



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 РАСПРЕДЕЛЕННЫХ
КОРПОРАТИВНЫХ ПРОГРАММНЫХ ПРИЛОЖЕНИЙ
ДЛЯ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ СЛОЖНЫХ СИСТЕМ
И ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПО УПРАВЛЕНИЮ
КОМПЛЕКСНЫМИ ОБЪЕКТАМИ (МИП-К)**

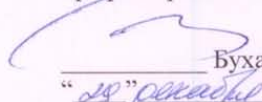
Описание применения

ЛИСТ УТВЕРЖДЕНИЯ

RU.CHAБ.80066-06 31 02-ЛУ

Представители
Организации-разработчика

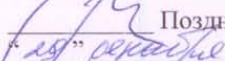
Руководитель разработки,
профессор НИУ ИТМО


Бухановский А.В.
“ 28 ” сентября 2011 г.

Ответственный исполнитель,
с.н.с. НИУ ИТМО


Луценко А.Е.
“ 29 ” сентября 2011 г.

Нормоконтролер
ведущий инженер НИУ ИТМО


Позднякова Л.Г.
“ 29 ” сентября 2011 г.

2011

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

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

УТВЕРЖДЕН
RU.СНАБ.80066-06 31 02-ЛУ

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

**ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА СОЗДАНИЯ РАСПРЕДЕЛЕННЫХ
КОРПОРАТИВНЫХ ПРОГРАММНЫХ ПРИЛОЖЕНИЙ
ДЛЯ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ СЛОЖНЫХ СИСТЕМ
И ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПО УПРАВЛЕНИЮ
КОМПЛЕКСНЫМИ ОБЪЕКТАМИ (МИП-К)**

Описание применения

RU.СНАБ.80066-06 31 02

ЛИСТОВ 34

2011

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

АННОТАЦИЯ

Документ содержит описание применения технологической платформы создания распределенных корпоративных программных приложений для компьютерного моделирования сложных систем и поддержки принятия решений по управлению комплексными объектами (МИТП-К) RU.СНАБ.80066-06 01 41. Технологическая платформа МИТП-К входит в состав многопрофильной инструментально-технологической среды CLAVIRE (Cloud Applications Virtual Environment) RU.СНАБ.80066-06. Она предназначена для создания предметно-ориентированных высокопроизводительных композитных приложений корпоративного уровня в неоднородной распределенной вычислительной среде с обеспечением эффективного использования вычислительных ресурсов организации потребителя. МИТП-К 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-К	9
2.3. Необходимые технические средства подсистемы вычислительной инфраструктуры МИТП-К	10
3. ОПИСАНИЕ ЗАДАЧИ	11
3.1 Определение задачи	11
3.2 Методы решения задачи	13
3.2.1. Унифицированное описание прикладных пакетов в МИТП-К	15
3.2.2. Унифицированное описание композитных приложений в МИТП-К	18
3.2.3. Планирование и оптимизация процесса исполнения композитного приложения в распределенной среде	20
3.2.4. Организация процесса исполнения композитного приложения в МИТП-К	23
3.2.5. Особенности создания и управления распределенных корпоративных программных приложений для компьютерного моделирования сложных систем и поддержки принятия решений по управлению комплексными объектами	26
3.2.6. Решение типовой задачи компьютерного моделирования сложных систем с использованием МИТП-К	28
4. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ	32
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ	33
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ	34

RU.СНАБ.80066-06 31 02 **Ошибка! Источник ссылки не найден.**

1. НАЗНАЧЕНИЕ ПРОГРАММЫ

Технологическая платформа создания распределенных корпоративных программных приложений для компьютерного моделирования сложных систем и поддержки принятия решений по управлению комплексными объектами (МИТП-К) RU.СНАБ.80066-06 01 41 входит в состав многопрофильной инструментально-технологической среды (МИТП) CLAVIRE (Cloud Applications Virtual Environment) RU.СНАБ.80066-06. Она предназначена для создания предметно-ориентированных высокопроизводительных композитных приложений корпоративного уровня в неоднородной распределенной вычислительной среде с обеспечением эффективного использования вычислительных ресурсов организации потребителя.

1.1. Функциональное назначение

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

- 1) эффективного управления вычислительными, информационными и программными ресурсами неоднородной корпоративной среды в рамках модели облачных вычислений (частного облака);
- 2) создания, исполнения, управления и предоставления сервисов доступа к предметно-ориентированным высокопроизводительным композитным приложениям, функционирующим на основе облака распределенных прикладных сервисов в корпоративной среде¹;
- 3) обеспечения функционирования программно-аппаратного комплекса (ПАК) поддержки инфраструктуры предметно-ориентированных облачных вычислений в различных предметных областях на основе корпоративных ресурсов организации потребителя.

