

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

Описание применения

ЛИСТ УТВЕРЖДЕНИЯ

RU.СНАБ.80066-06 31 01-ЛУ

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

Представители
Организации-разработчика

Руководитель разработки,
профессор НИУ ИТМО

Бухановский А.В.
"28" декабря 2011 г.

Ответственный исполнитель,
с.н.с. НИУ ИТМО

Луценко А.Е.
"28" декабря 2011 г.

Нормоконтролер
ведущий инженер НИУ ИТМО

Позднякова Л.Г.
"28" декабря 2011 г.

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

УТВЕРЖДЕН
RU.СНАБ.80066-06 31 01-ЛУ

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

Описание применения

RU.СНАБ.80066-06 31 01

ЛИСТОВ 32

Инв. № подл.		Подп. и дата		Инв. № дубл.		Подп. и дата	
Взам. инв. №							

АННОТАЦИЯ

Документ содержит описание применения многопрофильной инструментально-технологической среды CLAVIRE (Cloud Applications Virtual Environment) RU.СНАБ.80066-06. Многопрофильная инструментально-технологическая платформа (МИТП) CLAVIRE предназначена для создания, исполнения и предоставления сервисов доступа к предметно-ориентированным высокопроизводительным композитным приложениям, функционирующим в облаке неоднородных вычислительных ресурсов корпоративного уровня, уровня центров компетенции, центров обработки данных, инфраструктур экстренных вычислений и систем распределенного хранения и обработки данных. МИТП CLAVIRE 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	4
1.3. Основные характеристики	5
1.4. Ограничения, накладываемые на область применения	9
2. УСЛОВИЯ ПРИМЕНЕНИЯ	9
2.1. Условия развертывания программы	9
2.2. Необходимые технические средства	12
3. ОПИСАНИЕ ЗАДАЧИ	12
3.1. Определение задачи	13
3.2. Методы решения задачи	15
3.2.1. Унифицированное описание прикладных пакетов в МИТП	16
3.2.2. Унифицированное описание композитных приложений в МИТП	19
3.2.3. Разбор и интерпретация скрипта на языке EasyFlow	21
3.2.4. Планирование и оптимизация процесса исполнения композитного приложения в распределенной среде	24
3.2.5. Организация процесса исполнения композитного приложения в МИТП	27
4. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ	30
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ	31
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ	32

1. НАЗНАЧЕНИЕ ПРОГРАММЫ

Многопрофильная инструментально-технологическая платформа (МИТП) создания и управления распределенной средой облачных вычислений CLAVIRE RU.СНАБ.80066-06 предназначена для создания, исполнения и предоставления сервисов доступа к предметно-ориентированным высокопроизводительным композитным приложениям, функционирующим в облаке неоднородных вычислительных ресурсов корпоративного уровня, уровня центров компетенции, центров обработки данных, инфраструктур экстренных вычислений и систем распределенного хранения и обработки данных.

1.1. Функциональное назначение

МИТП представляет собой комплекс программного обеспечения для разработки, настройки и эксплуатации сред распределенных вычислений, предназначенный для:

- 1) эффективного управления вычислительными, информационными и программными ресурсами распределенных неоднородных вычислительных инфраструктур в рамках модели облачных вычислений;
- 2) создания, исполнения, управления и предоставления сервисов доступа к предметно-ориентированным высокопроизводительным композитным приложениям, функционирующим на основе облака распределенных прикладных сервисов¹;
- 3) обеспечения функционирования программно-аппаратных комплексов (ПАК) поддержки инфраструктуры предметно-ориентированных облачных вычислений в различных предметных областях.

1.2. Область применения

Областью применения МИТП является создание инновационных производств на основе предоставления пользователям через Интернет высокопроизводительных предметно-ориентированных сервисов в рамках модели облачных вычислений для различных нужд науки, промышленности, бизнеса и социальной сферы, с возможностью

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 через корпоративную сеть передачи данных доступна для выполнения без использования вычислительных и программных ресурсов на стороне пользователя.

RU.СНАБ.80066-06 31 01 **Ошибка! Источник ссылки не найден.**

использования ресурсов уже существующих распределенных вычислительных инфраструктур². В частности, это включает в себя:

- 1) создание предметно-ориентированных высокопроизводительных композитных приложений корпоративного уровня в неоднородной распределенной вычислительной среде с обеспечением эффективного использования вычислительных ресурсов организации-потребителя;
- 2) создание центров обработки данных, реализующих сервисы в рамках концепции SaaS на основе распределенной облачной среды;
- 3) создание программно-аппаратной инфраструктуры центров компетенции, предоставляющих доступ к облачным продуктам – высокопроизводительным приложениям в рамках определенных предметных областей;
- 4) поддержка вычислительной, программной и информационной инфраструктуры экстренных вычислений³ для обеспечения процессов поддержки принятия решений в экстремальных ситуациях;
- 5) обеспечение процессов обработки больших объемов данных в распределенных хранилищах, включая мониторинг медиаконтента в социальных сетях и потоковую обработку данных с целью извлечения из них знаний.

1.3. Основные характеристики

МИТП CLAVIRE реализована на основе технологии iPSE (Intelligent Problem Solving Environment). Она ориентирована на развитие интеллектуальных технологий поддержки жизненного цикла проблемно-ориентированных сред распределенных вычислений и позволяет объединять распределенные вычислительные сервисы различных разработчиков в проблемно-ориентированных оболочках. Преимущества данного подхода состоят в следующем.

1. В рамках МИТП на основе технологии iPSE формализуются не только методы и вычислительные алгоритмы, но и *экспертные знания* о процессах решения конкретных задач с использованием прикладных пакетов и облачных сервисов.
2. МИТП предоставляет *единый интерфейс* взаимодействия в «облаке» для предметно-ориентированных программных пакетов, которые могут

² Грид-системы различного уровня, суперкомпьютеры с удаленным доступом.

³ Urgent Computing.

RU.СНАБ.80066-06 31 01 **Ошибка! Источник ссылки не найден.**

разрабатываться различными коллективами, могут быть написаны на разных языках и иметь различные условия распространения и использования.

3. МИТП изначально ориентирована на поддержку высокопроизводительных вычислений в *неоднородных иерархических системах* (например, состоящих из выделенных суперкомпьютеров, виртуальных машин в облаке, Грид-сред и пр.). При этом управление эффективностью вычислений выполняется на уровне реализации iPSE, изолируя пользователя от работы непосредственно с распределенными ресурсами.

МИТП состоит из ядра RU.СНАБ.80066-06 01 01, включающего набор следующих программных компонентов:

- 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;
- компонент сбора данных в социальных сетях в Интернет RU.СНАБ.80066-06 01 39.
- компонент развертывания и конфигурирования RU.СНАБ.80066-06 01 36.

