.СНАБ.80066-06 31 04 **Ошибка! Источник ссылки не найден.**

7. Пользователь запускает задачу на выполнение в среде МИТП-Ц. Использование вычислительных ресурсов и сервисов производится с учетом единого сертификата МИТП-Ц, с которым в средах внешних провайдеров распределенных вычислений ассоциированы сертификаты «суперпользователей» МИТП-Ц. Для Грид-систем это выполняется путем ассоциирования сертификата МИТП-Ц с персональным сертификатом данного пользователя в Грид.
8. Пользователь (при необходимости) осуществляет мониторинг процесса исполнения (в форме динамического отображения WF); при этом прогнозируется (в вероятностной форме) время завершения вычислений. Пользователь может на время расчетов завершить рабочую сессию с МИТП (закрыть интерфейс доступа к МИТП-Ц) и начать ее только при необходимости использования результатов (необязательно в тот момент, когда расчеты завершены).
9. Когда расчет задачи завершен, результаты помещаются в соответствующее хранилище данных МИТП-Ц центра компетенции; пользователю отправляется уведомление. Пользователь может получить доступ к результатам расчетов (в форме выдачи результатов соответствующих пакетов в составе сервисов) через интерфейс МИТП-Ц. МИТП-Ц также обеспечивает работу специализированных средств визуализации (в рамках задач центра компетенции); при этом они могут быть доступны как через Интернет, так и непосредственно на пользовательском компьютере.

Таким образом, основная задача МИТП-Ц сводится к технологическому обеспечению операций алгоритма.

3.2 Методы решения задачи

В данном разделе рассматриваются основные методы решения задачи раздела 3.1, характерные для МИТП-Ц. Они предназначены для:

- 1) унифицированного описания прикладных пакетов с целью организации единообразного доступа к ним в форме облачных сервисов;
- 2) описания структуры композитных приложений, состоящих из нескольких взаимодействующих сервисов в распределенной среде центра компетенции;
- 3) организации всего процесса исполнения композитного приложения под управлением МИТП-Ц.

RU.СНАБ.80066-06 31 04 **Ошибка! Источник ссылки не найден.**

- 4) обеспечения интеллектуальной поддержки пользователей при выборе сервисов и конструировании композитных приложений.

3.2.1. Унифицированное описание прикладных пакетов в МИТП-Ц

В МИТП-Ц простейшая форма WF представляет собой описание исполнения одного вычислительного пакета на ресурсе корпоративной вычислительной среды с загрузкой входных данных и получением выходных. Однако унифицированное описание этого действия осложнено тем, что разные вычислительные пакеты используют свою стратегию работы с данными (использование конфигурационного файла, командной строки аргументов, переменных окружения, проектов, хранящихся в структуре директорий и файлов). Ситуация осложняется требованием единообразных принципов работы с одним и тем же пакетом, установленным на ресурсах с различными операционными системами, средами управления и исполнения и пр.

В МИТП-Ц для решения задачи унификации описаний пакетов использован предметно-ориентированный язык (Domain Specific Language, DSL) EasyPackage, позволяющий описывать пакеты в наглядной форме, понятной специалистам-предметникам, и поддающийся программной обработке. EasyPackage разработан на основе реализации языка Ruby (IronRuby), он является интерпретируемым со строгой динамической типизацией и явным приведением типов. Его базовые элементы идентичны элементам Ruby.

Описание пакета представляет собой один или несколько текстовых файлов. Оно использует следующие понятия: пакет, входной/выходной параметр, входной/выходной файл, режим запуска. *Пакет* – это исполняемое приложение, запускаемое в пакетном режиме (модель IPO – Input–Process–Output), которое принимает на входе определенный набор файлов, параметров командной строки, переменных окружения и других источников данных, а на выходе генерирует набор выходных файлов. *Параметр пакета* – это элемент данных, имеющий имя, тип и значение. Параметр может быть входным или выходным, а также может быть параметром исполнения. Тип параметра может быть одним из базовых: строка, логический тип, число с плавающей точкой, перечислимый тип, целое число, список. Режим запуска характеризуется набором используемых в нем параметров.

Структура описания пакета состоит из раздела объявления расширений, общего описания пакета, секционного описания входных и выходных данных пакета (секции *inputs* и *outputs*), описания параметров исполнения (рис. 3.2). Раздел объявления

RU.СНАБ.80066-06 31 04 **Ошибка! Источник ссылки не найден.**

расширений предназначен для определения процедур, позволяющих расширить функциональные возможности базовой библиотеки языка. Общее описание пакета включает в себя набор полей, несущих общую информацию о пакете: имя, версия, лицензия, поставщик и т.д. (строки 1–6). Раздел секционного описания содержит определение входных и выходных параметров и файлов. Параметры характеризуются следующим набором полей: тип, значение по умолчанию, процедура проверки значения параметра на корректность (например, параметр в строках 15–21). Параметры могут быть вычислимыми (строки 22–27), тогда для них указывается процедура вычисления из рабочего контекста – *evaluator* (строка 26).

```

1  name "TESTP"
2  display_as "Testp"
3  vendor "SPbSU ITMO"
4  url "http://escience.ifmo.ru"
5  license "GPLv3"
6  description "Simple package example"
7  inputs {
8    raw file {
9      name "inf"
10     filename "arg.txt"
11     place "/"
12     extractor IntegerFileExtractor.new("in")
13     assembler ObjectToSAssembler.new("in")
14   }
15   meta param {
16     name "in"
17     required
18     type int
19     validator lambda { |val, ctx| val > 0 and val < 10000 }
20     validation_error_msg "num have to be in [0; 10000]"
21   }
22   meta param {
23     name "abs_plus_3"
24     required
25     type int
26     evaluator { |ctx| ctx.in.abs + 3 }
27   }
28   cmdline lambda { |ctx| "{0} arg.txt out.txt" }
29 }
30 outputs {
31   auto file {
32     name "output_num"
33     required
34     filename "out.txt"
35     place "/"
36     extractor IntegerFileExtractor.new("out")
37   }
38   auto param {
39     name "out"
40     required
41     display_as "Output number"
42     type int
43   }
44 }
45 prepare_package