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-К предназначена для создания распределенных корпоративных программных приложений с целью компьютерного моделирования сложных систем и поддержки принятия решений по управлению комплексными объектами. Она применима для решения следующих классов задач.

1. Численное моделирование сложных неоднородных систем и объектов с применением набора прикладных пакетов, поддерживаемых различными подразделениями организации-потребителя.
2. Обеспечение удаленного доступа к типовым сервисам и готовым композитным приложениям в области компьютерного моделирования и обработки данных, связанным с основными технологическими процессами организации-потребителя.
3. Выполнение массовых расчетов с автоматизированным распараллеливанием по данным на свободных вычислительных ресурсах организации-потребителя.
4. Эффективное использование совокупных ресурсов корпоративной вычислительной среды и их динамическое распределение между подразделениями организации потребителя.

1.3. Основные характеристики

Технологическая платформа МИТП-К 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 на основе корпоративных вычислительных ресурсов в рамках локальной сети организации-потребителя.

МИТП-К представляет собой программную платформу корпоративного уровня, которая разворачивается на вычислительных ресурсах заказчика (предприятия, имеющего потребность в осуществлении соответствующих расчетов) и далее использует только вычислительные ресурсы и сервисы в его локальной вычислительной сети (модель «частного облака»). Она обеспечивает выполнение следующих функций.

- 1) Поддержку разработки и исполнения композитных приложений на основе ресурсов корпоративной распределенной среды – набора программно-аппаратных комплексов, принадлежащих одному владельцу, связанных локальной вычислительной сетью и предназначенных для решения задач, координируемых единым набором требований.

RU.СНАБ.80066-06 31 02Ошибка! Источник ссылки не найден.

- 1) Динамическое управление (мониторинг состояния, запуск приложений, передача данных, распределение нагрузки) в автоматическом режиме набором распределенных ресурсов, доступных в корпоративной распределенной среде.
- 2) Автоматическую оптимизацию по времени процесса использования доступных вычислительных ресурсов и прикладных сервисов в корпоративной распределенной среде..
- 3) Представление описания композитных приложений в корпоративной распределенной среде на основе цепочек заданий (workflow), обеспечивающих запуск, выполнение, остановку и возобновление работы цепочки заданий в ручном и автоматическом режимах.
- 4) Унифицированный доступ к вычислительным ресурсам корпоративной распределенной среды на основе интерфейсов доступа в составе программно-аппаратной архитектуры SMP, MPP, GPGPU, СВЕА.
- 5) Поддержку процесса установки и первоначальной конфигурации технологической платформы и ее составных частей на ресурсах корпоративной распределенной среды.
- 6) Поддержку многопользовательского режима в корпоративной распределенной среде.
- 7) Квотирование использования ресурсов корпоративной распределенной среды.
- 8) Каталогизацию входных данных пользователей на основе метаданных.
- 9) Администрирование и контроль работы с дифференцированными правами администраторов в рамках единой политики доступа к ресурсам корпоративной распределенной среды.
- 10) Модификация знаний, используемых системой в ручном и автоматическом режимах.
- 11) Функционирование сервисов резервирования и отката изменений пользовательских данных в удаленном хранилище в составе корпоративной распределенной среды.
- 12) Функционирование механизмов конвертирования данных между различными прикладными сервисами, по заданию пользователя.
- 13) Функционирование сервисов, обеспечивающих интерфейс с многофункциональными и предметно-ориентированными системами визуализации результатов расчетов с использованием прикладных сервисов и композитных приложений на их основе.

RU.СНАБ.80066-06 31 02 **Ошибка! Источник ссылки не найден.**

1.4. Ограничения, накладываемые на область применения

Специфика использования корпоративной вычислительной среды накладывает на применение МИТП-К следующие ограничения:

- 1) в состав МИТП-К не входят прикладные пакеты; они должны устанавливаться и настраиваться на корпоративных ресурсах отдельно, в зависимости от предпочтений организации-потребителя;
- 2) идентификация и аутентификация корпоративных пользователей происходит в упрощенном режиме (без применения дополнительных средств, таких как сертификаты пользователей, делегирование прав для аутентификации на вычислительных ресурсах и пр.);
- 3) пользователь МИТП-К имеет ограниченные возможности разработки собственных композитных приложений (предполагается использование типовых шаблонов сервисов и приложений);
- 4) ограничены права пользователя по целевому выбору ресурса (предоставление ресурсов регулируется политикой организации потребителя).

RU.СНАБ.80066-06 31 02 **Ошибка! Источник ссылки не найден.**

2. УСЛОВИЯ ПРИМЕНЕНИЯ

2.1 Условия развертывания программы

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

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

- 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 доступа к вычислительным ресурсам RU.СНАБ.80066-06 01 38.