RU.СНАБ.80066-06 31 01 **Ошибка! Источник ссылки не найден.**

На основе перечисленных компонентов посредством компонента развертывания и конфигурирования CLAVIRE/Deployment RU.СНАБ.80066-06 01 36 могут быть сформированы следующие технологические платформы МИТП:

МИТП-К RU.СНАБ.80066-06 01 41	– технологическая платформа создания распределенных корпоративных программных приложений для компьютерного моделирования сложных систем и поддержки принятия решений по управлению комплексными объектами;
МИТП-М RU.СНАБ.80066-06 01 42	– технологическая платформа создания межорганизационных (коллаборативных) композитных программных приложений;
МИТП-Ц RU.СНАБ.80066-06 01 42	– технологическая платформа создания информационно-программных предметно-ориентированных центров компетенции на основе прикладных сервисов (программ) в рамках концепции SaaS;
МИТП-Э RU.СНАБ.80066-06 01 43	– технологическая платформа создания и поддержки инфраструктуры экстренных вычислений (urgent computing) в распределенных средах с динамическими вычислительными ресурсами;
МИТП-Д RU.СНАБ.80066-06 01 44	– технологическая платформа потоковой обработки сверхбольших объемов данных и извлечения из них знаний на основе облачных вычислений.

Для функционирования МИТП необходим программно-аппаратный комплекс, обеспечивающий поддержку инфраструктуры предметно-ориентированных облачных вычислений на основе неоднородной распределенной вычислительной среды и предоставления услуг по управлению композитными приложениями с использованием указанной инфраструктуры, а также с целью поддержки деятельности по учету и управлению предоставлением услуг. МИТП может функционировать в распределенной вычислительной среде суммарной производительностью от 500 ТФлопс и выше, поддерживать не менее 30 прикладных сервисов и не менее 1000 пользователей одновременно.

МИТП обеспечивает следующие временные характеристики:

RU.СНАБ.80066-06 31 01 Ошибка! Источник ссылки не найден.

- реакция системы на запрос пользователя – не более 3 секунд;
- время исполнения прикладных сервисов⁴ – не более 5 минут.

Для обеспечения данных временных характеристик в составе МИТП предусмотрены средства оптимизации исполнения композитных приложений (компонент RU.СНАБ.80066-06 01 28), которые обеспечивают планирование:

- по времени исполнения сервиса в монопольном режиме;
- по времени выполнения всех задач композитного приложения;
- по трафику данных в распределенной среде;
- по времени выполнения WF в целом;
- по общему времени простоев распределенной инфраструктуры.

При этом в процессе планирования определяются (см. раздел 3.2.4):

- количество вычислительных ресурсов, обеспечивающих минимальное время исполнения прикладных сервисов и композитного приложения в целом;
- количество вычислителей, обеспечивающих наибольшее параллельное ускорение каждого из прикладных сервисов в составе композитного приложения;
- количество различных вычислительных ресурсов, используемых для выполнения одного композитного приложения.

МИТП удовлетворяет следующим требованиям к надежности:

- средняя наработка на отказ 10 тыс. процессоро-часов, не менее;
- средний срок службы 5 лет, не менее.
- среднее время восстановления работоспособного состояния после отказа, вызванного неисправностью (сбоем) МИТП должно составлять 2 часа, не более;
- время восстановления после отказа, вызванного сбоем электропитания технических средств (и/или иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, при условии соблюдения условий эксплуатации технических и программных средств 6 часов, не более.

МИТП обеспечивает соответствие технологических платформ нормативным значениям (не ниже) индексов производительности⁵ (см. таблицу 1.1).

⁴ Временем выполнения прикладного сервиса считается время, связанное со всеми операциями МИТП по подготовке данных, запуску и управлению композитным приложением в распределенной среде. Определяется как

⁵ Способы оценки индексов производительности приведены в документе RU.СНАБ.80066-06 И2 01 «Методика определения состава, конфигурирования и настройки программных и технических средств»

RU.СНАБ.80066-06 31 01 **Ошибка! Источник ссылки не найден.**

Таблица 1.1 – Нормативные значения индексов производительности (%) для технологических платформ (не ниже указанной величины)

Индекс	МИТП-К	МИТП-М	МИТП-Ц	МИТП-Э	МИТП-Д
Технологическая эффективность использования (ТЭИ)	80	75	75	80	90
Средняя пользовательская эффективность использования (ПЭИ)	*	*	90	*	*
Абсолютная эффективность технических решений (АЭТР)	25	90	*	*	*
Относительная эффективность технических решений (ОЭТР)	60	*	65	*	65

* - значение для платформы не определено.

1.4. Ограничения, накладываемые на область применения

МИТП является инструментально-технологической платформой – средством разработки и эксплуатации облачных сред. В ее состав не входят прикладные пакеты по предметным областям; они должны устанавливаться и настраиваться отдельно.

МИТП предоставляет управляющую инфраструктуру для организации и поддержки сред облачных вычислений с использованием *внешних* вычислительных ресурсов, которые предоставляются, настраиваются и конфигурируются отдельно от МИТП.

2. УСЛОВИЯ ПРИМЕНЕНИЯ

2.1. Условия развертывания программы

Установка и настройка МИТП 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;
- 2) конфигурирование компонентов МИТП при подготовке и настройке технологической платформы, а также во время эксплуатации;
- 3) автоматизированная проверка корректности развертывания компонентов за счет выполнения и проверки тестов для развернутых системных сервисов МИТП.

RU.СНАБ.80066-06 31 01 Ошибка! Источник ссылки не найден.

Развертывание МИТП производится из установочного пакета, 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 информационного портала RU.СНАБ.80066-06 01 31;
- 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 доступа к вычислительным ресурсам RU.СНАБ.80066-06 01 38;
- компонент сбора данных в социальных сетях в Интернете RU.СНАБ.80066-06 01 39.

Для развертывания компонентов МИТП 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.

RU.СНАБ.80066-06 31 01 Ошибка! Источник ссылки не найден.

Дополнительно для функционирования МИТП должен быть установлен сервер баз данных: MongoDB версии 1.6.5. В ходе установки и настройки используются стандартные конфигурации указанных программных средств, не требующие специальной модификации. 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 метаинформации, 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). СУБД MySQL используется компонентами CLAVIRE/Portal RU.СНАБ.80066-06 01 31 – для хранения служебных данных системы управления контентом (CMS), хранения контента (содержимого страниц), текстов учебных материалов и модулей, содержимого лабораторных работ и их пользовательских результатов, данных и материалов социальной сети, а также имен файлов документов; CLAVIRE/AdminTool RU.СНАБ.80066-06 01 64 – для хранения служебных данных системы управления контентом (CMS), контента (содержимого страниц), данных о настройках web-сервисов, к которым обращается данная утилита, а также для хранения промежуточных данных. К служебным данным системы управления контентом относятся данные о страницах, модулях, меню, пользователях и правах доступа к страницам.