```

Рисунок 3.2 – Фрагмент описания прикладного пакета на языке EasyPackage

Контекст работы представляет собой набор уже вычисленных значений параметров (*ctx*). Файловые параметры дополнительно имеют следующий набор полей (строки 8–14): имя файла, путь до файла, процедура извлечения данных из файла (*extractor*), процедура сборки файла (*assembler*). Процедура сборки файла позволяет создавать входной файл, основываясь на значениях входных параметров. На практике используются стандартные процедуры, например, сборка файла по шаблону (библиотека ERB). Процедура извлечения данных из файла, как правило, применяется для выходных файлов пакета с целью определения значений выходных параметров (строка 12). Базовый набор процедур извлечения значений из файлов и их сборки из параметров можно дополнять за счет написания своих процедур в секции расширений. Последним в файле описания является раздел параметров исполнения, который позволяет при работе с пакетом не учитывать неоднородность ресурсов (различных ОС, архитектур). К параметрам данного раздела относятся: скрипт запуска пакета (точнее, процедура его сборки), командная строка, переменные окружения.

Таким образом, описание на языке EasyPackage позволяет не только задать правила обращения к конкретному пакету в распределенной вычислительной среде, но и корректно интерпретировать его входные и выходные данные (посредством процедур *extractor* и *assembler*). Это обеспечивает совместимость (по данным) пакетов различных разработчиков в составе WF.

Подробно особенности языка EasyPackage изложены в документе RU.СНАБ.80066-06 33 01.

3.2.2. Унифицированное описание композитных приложений в МИТП-Ц

Описание композитных приложений, состоящих из нескольких прикладных пакетов, требует определения не только правил работы с пакетами (см. раздел 3.2.1), но и структуры взаимодействия между ними. Специализированный язык EasyFlow, поддерживаемый МИТП-Ц, позволяет упростить процедуру задания композитных приложений. Он предоставляет конечному пользователю гибкие возможности по заданию различных форм WF, в рамках которых выполняются различные прикладные пакеты, происходит генерация выходных данных, их получение, конвертация и обработка.

Характерной чертой языка является полное абстрагирование от особенностей распределенной вычислительной среды, в которой работает пользователь. Фактически EasyFlow – это высокоуровневый язык описания AWF. Такой подход позволяет описывать

RU.СНАБ.80066-06 31 04 **Ошибка! Источник ссылки не найден.**

саму решаемую задачу, а не способ ее исполнения на конкретной вычислительной архитектуре. На рис. 3.3 приведен пример описания простого AWF, представляющего собой скрипт. Тело скрипта состоит из описания вызовов прикладных пакетов – *шагов*, которые задаются с помощью директивы *step* и представляют собой узлы графа WF. Для описания каждого шага необходимо задать его имя (в примере это *Step1*, *Step2*, *Step3*), название запускаемого пакета (*EmptyPackage*, *Package1* и *Package2*) и перечень предметных параметров этого пакета (см. раздел 3.2.1).

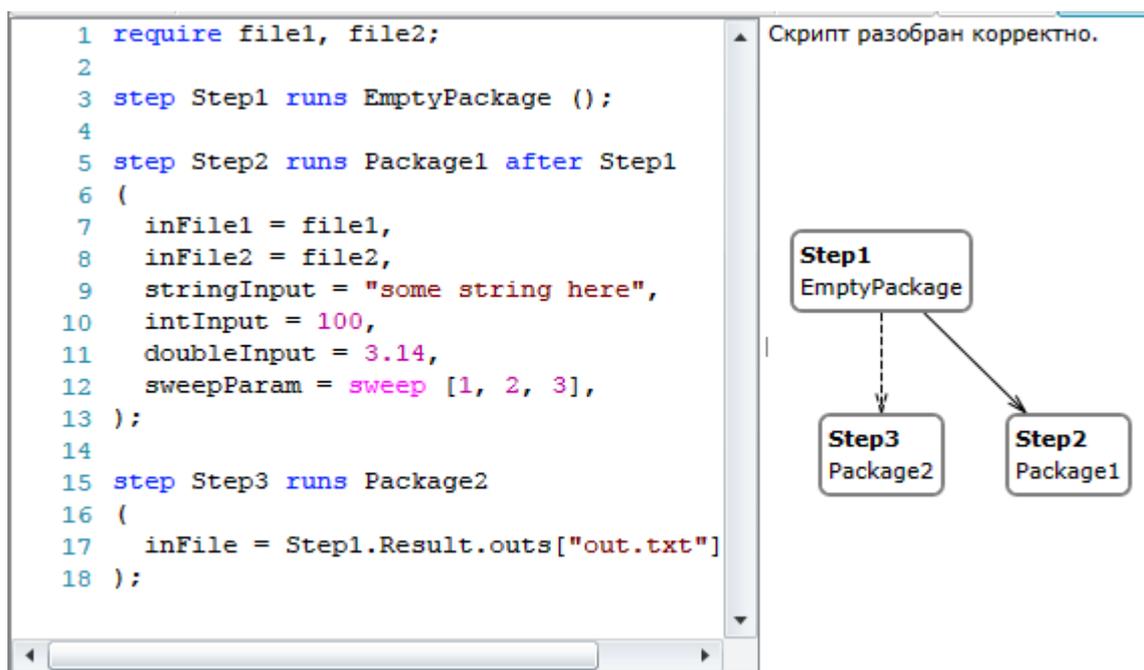


Рисунок 3.3 – Пример описания композитного приложения на языке EasyFlow и его графическое представление в МИТП-Ц

Язык EasyFlow позволяет задавать параметры для следующих типов данных: целое число, строка, число с плавающей точкой, список, структура, указание на использование файла (см. описание шага *Step3*).