Для развертывания компонентов МИТП-К 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

RU.СНАБ.80066-06 31 02 **Ошибка! Источник ссылки не найден.**

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-К 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

2.2. Необходимые технические средства управляющей подсистемы МИТП-К

Компоненты МИТП-К 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.

RU.СНАБ.80066-06 31 02 **Ошибка! Источник ссылки не найден.**

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

2.3. Необходимые технические средства подсистемы вычислительной инфраструктуры МИТП-К

В состав комплекса технических средств подсистемы вычислительной инфраструктуры МИТП-К входят высокопроизводительные вычислительные системы, а также специализированные системы. МИТП-К обеспечивает унифицированный доступ и управление вычислительным процессом на базе различных классов технических средств, различающихся архитектурой, характеристиками, режимом доступа и т.п.

1. *Вычислительные кластеры.* Доступные по локальной сети высокопроизводительные ресурсы, предназначенные для установки (в т.ч. автоматической) и последующего использования прикладных сервисов МИТП-К. Ввиду того что МИТП-К 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 облачная инфраструктура.* Виртуальная вычислительная инфраструктура, конфигурируемая по запросу со стороны МИТП-К или пользователя системы (при организации доступа к уже сконфигурированным

RU.СНАБ.80066-06 31 02 **Ошибка! Источник ссылки не найден.**

виртуальным ресурсам). Со стороны МИТП-К 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. *Прочие виды корпоративных ресурсов* (рабочие станции, серверные ЭВМ и пр.).

Интеграция широкого спектра разнородных ресурсов позволяет сформировать инфраструктуру, обеспечивающую, с одной стороны, исполнение заданий, оптимизированных для различных архитектур, и с другой – использование свободных вычислительных ресурсов организации потребителя. Минимальные требования к корпоративным ресурсам для использования в МИТП-К:

- 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;
- тактовая частота каждого процессора – не менее 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.

4. *Специализированные хранилища данных*. МИТП-К обеспечивает унифицированный доступ как к локальным, так и распределенным хранилищам и источникам данных при условии их нахождения в локальной сети предприятия.

3. ОПИСАНИЕ ЗАДАЧИ

3.1 Определение задачи

Основной задачей МИТП-К является обеспечение работы пользователя с облачными сервисами и композитными приложениями в корпоративной распределенной вычислительной среде. Организация процесса создания и исполнения композитного приложения под управлением МИТП-К в рамках концепции iPSE сводится к последовательной формализации наборов описаний в терминах потоков заданий (workflow, WF). На первом этапе процесса проектирования композитного приложения

RU.СНАБ.80066-06 31 02 **Ошибка! Источник ссылки не найден.**

создается 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 его основе 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 – конкретное местоположение 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.

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

1. Пользователь авторизуется в МИТП-К через Интернет/Интранет-портал организации потребителя, что дает ему возможность доступа к соответствующим ресурсам и сервисам в рамках корпоративной вычислительной инфраструктуры.
2. Пользователь МИТП-К через соответствующий web-интерфейс МИТП может выбрать конкретные сервисы или уже заранее созданные WF композитных приложений на их основе, а также получить (при необходимости) доступ к эксплуатационной документации. Предполагается, что пользователи заранее ознакомлены с характеристиками и приемами работы с соответствующими сервисами, соответствующими профилю деятельности предприятия.
3. Выбрав сервис или приложение, пользователь конфигурирует условия вычислений: определяет требуемые параметры WF, редактирует (при необходимости) его описание, готовит и загружает в хранилище МИТП-К входные данные для расчетов.
4. Пользователь определяет режим исполнения задачи в МИТП-К (утверждает предлагаемые ему варианты) в соответствии с требованиями к временным характеристикам расчета.
5. Пользователь запускает задачу на выполнение в среде МИТП-К. Использование вычислительных ресурсов и сервисов производится с учетом соответствующих

RU.СНАБ.80066-06 31 02 **Ошибка! Источник ссылки не найден.**

квот, выделенных пользователю службой администрирования корпоративной среды. Распределение задач по ресурсам выдается пользователю в виде справочной информации.

6. Пользователь (при необходимости) осуществляет мониторинг процесса исполнения (в форме динамического отображения WF); при этом прогнозируется время завершения вычислений. Пользователь может на время расчетов завершить рабочую сессию с МИТП-К и начать ее только при необходимости использования результатов (необязательно в тот момент, когда расчеты завершены).
7. Когда расчет задачи завершен, результаты помещаются в соответствующее хранилище данных МИТП-К (на основе корпоративного хранилища данных предприятия); пользователю отправляется соответствующее уведомление. Пользователь может получить доступ к результатам расчетов (в форме выдачи результатов соответствующих прикладных пакетов в составе сервисов) через интерфейс МИТП-К.