Для работы компонента хранения знаний RU.СНАБ.80066-06 01 17 требуется установка СУБД Microsoft SQL Server Compact Edition (версии 3.5 или выше). СУБД Microsoft SQL Server Compact Edition используется компонентом CLAVIRE/iKnow RU.СНАБ.80066-06 01 17 для хранения формализованных экспертных знаний предметной

RU.СНАБ.80066-06 31 01 **Ошибка! Источник ссылки не найден.**

области. Выбор данной СУБД обусловлен наиболее полной интеграцией с платформой .NET, а также возможностью использования библиотеки Microsoft Entity Framework.

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

2.2. Необходимые технические средства

Компоненты МИТП 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 менее 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 МИТП 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 увеличения производительности и реактивности МИТП отдельные компоненты могут функционировать на разных вычислительных системах в рамках общей локальной сети.

3. ОПИСАНИЕ ЗАДАЧИ

3.1. Определение задачи

Непрерывно возрастающая сложность современных систем распределенных вычислений ограничивает возможности их широкого использования специалистами предметных областей. Потому важной задачей является развитие специализированных подходов к разработке распределенных приложений с максимальным уровнем абстрагирования от специфики их технической реализации и исполнения на вычислительных ресурсах, функционирующих в рамках различных моделей облачных вычислений.

К настоящему времени хорошо зарекомендовал себя подход, основанный на описании сложных вычислительных процессов в форме рабочих потоков, или потоков заданий (workflow, WF). Под WF понимается ориентированный граф, узлами которого, как правило, являются запуски отдельных задач, а ребрами – зависимости между ними, соответствующие операциям обмена данными и управляющими параметрами в распределенной вычислительной среде. В отличие от алгоритма, WF не определяет последовательность выполнения операций; он регламентирует только их состав и зависимости. Организация процесса создания и исполнения композитного приложения под управлением МИТП в рамках концепции iPSE сводится к последовательной формализации наборов описаний приложений в терминах WF. Верхним уровнем описания приложения является мета-WF (далее MWF). Его отдельные блоки содержат лишь для описания расчетных задач, нужных пользователю, в виде неявных указаний. Таким образом, составленный MWF представляет собой формальное описание пользовательской задачи в терминах предметной области, без указаний на условия ее реализации. Помимо описания действий и данных, необходимых для вычислений, пользователь может задавать критерии, по которым будут подбираться конкретные сервисы, ресурсы, данные (например, время выполнения или высокая надежность), а также дополнительные ограничения и параметры отдельных действий (например, требуемая точность результата).

Проектирование композитного приложения при таком задании исходных данных представляет собой процесс поэтапного уточнения MWF через стадии AWF и CWF вплоть до создания конкретных сценариев запуска сервисов в облачной среде и дальнейшего их выполнения. На первом этапе процесса проектирования композитного приложения создается MWF. Пользователь может, например, осуществлять выбор классов сервисов, которые доступны в облачной среде, и уточнять их по мере ввода дополнительной информации. Указанные пользователем классы сервисов будут использоваться на

RU.СНАБ.80066-06 31 01 **Ошибка! Источник ссылки не найден.**

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 CWF, который представляет собой поток заданий с полностью определенными блоками. Для блоков действий указаны сервисы и узлы для исполнения, а для блоков данных – конкретное местоположение необходимых данных. На рис. 3.1 представлена принципиальная схема создания и исполнения композитного приложения под управлением МИТП, общая для всех способов организации облачной среды.

Таким образом, в рамках общей схемы (см. рис. 3.1) МИТП решает прикладные задачи технологического обеспечения следующих операций:

- 1) обеспечение дружественной среды доступа пользователя к программным сервисам в среде облачных вычислений в рамках модели SaaS;
- 2) предоставление средств разработки новых сервисов и композитных приложений на их основе в рамках моделей SaaS и AaaS;
- 3) подготовка файлов входных данных и расчетных параметров, необходимых для выполнения композитных приложений;
- 4) автоматизированное определение целевых ресурсов, доступных для исполнения сервисов в составе композитного приложения, оптимизация его выполнения в распределенной среде;
- 5) обеспечение мониторинга процесса исполнения пользовательского задания и динамического управления расчетами;
- 6) обеспечение работы с пользовательскими данными в удаленном хранилище;
- 7) обеспечение аутентификации пользователя в среде Грид по персональному сертификату;
- 8) обеспечение информационной и интеллектуальной поддержки пользователя МИТП на всех этапах разработки и использования композитных приложений.