Большинство прикладных пакетов помимо параметров принимает и генерирует входные и выходные файлы, поэтому в EasyFlow предусмотрена поддержка работы с файлами. Их задание в скрипте представляет собой лишь абстрактное указание с помощью директивы *require*, что освобождает пользователя от необходимости указания абсолютных путей к файлам. В этой директиве через запятую перечислены файловые переменные, которые могут быть указаны в качестве значений параметров при описании шага (см. параметры *inFile1* и *inFile2* в описании шага *Step2*). В рамках одного скрипта директива требования файлов может появляться неограниченное число раз.

RU.СНАБ.80066-06 31 04 **Ошибка! Источник ссылки не найден.**

Так как WF представляет собой ориентированный граф, в EasyFlow введены механизмы определения порядка выполнения шагов, позволяющие организовать его структуру: зависимости по управлению и зависимости по данным.

Зависимости по управлению представляют собой явные указания на то, что один шаг должен начать свое исполнение после завершения другого. Это делается с помощью директивы *after* (см. рис. 3.3, шаг *Step2*).

Зависимости по данным представляют собой неявные указания на зависимости между шагами, которые анализируются при интерпретации скрипта EasyFlow. Они выражаются в том, что некоторые шаги могут использовать данные других шагов, что неявно влияет на последовательность их запуска. Такие зависимости могут присутствовать в описываемом WF одновременно с зависимостями по управлению, что позволяет очень гибко настраивать порядок выполнения шагов. Пример зависимостей по данным содержится в описании шага *Step3* (строка 17), где указано, что в качестве входного файла используется файл *out.txt*, полученный в результате выполнения шага *Step1*.

Еще одной полезной возможностью EasyFlow является автоматическое варьирование параметров (*parameter sweep*). Такая задача часто возникает, когда необходимо запустить один и тот же вычислительный пакет, закрепив одни и варьируя другие параметры. Для этого в язык введена директива *sweep*, которая принимает список параметров для варьирования и из одного шага создает *N* шагов, где *N* соответствует числу элементов в декартовом произведении списков варьирования для различных параметров.

Пример варьирования параметра приведен на рис. 3.3 (строка 12). В этом примере будут запущены три шага *Step2* с параметром *sweepParam*, равным соответственно единице, двум и трем при прочих зафиксированных параметрах.

Таким образом, WF, описанные на языке EasyFlow, полностью независимы от конкретной архитектуры вычислений и хранения данных, что позволяет пользователям распределенной среды беспрепятственно обмениваться ими и запускать их на различных вычислительных ресурсах.

Подробно особенности языка EasyFlow изложены в документе RU.СНАБ.80066-06 33 02.

RU.СНАБ.80066-06 31 04 **Ошибка! Источник ссылки не найден.**

3.2.3. Организация процесса исполнения композитного приложения в МИТП-Ц

Рассмотренные выше (разделы 3.2.1–3.2.2) подходы к описанию пакетов, композитных приложений и интерфейсов работы с ними позволяют организовать процесс исполнения приложения в рамках МИТП-Ц. Композитное приложение представляется в виде скрипта описания WF на языке EasyFlow, который может быть параметризован набором входных параметров и файлов, а также параметров исполнения, т.е. один и тот же WF может быть исполнен для разного набора входных данных, а также в различных условиях исполнения. За разбор скрипта WF и за исполнение WF в целом отвечает компонент интерпретации WF CLAVIRE/FlowSystem RU.СНАБ.80066-06 01 20. После получения скрипта он «разбирает» его и преобразует во внутреннее представление (предварительно проверив его корректность). Обработка представления WF производится непрерывно, согласно событийной модели функционирования, т.е. интерпретация WF происходит в рамках цикла обработки поступающих событий. При запуске отдельной задачи происходит интерпретация параметров узла WF и формируется описание запуска задачи, после чего сформированное описание передается в очередь компоненту исполнения WF CLAVIRE/Executor RU.СНАБ.80066-06 01 29. Далее компонент CLAVIRE/Executor подготавливает данные для пакета и производит запуск пакета в среде Грид, после чего обрабатывает выходные данные с помощью компонента хранения данных CLAVIRE/Storage RU.СНАБ.80066-06 01 25.

Для подготовки пакета к запуску и обработки его результатов используется компонент CLAVIRE/PackageBase RU.СНАБ.80066-06 01 35, который позволяет сопоставить абстрактные и фактические правила работы с каждым пакетом в составе WF. На рис. 3.4(а) приведена структура CLAVIRE/PackageBase (см. подробнее документ RU.СНАБ.80066-06 13 35). Он состоит из интерфейсной библиотеки и репозитория пакетов. Описание пакета в такой схеме хранится в виде файла со скриптом EasyPackage в репозитории. При этом системные модули для работы с данным описанием получают скрипты и сопутствующие файлы и интерпретируют их на своей стороне, используя лишь необходимую им информацию. Данный подход выгодно отличается от применения централизованного хранилища информации о пакетах, построенного на сервисно-ориентированной модели, так как позволяет легко масштабировать систему на большее количество пользователей за счет перераспределения нагрузки.