Таким образом, основная задача МИТП-К 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-К, 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;
- 3) организации всего процесса исполнения композитного приложения под управлением МИТП-К.

RU.СНАБ.80066-06 31 02 Ошибка! Источник ссылки не найден.

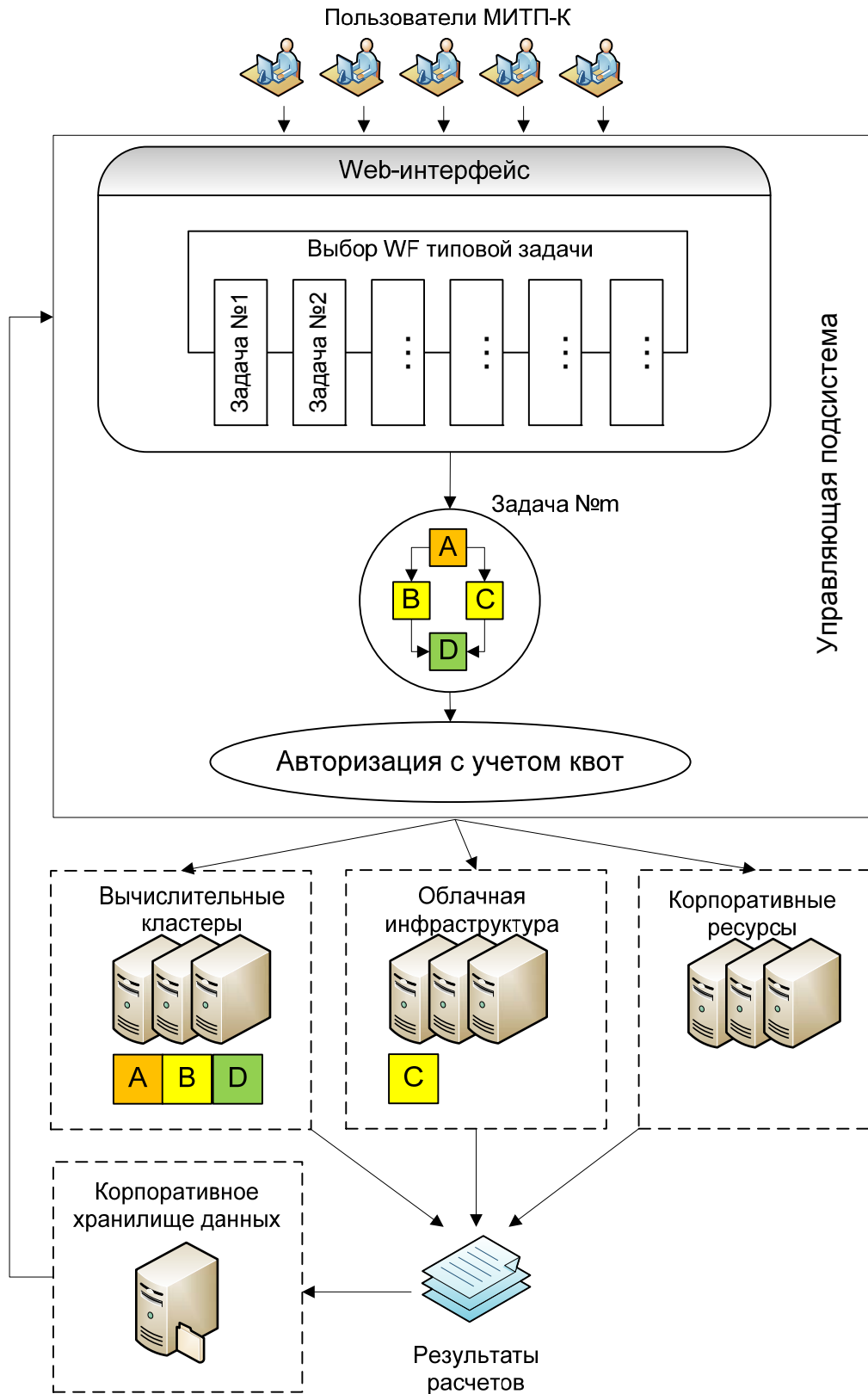


Рисунок 3.1 Принцип функционирования МИТП-К

RU.СНАБ.80066-06 31 02 **Ошибка! Источник ссылки не найден.**

3.2.1. Унифицированное описание прикладных пакетов в МИТП-К

В МИТП-К 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-К 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* – 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,

RU.СНАБ.80066-06 31 02 **Ошибка! Источник ссылки не найден.**

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 02 **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 02 **Ошибка! Источник ссылки не найден.**

3.2.2. Унифицированное описание композитных приложений в МИТП-К

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-К, позволяет упростить процедуру задания композитных приложений. Он предоставляет конечному пользователю гибкие возможности по заданию различных форм WF, в рамках которых выполняются различные прикладные пакеты, происходят 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, описание которых рассмотрено в разделе 3.2.1.