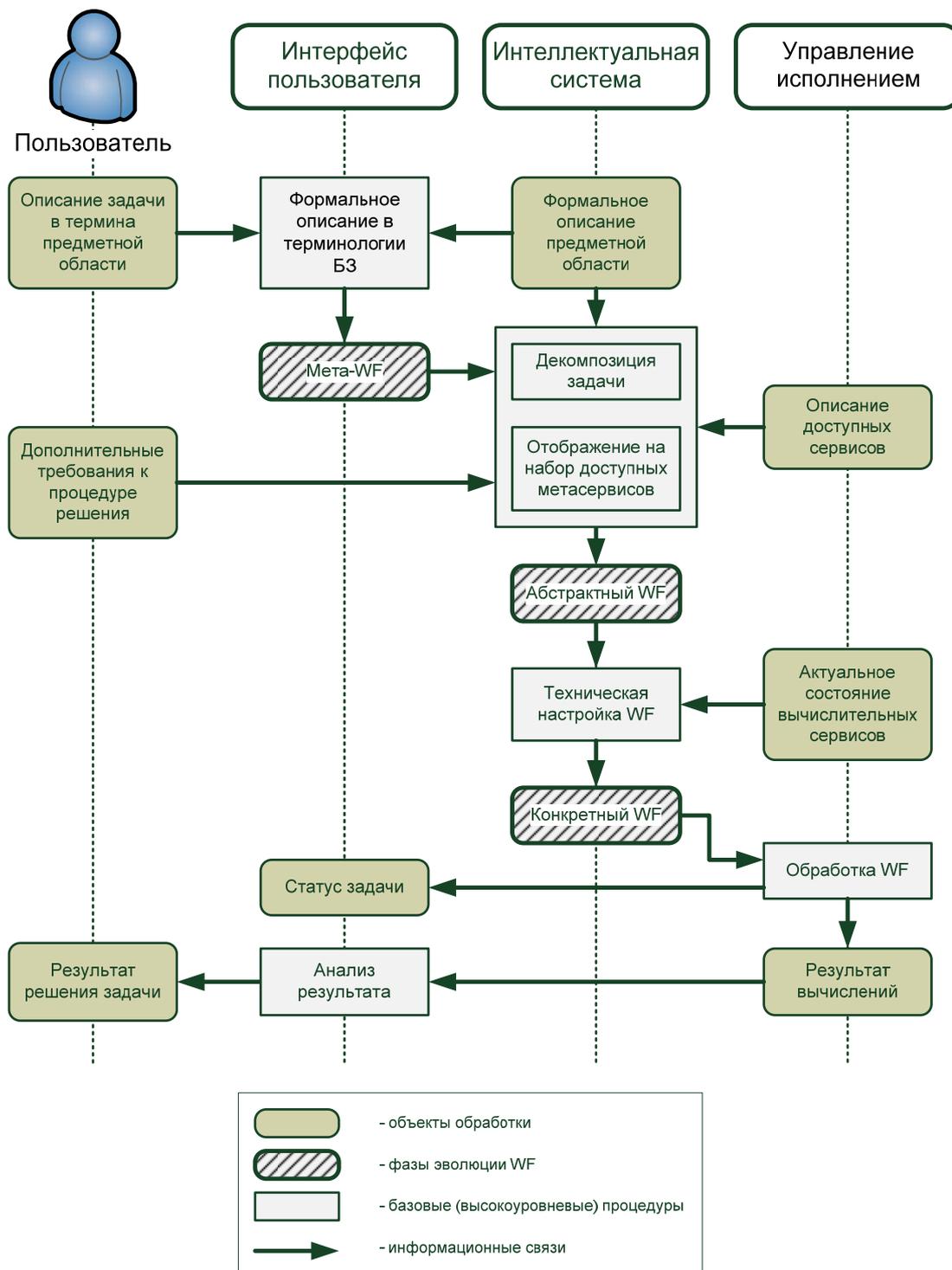


Рисунок 3.1 – Общая схема создания и исполнения композитного приложения

в МИТП на основе iPSE

3.2. Методы решения задачи

В данном разделе рассматриваются основные методы решения задачи раздела 3.1, реализуемые в рамках базового функционала МИТП, общего для всех технологических платформ. Они предназначены для:

RU.СНАБ.80066-06 31 01 **Ошибка! Источник ссылки не найден.**

- 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;
- 3) разборки и интерпретации описаний композитных приложений с целью формирования единого внутреннего представления МИТП, пригодного для запуска на различных классах вычислительных ресурсов распределенной среды;
- 4) выбора и планирования оптимального использования вычислительных ресурсов для исполнения композитных приложений
- 5) организации всего процесса исполнения композитного приложения в соответствии с рис. 3.1.

3.2.1. Унифицированное описание прикладных пакетов в МИТП

В МИТП простейшая форма WF представляет собой описание исполнения одного вычислительного пакета в распределенной среде 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 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) и 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* –

RU.СНАБ.80066-06 31 01Ошибка! Источник ссылки не найден.

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 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).

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.