RU.СНАБ.80066-06 31 04 **Ошибка! Источник ссылки не найден.**

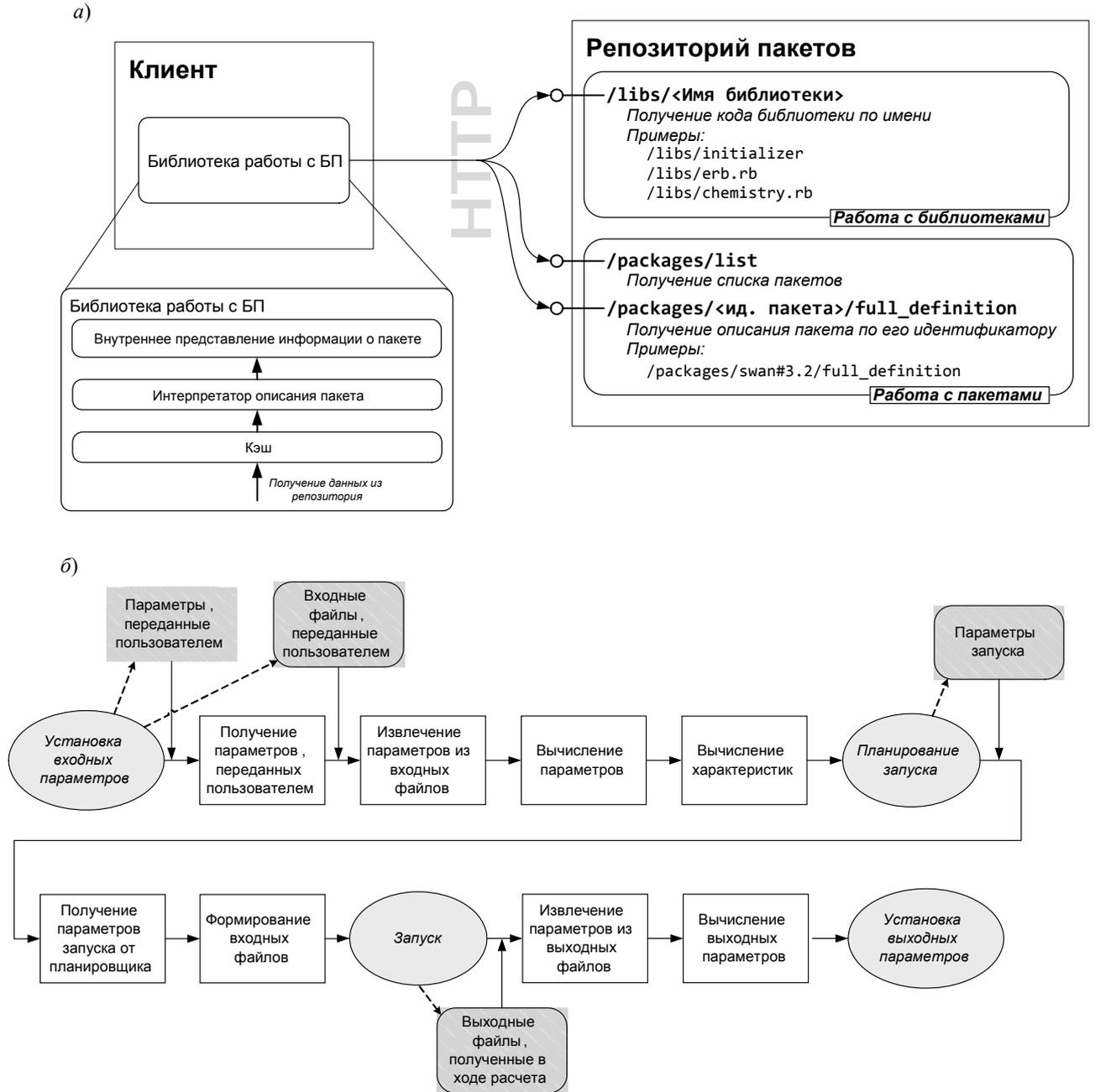


Рисунок 3.4 – Интерпретация и исполнение WF: (а) структура CLAVIRE/PackageBase ; (б) процесс исполнения композитного приложения под управлением МИТП-Ц

На рис. 3.4(б) представлен сценарий исполнения композитного приложения под управлением МИТП-Ц с использованием подходов, описанных в разделах 3.2.1–3.2.2. Характеристиками на данной схеме считаются вычисляемые параметры, необходимые только для планирования запуска в распределенной среде. После получения и обработки результатов данные о завершении работы WF передаются обратно в компонент CLAVIRE/FlowSystem RU.СНАБ.80066-06 01 20, там они становятся доступными

RU.СНАБ.80066-06 31 04 **Ошибка! Источник ссылки не найден.**

пользователю (через соответствующий интерфейс человеко-компьютерного взаимодействия CLAVIRE/Ginger RU.СНАБ.80066-06 01 21).

Таким образом, рассмотренные в разделах 3.2.1–3.2.3 методы и технологии обеспечивают решение основной задачи МИТП-Ц – поддержку разработки и исполнения прикладных сервисов и приложений на их основе на ресурсах информационно-программных предметно-ориентированных центров компетенции – набора программно-аппаратных комплексов, принадлежащих одному владельцу, предназначенных для решения специфических классов задач предметной области, с обеспечением доступа пользователей к ним через сети общего назначения (Интернет).

3.2.4. Интеллектуальная поддержка создания композитных приложений в МИТП-Ц

Для интеллектуальной поддержки выбора сервисов и разработки композитных приложений в МИТП-Ц используется система поддержки принятия решений на основе онтологического описания предметной области. Целесообразность такого подхода связана с тем, что обилие специфических задач и реализующих их методов приводит к необходимости выделить базовые понятия (концепты), позволяющие структурировать соответствующие знания. Исходя из общих принципов компьютерного моделирования в системе выделены пять базовых концептов.

1. *Объект моделирования* – ключевой концепт компьютерного моделирования, нацеленного на определение некоторых характеристик данного объекта за счет использования одной или нескольких моделей, ассоциированных с ним. Объект моделирования описывается набором характеристик, которые условно можно разделить на две группы: *определяющие* – характеристики, фиксирующие объект моделирования и отличающие его от других объектов данного класса; *переменные* – характеристики, описывающие состояние объекта моделирования и чаще всего являющиеся входными или выходными данными для процесса моделирования.