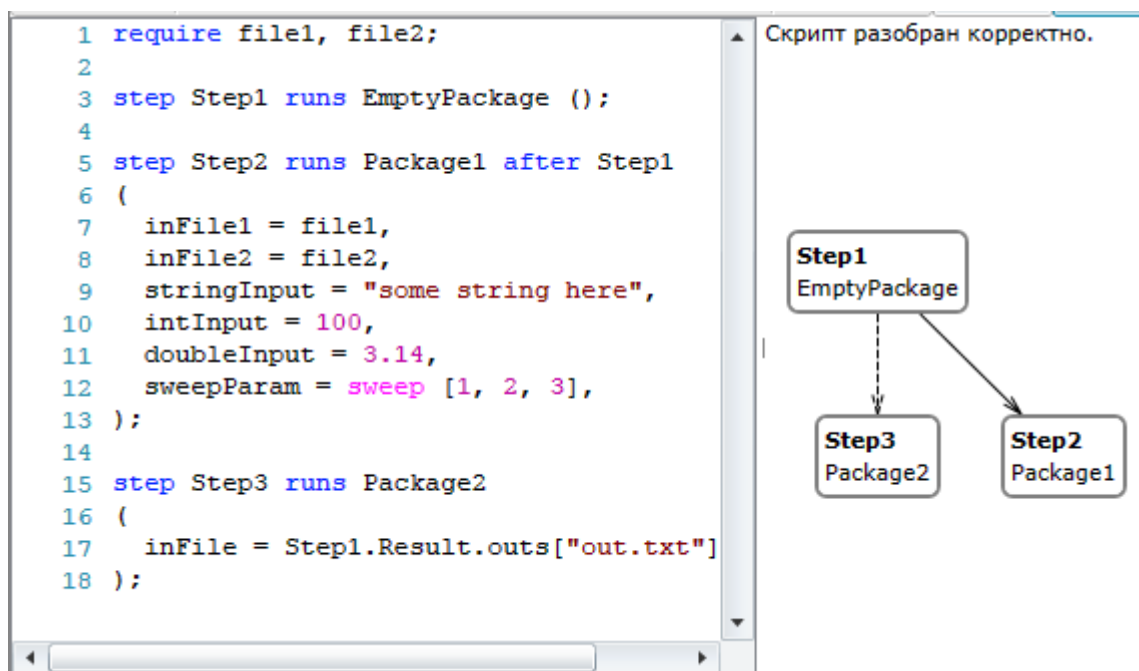


Рисунок 3.3 – Пример описания композитного приложения на языке EasyFlow и его графическое представление в МИТП

RU.СНАБ.80066-06 31 02 **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, 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.

RU.СНАБ.80066-06 31 02 **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. Планирование и оптимизация процесса исполнения композитного приложения в распределенной среде

Для запуска приложений в МИТП-К 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 их 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-К, 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

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

RU.СНАБ.80066-06 31 02 **Ошибка! Источник ссылки не найден.**

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 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

RU.СНАБ.80066-06 31 02 **Ошибка! Источник ссылки не найден.**

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. 3.4):

- 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.