RU.СНАБ.80066-06 31 01 **Ошибка! Источник ссылки не найден.**

```

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

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.

RU.СНАБ.80066-06 31 01 **Ошибка! Источник ссылки не найден.**

3.2.2. Унифицированное описание композитных приложений в МИТП

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, 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 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).

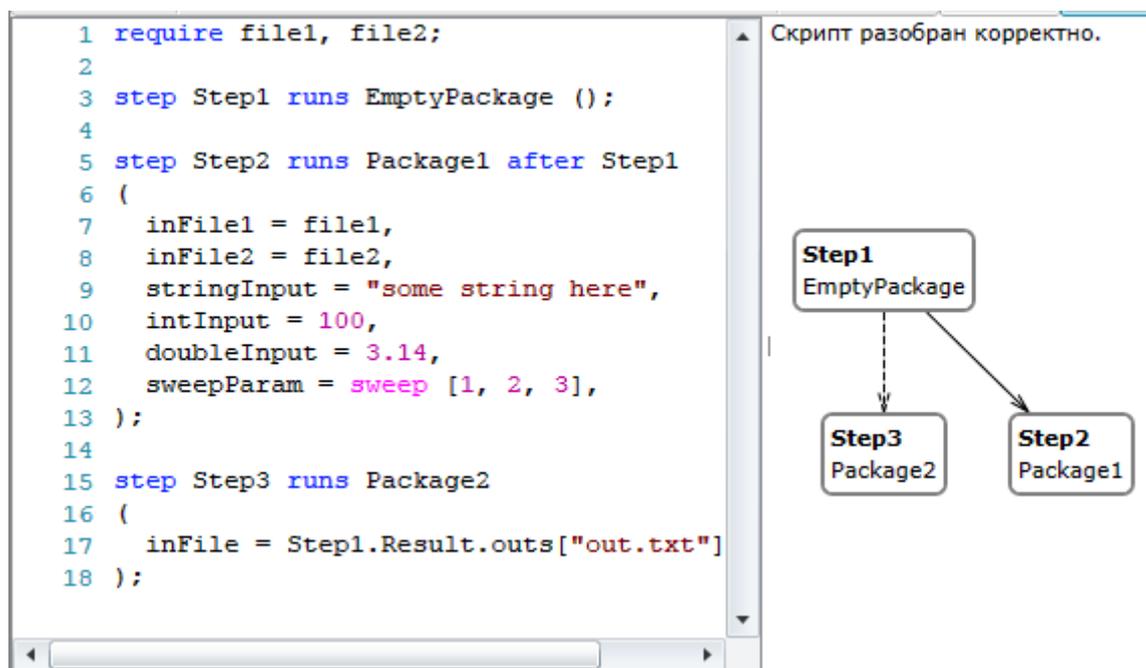


Рисунок 3.3 – Пример описания композитного приложения на языке EasyFlow и его графическое представление в МИТП

RU.СНАБ.80066-06 31 01 **Ошибка! Источник ссылки не найден.**

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.

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, это 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.

RU.СНАБ.80066-06 31 01 **Ошибка! Источник ссылки не найден.**

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.

3.2.3. Разбор и интерпретация скрипта на языке EasyFlow

Скрипт композитного приложения на языке EasyFlow (см. раздел 3.2.2) описывает AWF, т.е. композитное приложение, задачи в котором не ассоциированы с вычислительными ресурсами распределенной среды, на которых они будут выполняться. Потому для исполнения композитного приложения необходимо провести интерпретацию скрипта во внутреннее представление МИТП таким образом, чтобы преобразовать AWF в CWF, т.е. задать сценарий выполнения расчетов с конкретными прикладными пакетами.

Разбор скрипта EasyFlow подразумевает последовательное выполнение лексического, синтаксического и семантического анализа; в МИТП данные операции поддерживаются соответствующими блоками-анализаторами. Лексический анализатор предназначен для разбиения исходного текста EasyFlow на набор лексем. Синтаксический анализатор с помощью этого набора анализирует язык на основе формального описания его грамматики, а семантический анализатор – производит действия, связанные с определением сути описанных в скрипте операций. На рис. 3.4 представлена схема взаимодействия компонентов МИТП (см. документ RU.СНАБ.80066-06 01 01) при разборе и интерпретации скрипта EasyFlow. Из рисунка видно, что обработка скрипта представляет собой процесс последовательных трансформаций и попутной обработки с построением внутреннего представления.

Скрипт поступает в компонент интерпретации WF CLAVIRE/FlowSystem RU.СНАБ.80066-06 01 20 от компонента взаимодействия с пользователем CLAVIRE/Ginger RU.СНАБ.80066-06 01 21 в текстовом виде для первичного лексического анализа. На этом этапе на основе формального определения грамматики языка EasyFlow происходит разбиение исходного текста на набор лексем, соответствующих основным

RU.СНАБ.80066-06 31 01 **Ошибка! Источник ссылки не найден.**

понятиям языка (идентификатор, оператор, строка, число, комментарий и пр.) посредством лексического анализатора.

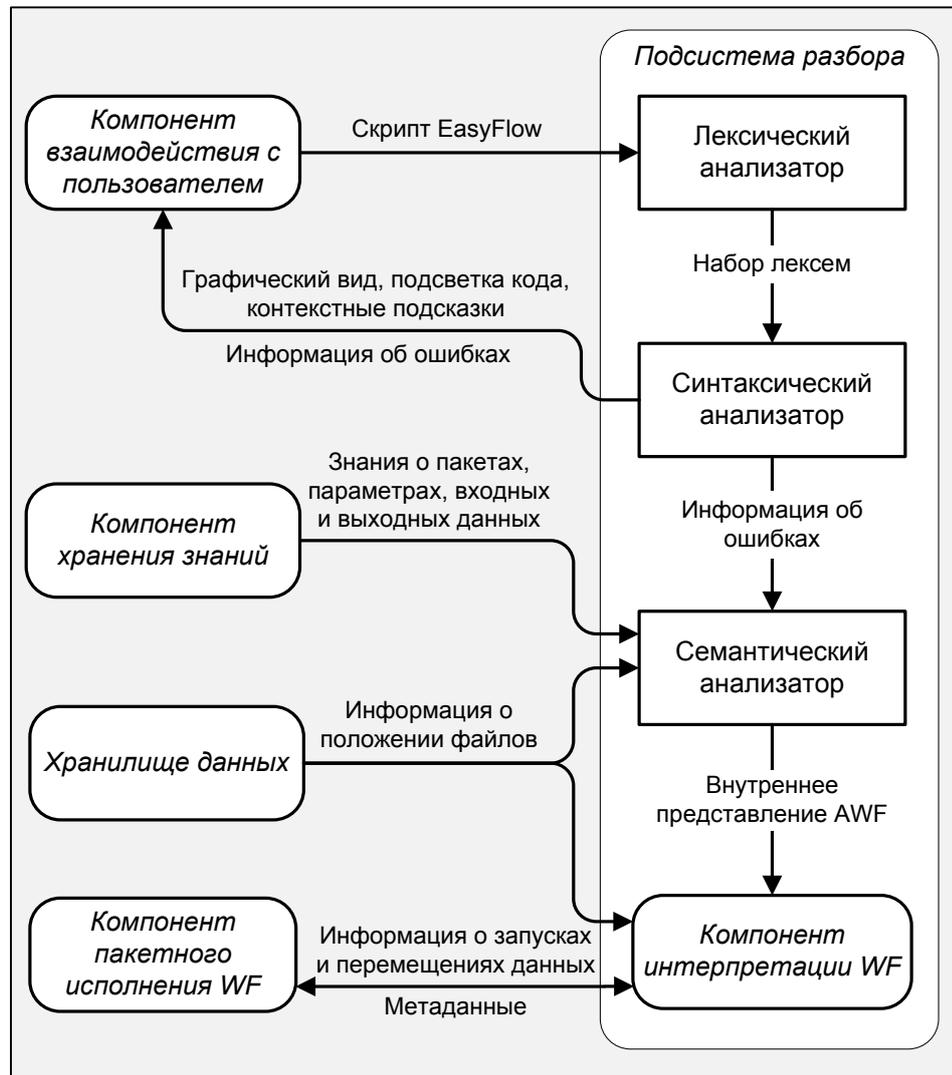


Рисунок 3.4 – Функциональная схема разбора и интерпретации скрипта EasyFlow в МИТП

При этом удаляются комментарии, пробелы, символы табуляции и возврата каретки. На выходе получается поток лексем с информацией об их типе и положении в тексте. На этом этапе происходит также первичная обработка ошибок, связанная с нахождением в тексте неприемлемых символов и лексем, не соответствующих формальной грамматике языка (см. приложение).

Полученный набор лексем является исходными данными для работы следующего блока – синтаксического анализатора, который на основе формальной грамматики языка выделяет из текста отдельные конструкции и строит внутреннее представление скрипта в объектном виде. На этом этапе пользовательский интерфейс может получать информацию о графическом представлении WF, данные о возникающих ошибках и служебную