2. *Модель* – служит для получения знаний об объекте моделирования. Представляет собой некоторую математическую или логическую структуру, отражающую заданный набор характеристик исследуемого объекта. Для работы с моделью могут быть применены различные методы моделирования, обеспечивающие, например, возможность получения по одним характеристикам модели других.

3. *Метод* – императивное описание процедуры работы с моделью для получения характеристик, проверки гипотезы и т.п. Именно методы чаще всего используются в

RU.SNAB.80066-06 31 04 **Ошибка! Источник ссылки не найден.**

качестве базового объекта для реализации в составе вычислительных пакетов и представляют наибольший интерес для пользователей этих пакетов. Очевидно, что может существовать несколько реализаций одного и того же метода.

4. *Реализация метода* – алгоритм, включенный в состав доступного пользователю вычислительного пакета и ассоциированный с некоторым методом.

5. *Вычислительный сервис* – вычислительный пакет, установленный на выделенном ресурсе в распределенной среде, используемый в режиме удаленного доступа.

На рис. 3.5 частично приведена структура онтологии, построенной на основе перечисленных выше концептов. Можно выделить два основных слоя: описание концептов (классов) и индивидов, реализующих концепты. Индивиды могут быть связаны отношениями, определенными на уровне концептов. Кроме того, допустимы отношения между отдельными концептами (например, отношение генерализации). В простейшем случае возможно ограничиться двумерными отношениями, что зачастую приводит к необоснованному усложнению структуры множества отношений за счет декомпозиции семантически связанных многомерных отношений на совокупность двумерных отношений.

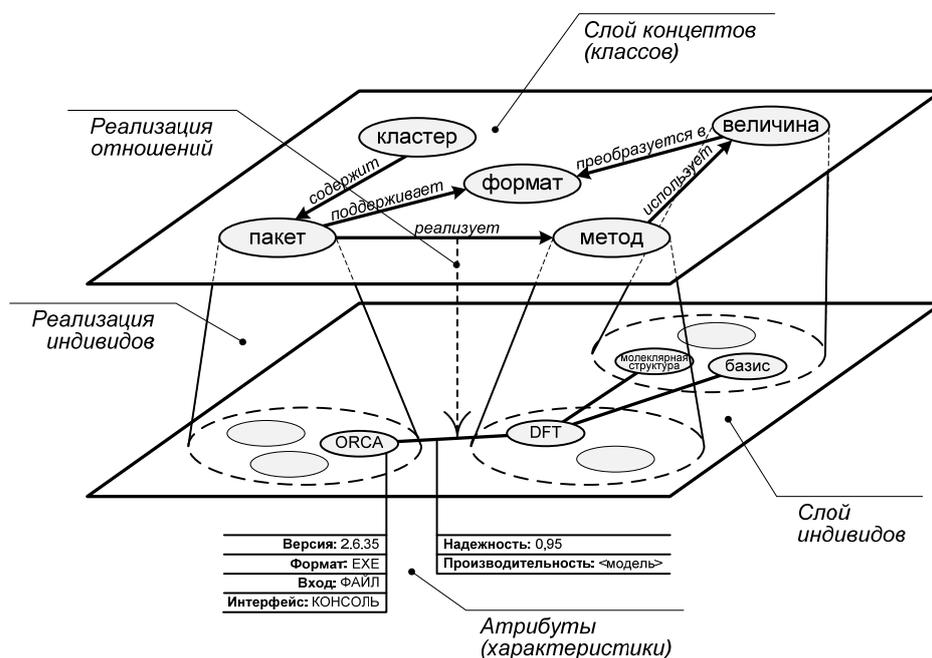


Рисунок 3.5 – Структура онтологии описания предметной области в МИП-Ц (на примере квантовой химии)

Одним из элементов онтологии являются атрибуты (характеристики) индивидов, детализирующие их описание. Кроме того, одним из возможных расширений классического подхода является ассоциация характеристик не только с индивидами (как реализациями

классов), но и со связями между ними (как реализациями классов допустимых связей). Такое расширение применимо при введении операций второго порядка (например, анализе «неидеальности» различного рода для связей, интерпретируемых как экспертные знания).

Используя логический вывод на базе онтологической структуры, возможно построить описания композитных приложений с привлечением экспертных знаний различных уровней абстракции. Это выполняется в ходе следующих действий.

1. Производится *формализация требований* пользователя к решаемой задаче: требования формируют базовый набор активных фактов, записанный в терминологии используемой базы знаний. Факты, предоставляемые пользователем, можно разделить на две допускающие пересечение группы: *определяющие* – набор фактов, на основании которого формируется базовая цепочка решения (достаточный набор для однозначного определения решаемой задачи); *дополнительные* – набор требований и ограничений, более точно определяющих специфику решаемой задачи. Базовый набор активных фактов расширяется на основании правил логического вывода, доступных системе. Полученный таким образом набор активных фактов (или несколько альтернативных наборов) используется в последующих процедурах.

2. *Фильтрация*. На основании построенного набора активных фактов производится оценка элементов онтологии в пространстве критериев качества, по которой проводится первоначальная фильтрация онтологии, позволяющая отсеивать элементы, неприменимые для решения поставленной задачи. Отфильтрованная таким образом онтология представляет собой ориентированный граф, в котором каждой вершине или ребру сопоставлено значение в пространстве критериев качества. В простейшем случае этот шаг позволяет выбрать допустимые в данной ситуации элементы онтологической структуры.

3. *Определение доступных методов*. В составе построенного графа могут быть выделены связанные подграфы, каждый из которых состоит из элементов онтологии, определяющих один из доступных методов предметной области. При этом доступность метода определяется: а) реализацией этого метода хотя бы одним из сервисов, доступных в распределенной среде; б) соответствием метода сформированному набору активных фактов (определяется на основании процедуры фильтрации (см. шаг 2)).