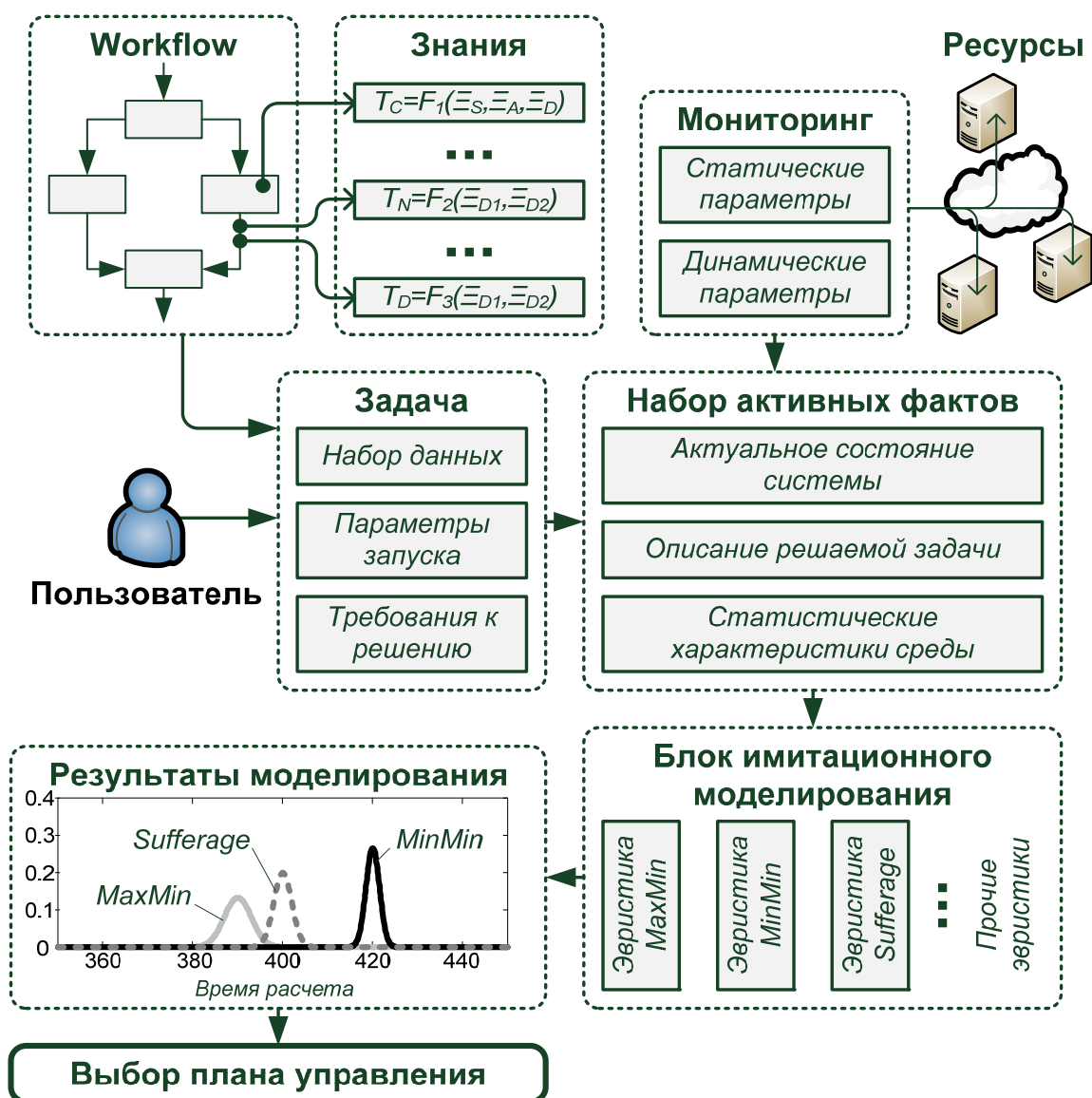


Рисунок 3.4 – Процедура планирования исполнения композитного приложения в МИП-К

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 МИП-К он реализован в компоненте планирования исполнения WF CLAVIRE/Scheduler RU.СНАБ.80066-06 01 28.

3.2.4. Организация процесса исполнения композитного приложения в МИП-К

Рассмотренные выше подходы (см. разделы 3.2.1–3.2.3) к описанию пакетов, композитных приложений и интерфейсов работы с ними позволяют организовать процесс исполнения приложения в рамках МИП-К. Композитное приложение представляется в виде скрипта описания WF на языке EasyFlow, который может быть

RU.СНАБ.80066-06 31 02 **Ошибка! Источник ссылки не найден.**

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. 3.5(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.

RU.СНАБ.80066-06 31 02 Ошибка! Источник ссылки не найден.