RU.СНАБ.80066-06 31 01 **Ошибка! Источник ссылки не найден.**

информацию для подсветки синтаксиса в текстовом редакторе и осуществления контекстных подсказок (при совместной работе с компонентом-базой пакетов CLAVIRE/PackageBase RU.СНАБ.80066-06 01 35).

После синтаксического анализа происходит вызов блока семантического анализа, который на основе знаний, получаемых от CLAVIRE/PackageBase RU.СНАБ.80066-06 01 35, определяет состав вызываемых пакетов, их формат, набор параметров, а также проверяет возможность исполнения переданного WF и его непротиворечивость (например, отсутствие циклических зависимостей). На выходе этого этапа, в случае отсутствия критических ошибок, получается внутреннее представление WF, пригодное для передачи в компонент запуска.

Пример разбора текста скрипта во внутреннее представление WF показан на рис. 3.5 (в левой части) стрелками показаны зависимости данных от запусков).

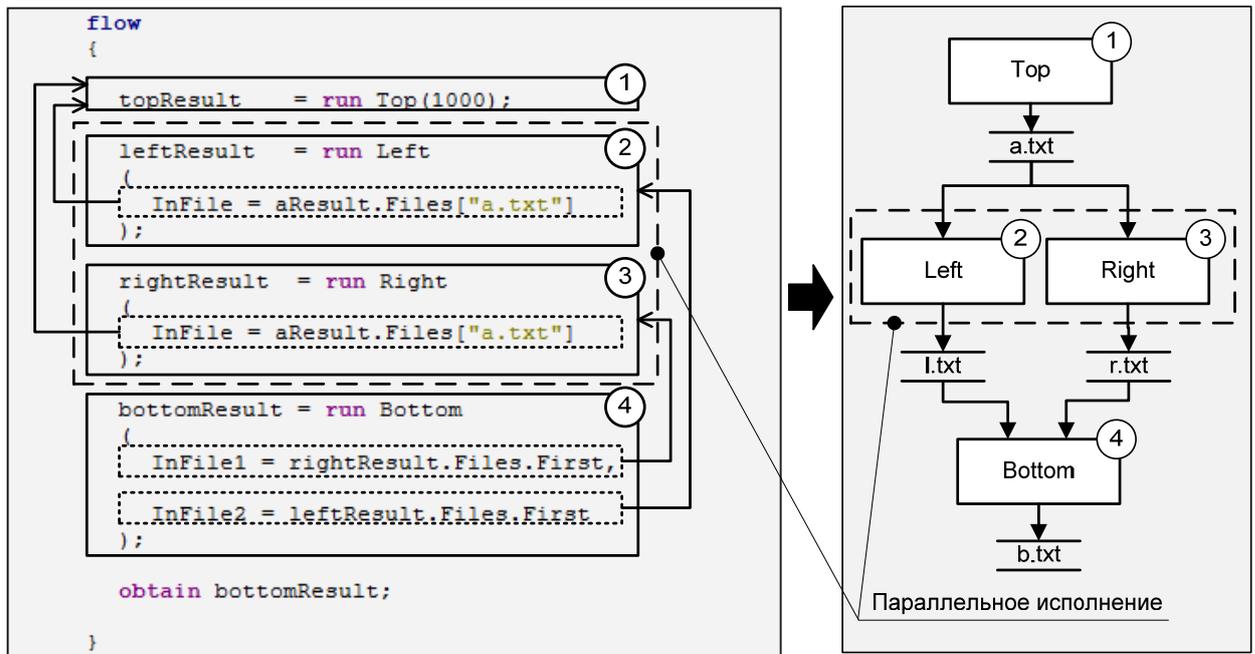


Рисунок 3.5 – Пример разбора скрипта EasyFlow в МИП

Из рис. 3.5 видно, что семантический анализатор учитывает зависимости по данным и строит соответствующее скрипту дерево исполнения. При этом учитывается возможность параллельного исполнения нескольких блоков в случае, если они не зависят друг от друга и находятся на одном уровне (т.е. им в одно и то же время становятся доступны все необходимые входные данные).

Таким образом, в результате разбора и интерпретации скрипта осуществляется формирование внутреннего представления композитного приложения, в котором изначально учтены особенности обмена данными (включая форматы входных и выходных

RU.СНАБ.80066-06 31 01 **Ошибка! Источник ссылки не найден.**

файлов) для отдельных сервисов, указанных в скрипте, вне зависимости от технологий их разработки и вычислительных ресурсов, на которых они установлены.

3.2.4. Планирование и оптимизация процесса исполнения композитного приложения в распределенной среде

Для запуска приложений в соответствии с внутренним представлением WF в МИТП (см. раздел 3.2.3) могут быть использованы различные вычислительные ресурсы. Задача планирования и оптимизации их исполнения решается в рамках концепции iPSE. Она предусматривает такой способ описания сервисов в распределенной среде, когда уже на этапе создания сервисной оболочки разработчики прикладных сервисов предоставляют информацию не только об интерфейсах их взаимодействия, но и о характеристиках производительности. Фактически эта информация также представляет собой экспертное знание, заданное в форме уравнения (параметрической модели) или табличной функции (профиля приложения). Эффективное взаимодействие сервисов в этом случае организуется самой МИТП, которая выполняет операцию логического вывода (строит субоптимальное расписание) на основе знаний о производительности прикладных сервисов и данных о функционировании распределенной системы в целом, получаемых посредством ее мониторинга в режиме реального времени. Это позволяет выбрать субоптимальную схему исполнения WF за счет управления распределением отдельных сервисов на ресурсах, способами их распараллеливания и маршрутами передачи данных.

Формальный механизм построения описания композитного приложения сводится к последовательности преобразований описания абстрактного WF в конкретный (или частично конкретный) WF. В качестве модели абстрактного WF выступает ориентированный ациклический граф

$$W_a = \{w_a = (V_a, E_a)\},$$

где множество вершин V_a – решаемые подзадачи, а множество ребер E_a – зависимости между ними по данным. Промежуточным этапом построения приложения является частично конкретный WF:

$$\begin{aligned} W_i &= \{(w_i = (V_i, E_i), state, resource)\}, \\ state &: V_i \rightarrow \{done, running, scheduled, not_scheduled\}, \\ resource &: V_i \rightarrow C \cup \{\emptyset\}, \end{aligned} \quad (3.1)$$

где $state$ – функция отображения множества решаемых подзадач на множество состояний планирования, включающего такие состояния, как «выполнено», «запущено», «спланировано», «не спланировано»; $resource$ – функция отображения множества

RU.СНАБ.80066-06 31 01 **Ошибка! Источник ссылки не найден.**

решаемых задач на множество доступных ресурсов C (в случае, если задача находится в состоянии, отличном от «не спланировано»); i – шаг частично конкретного WF.

Для составления расписания используется процедура планирования, которая может быть представлена в виде функции следующего вида:

$$sched : W_i \times T'_0 \times H \rightarrow W_i, \quad (3.2)$$

где T'_0 – множество, содержащее характеристики времени исполнения основных сервисов в составе WF, H – характеристики распределенной среды. Ход исполнения WF в целом может быть представлен в виде последовательности частично конкретных WF:

$$\begin{aligned} W_c(w_a \in W_a, sched, t'_0, h \in H) &= \{(w_i)\}, \\ w_0 &= \{w_a, state(v) = not_scheduled, resource(v) = \emptyset\}, \\ w_i &= sched(w_{i-1}, t'_0, h), i > 0, \end{aligned} \quad (3.3)$$

при этом функция оценки времени окончания счета на вычислительном ресурсе t'_0 (как основная характеристика процесса синхронизации) представляет собой отображение вида

$$t'_0 : \tilde{N} \rightarrow R^+. \quad (3.4)$$

Значения t'_0 могут быть получены различными способами, в том числе путем профилировки. Однако в рамках МИТП они интерпретируются как исходные знания предметной области, формой представления которых являются параметрические модели производительности, ассоциированные с доступными вычислительными сервисами предметной области. В компоненте CLAVIRE/PackageBase RU.СНАБ.80066-06 01 35 предусмотрены соответствующие механизмы записи параметрических моделей производительности на языке EasyPackage (см. раздел 3.2.1).

Параметрические модели производительности позволяют эффективно описывать характеристики отдельных прикладных сервисов в составе композитного приложения. Для определения времени работы WF в целом в МИТП реализована численная процедура планирования на основе различных эвристик, входными данными для которых, в соответствии с (3.2) и (3.4), являются значения времени работы отдельных сервисов. Она допускает использование различных эвристик для решения задачи управления процессом исполнения композитного приложения в распределенной среде (в частности, алгоритмы планирования MaxMin, MinMin и Sufferage) и включает в себя следующие этапы (см. рис. 3.6):

RU.СНАБ.80066-06 31 01 **Ошибка! Источник ссылки не найден.**

- 1) формализация композитного приложения: формирование структуры AWF исходя из пользовательского описания, состава данных Ξ и ограничений на режимы исполнения отдельных сервисов;
- 2) определение актуальных параметров распределенной среды (состава и текущих характеристик доступных ресурсов) с использованием инструментов мониторинга вычислительных ресурсов;
- 3) формирование набора активных фактов: оценка характеристик производительности отдельных прикладных сервисов по параметрическим моделям (как форме представления знаний, ассоциированных с элементами WF), а также определение накладных расходов, связанных с вызовом сервисов (T_C), передачей (T_N) и конвертированием (T_D) данных;
- 4) имитационное моделирование сценариев исполнения WF на основе набора конкурирующих эвристик: с использованием исходных знаний о стохастической изменчивости параметров распределенной среды методом Монте-Карло генерируются модельные ансамбли вариантов исполнения композитного приложения;