4. *Композиция*. На основании полученного набора доступных методов должен быть построен абстрактный сценарий решения поставленной задачи. Структурная связь методов определяется: а) комбинацией типовых схем взаимодействия (последовательный запуск, цикл, параллельный перебор, ветвление и пр.); б) соответствием зависимостей по

RU.СНАБ.80066-06 31 04 **Ошибка! Источник ссылки не найден.**

данным (входные данные для каждого из реализуемых методов должны быть либо предоставлены пользователем, либо получены в ходе выполнения предыдущих методов). Простейшим случаем можно назвать последовательный запуск сервисов, позволяющий в конечном итоге получить требуемую характеристику.

5. *Оценка.* Каждый из построенных вариантов композитного приложения должен быть подвергнут автоматическому анализу, результатом которого является интегральная оценка, позволяющая: а) провести ранжирование альтернативных вариантов решения для автоматического выбора лучшего из них (по совокупности критериев) или предоставления выбора пользователю; б) отсеять варианты решения заведомо, не удовлетворяющие заданным критериям качества. При этом входными данными для построения интегральной оценки служат оценки элементов онтологии, входящих в состав подграфов, описывающих отдельные методы, используемые в решении (см. шаг 3).

В общем случае на каждом этапе могут порождаться дополнительные варианты решения. Данная процедура проводится итеративно на каждом из концептуальных уровней базы знаний. Таким образом, строится дерево решений, структура которого определяется базовой иерархией. На заключительном этапе производится оценка каждого из листьев полученного дерева, на основании которой можно либо провести автоматический выбор оптимального (с точки зрения качества) решения, либо предоставить пользователю осмысленные и обоснованные результаты оценки с предложением осуществить самостоятельный выбор (например, установив какой-либо из критериев качества как приоритетный).

На рис. 3.6 приведен пример формирования дерева принятия решений, позволяющего выбрать вариант решения квантово-химической задачи одной точки методом Хартри–Фока с помощью пакета GAMESS (конкурирующего с пакетами ORCA и MOLPRO) – на примере предметной области «квантовая химия».

Таким образом, рассмотренный подход обеспечивает решение задачи интеллектуальной поддержки пользователя МИТП-Ц при формировании композитных приложений в заданной предметной области на основе готовых прикладных пакетов. Данный функционал реализован в программных компонентах хранения знаний CLAVIRE/iKnow RU.СНАБ.80066-06 01 17 и диалога поддержки принятия решений CLAVIRE/iTree RU.СНАБ.80066-06 01 18.

RU.CHAБ.80066-06 31 04 Ошибка! Источник ссылки не найден.

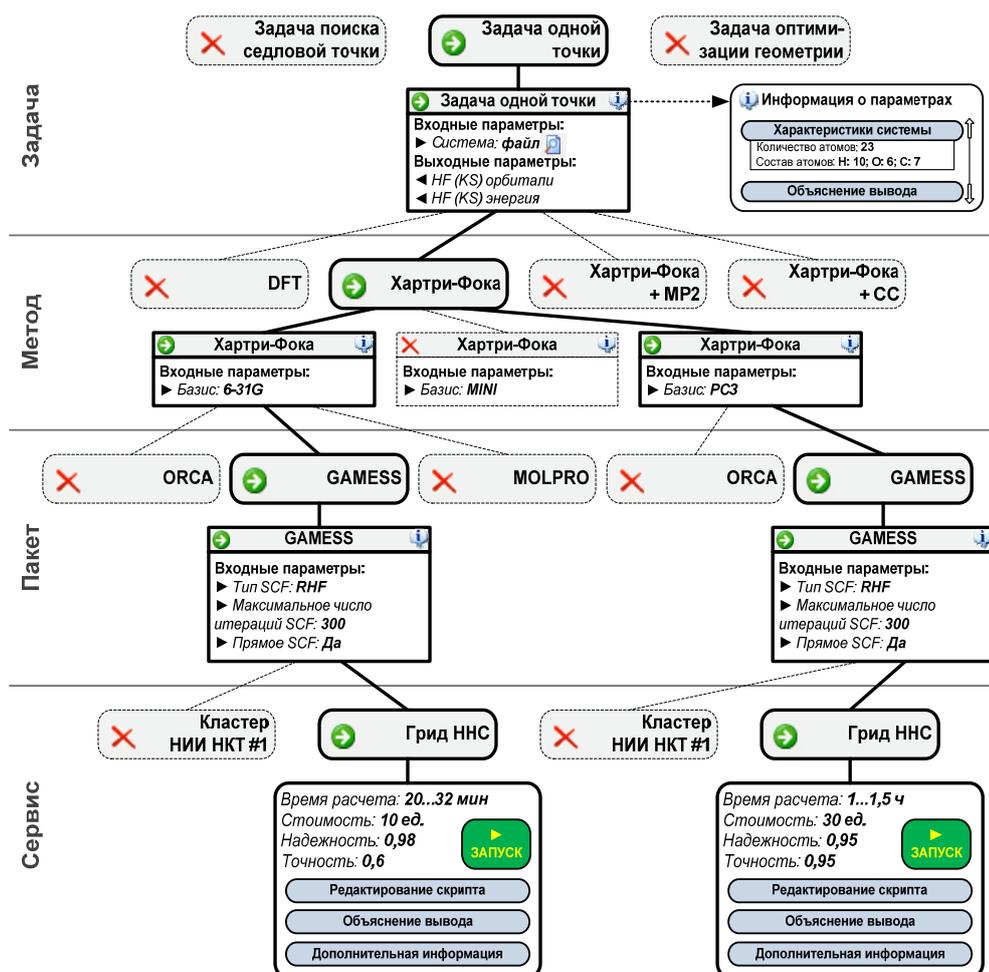


Рисунок 3.6 – Пример формирования дерева принятия решений

3.2.5. Особенности создания и управления информационно-программных предметно-ориентированных центров компетенции на основе прикладных сервисов (программ) в рамках концепции SaaS

Основные отличительные особенности создания и управления информационно-программных предметно-ориентированных центров компетенции на основе прикладных сервисов выражаются следующими пунктами.