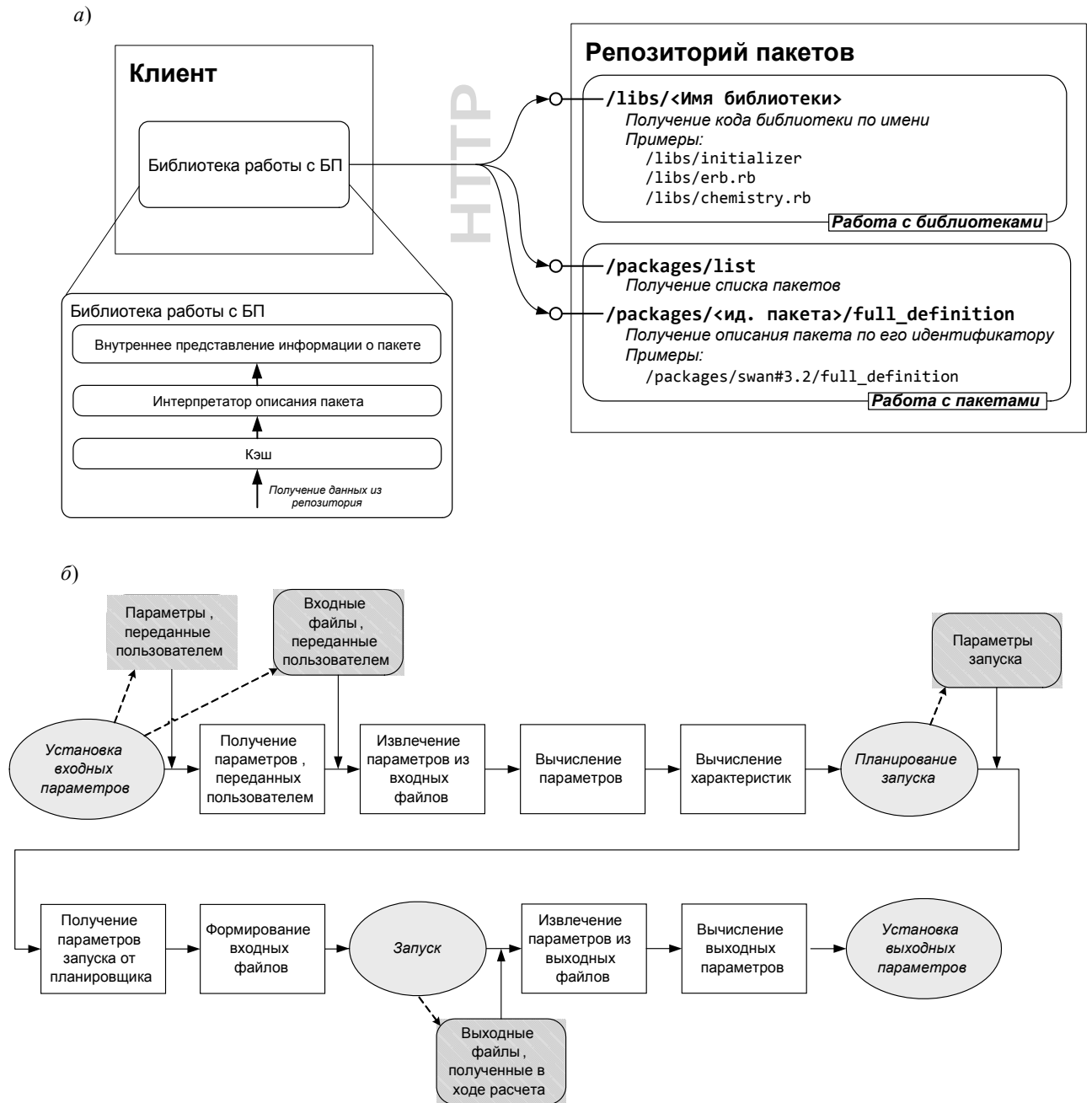


Рисунок 3.5 – Интерпретация и исполнение WF: (а) структура CLAVIRE/PackageBase ; (б) процесс исполнения композитного приложения под управлением МИТП-К

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. 3.5(б)

RU.СНАБ.80066-06 31 02 **Ошибка! Источник ссылки не найден.**

представлен сценарий исполнения композитного приложения под управлением МИТП, с использованием подходов, описанных в разделах 3.2.1–3.2.3. 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 пользователю (через соответствующий интерфейс человеко-компьютерного взаимодействия CLAVIRE/Ginger RU.СНАБ.80066-06 01 21).

Таким образом, рассмотренные в разделах 3.2.1–3.2.4 методы и технологии обеспечивают решение основной задачи МИТП-К – создания предметно-ориентированных высокопроизводительных композитных приложений корпоративного уровня в неоднородной распределенной вычислительной среде с обеспечением эффективного использования вычислительных ресурсов организации потребителя.

3.2.5. Особенности создания и управления распределенных корпоративных программных приложений для компьютерного моделирования сложных систем и поддержки принятия решений по управлению комплексными объектами

В качестве особенностей создания и управления распределенными корпоративными программными приложениями, стоит выделить следующие моменты.

- 1) Используя инструментальную среду, на языке EasyFlow разработчиком (оператором) МИТП-К реализуются короткие типовые WF для массового запуска прикладных пакетов под корпоративные задачи организации. Сами пользователи, в основном, выступают как потребители сервисов, а не как разработчики.
- 2) Основная часть WF, в виде шаблонов, реализуется программистами МИТП-К, что позволяет конечным пользователям при создании очередного расчета, избегать написания композитных приложений с нуля и быстро загружать нужный шаблон из репозитория WF, а также вносить минимальные изменения входных параметров прикладных пакетов, тем самым концентрируя своё внимание на результатах расчета задачи.
- 3) Присущая вычислительной среде однородность данных позволяет, в целом, обходиться без реализации механизмов конвертации данных между прикладными пакетами, и быстро описывать новые типовые WF. Однако, в ряде исключений, конвертация необходима и используется для унификации формата данных.