- 5) интервальное оценивание: по каждой конкурирующей эвристике строится распределение времени исполнения, после чего численно проверяется гипотеза о сходстве-различии результатов для эвристик; в результате выбирается отделимая эвристика, с минимальным средним временем исполнения и ограничением на разброс в сторону увеличения времени исполнения. В том случае, если сценарии исполнения статистически неразделимы, к реализации предлагается схема с минимальным средним временем исполнения.

Данная процедура позволяет выполнить интервальное сопоставление различных сценариев исполнения; при этом в зависимости от конкретного состояния распределенной среды может выигрывать та или иная эвристика. Это дает возможность в каждом конкретном случае рассматривать конкурирующие эвристики, вводя при этом критерии их ранжирования.

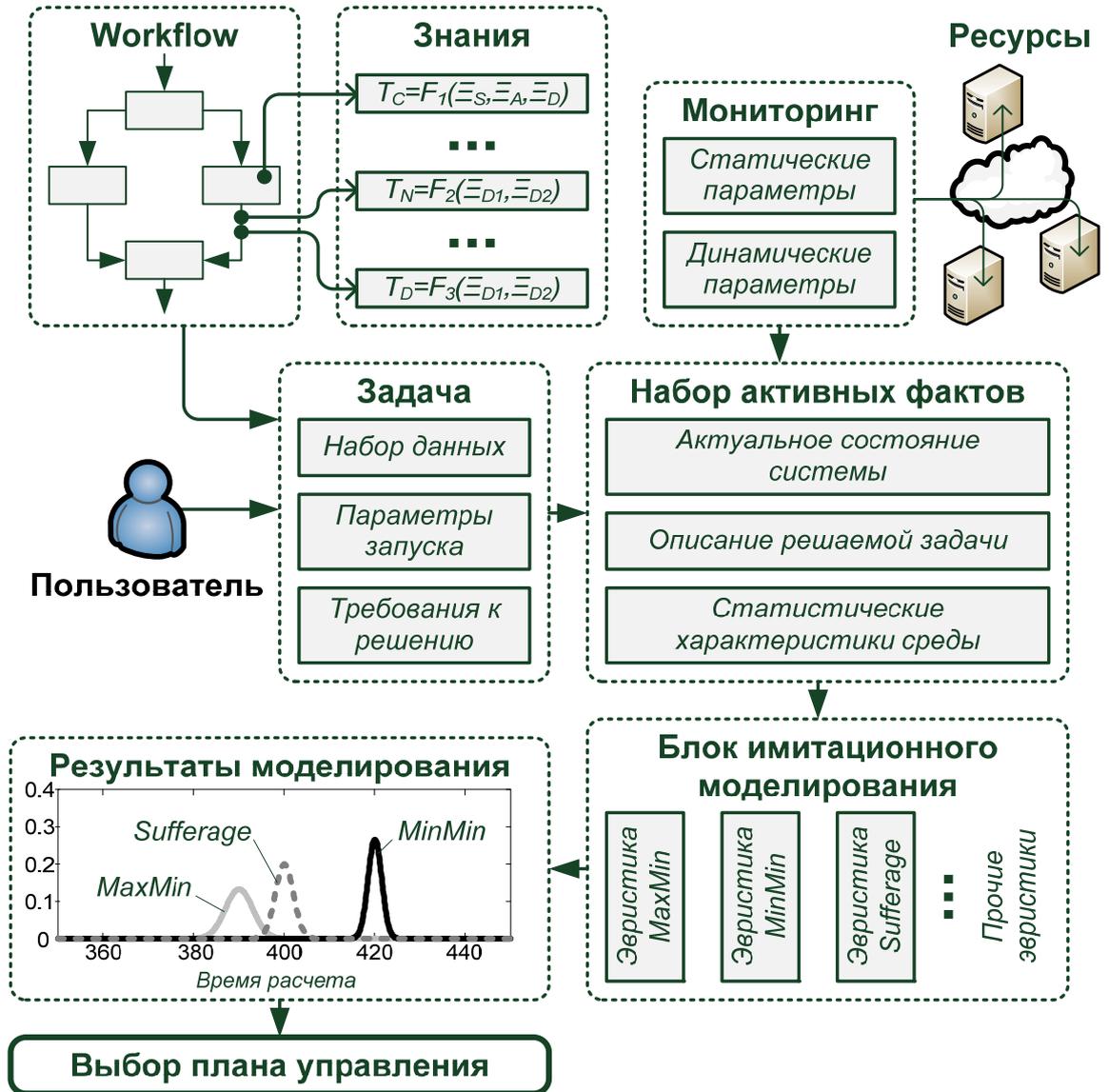


Рисунок 3.6 – Процедура планирования исполнения композитного приложения в МИТП

Таким образом, рассмотренный выше подход позволяет совокупно учесть стохастическую изменчивость характеристик распределенной среды и исходные знания о производительности прикладных сервисов в ходе планирования исполнения композитного приложения. В МИТП он реализован в компоненте планирования исполнения WF CLAVIRE/Scheduler RU.СНАБ.80066-06 01 28.

3.2.5. Организация процесса исполнения композитного приложения в МИТП

Рассмотренные выше (разделы 3.2.1–3.2.4) подходы к описанию пакетов, композитных приложений и интерфейсов работы с ними позволяют организовать процесс исполнения приложения в рамках МИТП в соответствии с рис. 3.1. Композитное приложение представляется в виде скрипта описания WF на языке

RU.СНАБ.80066-06 31 01 Ошибка! Источник ссылки не найден.

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 на конкретном ресурсе (в корпоративной среде, в среде Грид и пр.), 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. 3.7(а) приведена структура CLAVIRE/PackageBase (см. подробнее документ RU.СНАБ.80066-06 13 35). Компонент 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 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. На рис. 3.7(б) представлен сценарий исполнения композитного приложения под управлением МИТП, с использованием подходов, описанных в разделах 3.2.1–3.2.4. 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

RU.СНАБ.80066-06 31 01 **Ошибка! Источник ссылки не найден.**

01 20, там они становятся доступными пользователю (через соответствующий интерфейс человеко-компьютерного взаимодействия CLAVIRE/Ginger RU.СНАБ.80066-06 01 21).

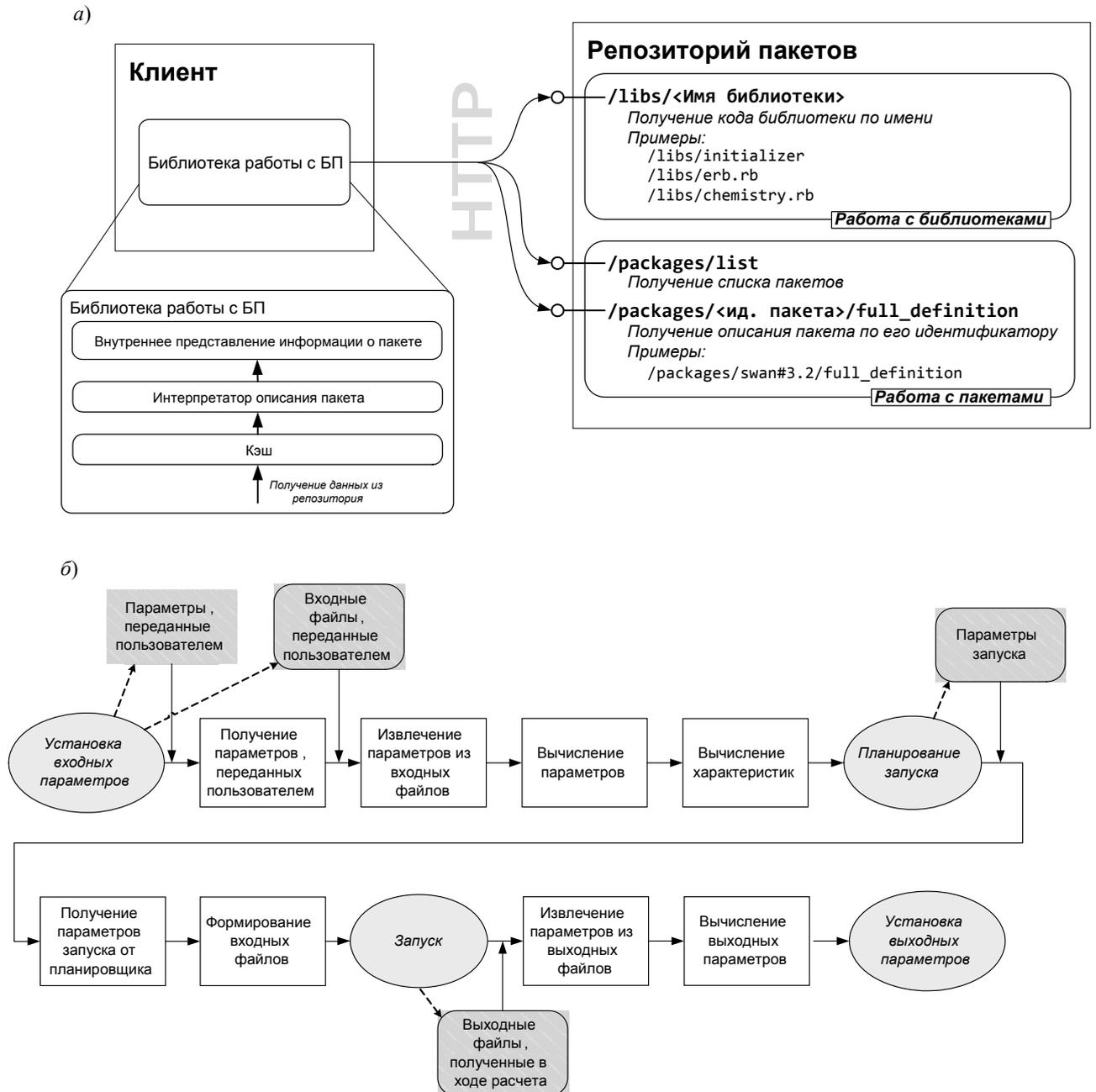


Рисунок 3.7 – Интерпретация и исполнение WF: (а) структура CLAVIRE/PackageBase ; (б) процесс исполнения композитного приложения под управлением МИТП

Таким образом, рассмотренные в разделах 3.2.1–3.2.5 методы и технологии обеспечивают решение общей задачи раздела 3.1 на уровне ядра МИТП. Для технологических платформ RU.СНАБ.80066-06 01 41...45 требуется уточнение методов,

RU.СНАБ.80066-06 31 01 **Ошибка! Источник ссылки не найден.**

связанных с решением прикладных задач и их использованием (см. документы RU.СНАБ.80066-06 31 02...06).

4. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

Для работы с МИТП не требуется специальных видов входных данных. В ходе работы входными данными 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. 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 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
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	Workflow, поток заданий