- 1) Разработка сложных типовых предметно-ориентированных WF, опираясь на экспертные знания и многообразие разнородных прикладных пакетов. Унификация и аккумуляция в репозиторий типовых WF осуществляется с использованием возможности аннотирования, т.е. краткого пояснения решаемых задач WF, названия WF и его автора.
- 2) Создание и расширение механизма интеллектуального конструирования WF, путем дополнения базы знаний новыми данными прикладной области и обновленной информацией о инфраструктуре платформы.

RU.СНАБ.80066-06 31 04 **Ошибка! Источник ссылки не найден.**

- 3) Работа с WF опирается на два основных механизма: шаблонное использование сложных предметно-ориентированных WF; построение уникальных WF на базе интеллектуальной системы поддержки пользователя.
- 4) При описании прикладных сервисов в базе пакетов, разнородные входные/выходные данные приводятся к единому формату в рамках предметно-ориентированной области вычислений, используя для этого механизмы конвертации в РВ (т.е. extractor-ы и assembler-ы)

3.2.6. Решение типовой задачи компьютерного моделирования сложных систем с использованием МИТП-Ц

В качестве характерного композитного приложения для МИТП-Ц рассмотрим WF моделирования блокирования тромбина его прямыми ингибиторами. В ходе расчетов автоматически определяется аффинитет, с которым каждый из лигандов блокирует сайт тромбина. Актуальность выбранного примера во многом обусловлена тем, что тромбозы являются одной из наиболее частых причин смертности. В качестве структур для докинга выбрано несколько прямых ингибиторов тромбина и их ближайшие модификации. Моделирование осуществляется в два этапа, на каждом происходит расчет встраивания лигандов в сайт белка-мишени, сначала быстрыми высокопроизводительными методами докинга, после чего выполняется высокоточный расчет для кандидатов, прошедших порог аффинитета.

На первом этапе решения задачи интересующая пользователя структура белка-мишени помещается в систему и подготавливается для докинга при помощи пакета MGLTools. После этого запускается процесс скрининга лигандов, при котором происходит докинг каждого лиганда из базы данных в указанный пользователем сайт белка. Лиганды передаются из базы данных для процесса скрининга посредством скриптового приложения CоруPack. Моделирование осуществляется программой Autodock Vina. Далее отбираются лучшие кандидаты и передаются на второй этап, начинающийся с работы препроцессора QRegionSelector, который разработан специально для данного композитного приложения. Для каждого лиганда с высоким аффинитетом препроцессор определяет окружающие его атомы белка и содержащие их аминокислотные остатки. Данные остатки вместе с лигандом формируют область для высокоточного расчета. Далее запускается основной расчет (используется пакет NWChem), который решает задачу оптимизации геометрии для всей системы в целом гибридным методом QM/MM. Результатом работы композитного приложения является

RU.СНАБ.80066-06 31 04 **Ошибка! Источник ссылки не найден.**

структура смоделированного комплекса макромолекулярного белка-мишени и выбранного лиганда.

В листинге 3.1 приведен скрипт описания соответствующего композитного приложения на языке на EasyFlow.

Листинг 3.1 – Код композитного приложения моделирования блокирования тромбина

```
require receptor_pdb;

step ligand_produce runs copy_pack
();

step convert runs MGL_Tools
(
    receptor_pdb = sweep ligand_produce.Result.ligands
);

step Docking runs vina
(
    receptor = sweep convert.Result.outs["receptor.pdbqt"],
    ligand = ligand_pdbqt,
    search_space_center_x = 15,
    search_space_center_y = 8.5,
    search_space_center_z = 2,
    search_space_size_x = 6.0,
    search_space_size_y = 6.0,
    search_space_size_z = 4
);

step NWChem_prepare runs q_region_selector
(
    receptor = receptor_pdb,
    ligand_confs = Docking.Result.outs["ligand_out.pdbqt"]
);

step QMMM runs NWChem
(
    config_input      = sweep NWChem_prepare.Result.models,
    coordinates_input = sweep NWChem_prepare.Result.configs,
);
```

На рис. 3.7 приведено окно интерпретации композитного приложения в МИТП-Ц с указанием всех шагов моделирования.

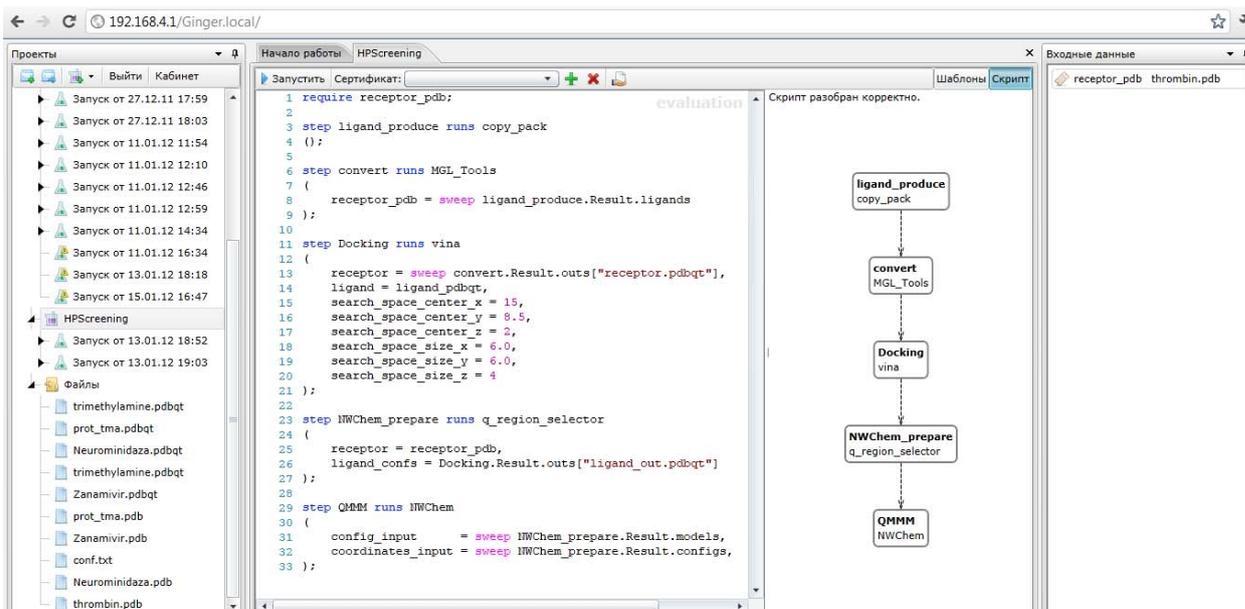


Рисунок 3.7 – Окно интерпретации композитного приложения в МИТП-Ц

В качестве результатов расчетов МИТП-Ц на рис. 3.8 приведена структура смоделированного комплекса макромолекулярного белка-мишени и выбранного лиганда. Структура комплекса выдается в файле достаточно большого размера и может быть проанализирована сторонними средствами визуализации, позволяющими анализировать и визуализировать молекулярные структуры.

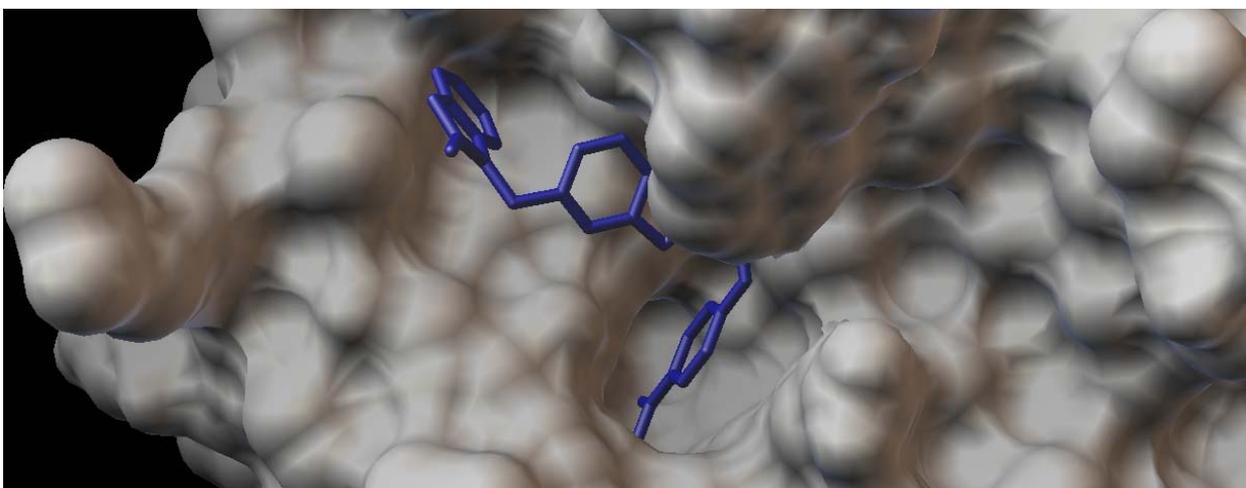


Рисунок 3.8 – Визуализация результатов расчетов – комплекс тромбина с лигандом

Подробное описание рассмотренной задачи и композитного приложения приведено в документе RU.СНАБ.80066-06 13 52.

4. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

RU.СНАБ.80066-06 31 04 **Ошибка! Источник ссылки не найден.**

Как видно из примера в разделе 3.2.5, для работы с МИТП-Ц не требуются специальные виды входных данных. Таковыми могут являться любые входные файлы, соответствующие по формату запускаемым прикладным сервисам (в текстовом, цифровом, графическом виде), а также данные, вводимые пользователем с клавиатуры по запросу сервиса, или в скрипте композитного приложения. В случае несоответствия данных условиям их использования будет выдано соответствующее системное сообщение. Для их приведения к общему формату используется технология описания на основе языка EasyPackage (раздел 3.2.1).

В качестве выходных данных МИТП-Ц предоставляет результаты расчетов, загруженные с удаленного хранилища CLAVIRE/Storage RU.СНАБ.80066-06 01 25 (в форме текстового, графического или цифрового файла, размещаемого в директории, указываемой пользователем через соответствующее диалоговое окно). Формат файла соответствует тому сервису, посредством которого был произведен расчет. Для обеспечения единого формата в целях унификации процесса передачи данных между сервисами используется технология описания на основе языка EasyPackage (раздел 3.2.1).

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

МИТП	Многопрофильная инструментально-технологическая платформа
МИТП-Ц	Технологическая платформа создания информационно-программных предметно-ориентированных центров компетенции на основе прикладных сервисов (программ) в рамках концепции SaaS (на основе CLAVIRE)
ОС	Операционная система
ПАК	Программно-аппаратный комплекс
СУБД	Система управления баз данных
ЭВМ	Электронная вычислительная машина
AaaS	Application as a Service, модель облачных вычислений
AWF	Абстрактный WF
CLAVIRE	Cloud Applications Virtual Environment, наименование МИТП
CWF	Конкретный WF
DSL	Domain Specific Language, предметно-ориентированный язык
iPSE	Intelligent Problem Solving Environment, концепция
MWF	Мета-WF
SaaS	Software as a Service, модель облачных вычислений
WF	Поток заданий, workflow