RU.СНАБ.80066-06 31 02 **Ошибка! Источник ссылки не найден.**

- 4) При использовании WF у пользователя есть возможность управлять выполнением конкретных шагов, определяя имя ресурса для выполнения каждого шага WF или только для некоторых из них, а также варьировать их приоритеты.
- 5) При необходимости, пользователь может самостоятельно расширить используемый типовой WF, оперируя информацией об уже встроенных в платформу прикладных пакетах.

Основными пользователями МИТП-К выступают представители инженерных специальностей, осуществляющие типовые расчеты и использующие для этого достаточно устоявшиеся наборы прикладных пакетов. При этом предполагается, что пользователи имеют достаточный опыт работы с данными пакетами без использования МИТП-К, в том числе, подготовки входных данных, настройки режима исполнения, а также интерпретации результатов.

Применение МИТП-К обеспечивает следующие преимущества в процессе компьютерного моделирования и поддержки принятия решений по управлению сложными объектами:

- Пользователь имеет возможность применения МИТП-К для создания и отработки вычислительной модели при прототипировании программных комплексов моделирования и управления. В процессе прототипирования он может сосредоточиться на отработке наиболее уникальных частей комплекса; для обеспечения функционирования прототипа в целом используются базовые (подходящие) сервисы, предоставляемые МИТП-К. По результатам успешных испытаний на прототипе, данные сервисы могут быть оптимизированы (переработаны или адаптированы) при переходе к опытному образцу (который разрабатывается уже без использования МИТП-К).
- Пользователь может с помощью МИТП-К обосновать выбор способа реализации различных компонентов расчетной модели путем сравнения альтернатив реализаций сервисов, предоставляемых МИТП, в рамках решения отдельных тестовых задач (как по содержательным характеристикам, так и по временным параметрам исполнения).
- Пользователь может создавать виртуальное окружение для инженерных расчетов и исследования процессов управления сложными техническими объектами (т.е. виртуальный полигон). В данном случае МИТП-К используется как эмулятор источников входных данных.

RU.СНАБ.80066-06 31 02 **Ошибка! Источник ссылки не найден.**

- Организация – владелец МИТП-К может повысить эффективность использования (меру утилизации) собственной вычислительной инфраструктуры.

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

Характерным примером композитного приложения для МИТП-К является задача моделирования развития брочинга. Брочинг – явление потери управляемости при движении судна на попутных волнах, когда судно, подхваченное догоняющей волной, начинает двигаться со скоростью волны на ее переднем склоне и стремится развернуться к волне лагом. Когда брочинг заканчивается полным разворотом судна, его крен за счет удара догоняющей волны и инерционного момента может достичь опасных пределов.

Брочинг является следствием сочетания ряда факторов, описываемых различными вычислительными моделями, а его исследование путем вероятностного моделирования требует большого объема вычислительных ресурсов. Ниже рассматривается композитное приложение, осуществляющее расчет в два этапа посредством прямого численного моделирования динамики судна. На первом этапе моделируется движение судна в спокойной воде при различных значениях тяги винта с целью определения зависимости скорости судна от тяги винта. В дальнейшем полученная функция, представленная в табличном виде, используется для определения тяги винта, необходимой для обеспечения движения судна с заданной скоростью. На втором этапе производится многократное моделирование динамики судна на нерегулярном попутном волнении в течение заданного периода времени при единых заданных параметрах волнения (спектра) и различных начальных профилях морской поверхности с целью получения ансамбля траекторий судна. В полученном ансамбле траекторий выделяются случаи движения судна в режиме брочинга и оценивается их доля от общего числа запусков.

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

Листинг 3.1 – Код композитного приложения моделирования брочинга

```
step ForceValGeneration runs valgenerator (
  max = 800000,
  number = 4,
  precision = 2,
  seqtype = "linear"
);

step PropTableSimulation runs sxds after ForceValGeneration (
```

```

    experiment = "propulsion",
    ship = "tugboat",
    force_file = sweep ForceValGeneration.Result.Outputs.values
);

step PropTableCompilation runs scilab after PropTableSimulation (
    script_name = "SHIPXDS_propulsion.sce",
    input_folder = PropTableSimulation.Result.sweep_outs["experiment.log"],
    output_file = "prop_table"
);

step RandomSeedGeneration runs valgenerator (
    max = 1000000,
    number = 50,
    precision = 0,
    seed = 5,
    seqtype = "random"
);

step BroachingSimulation runs sxds after PropTableCompilation,
RandomSeedGeneration (
    experiment = "broaching",
    ship = "tugboat",
    velocity = 6.7,
    period = 60,
    omega = 1.6,
    narrow = 64,
    gamma = 20,
    prop_table = PropTableCompilation.Result.outs["prop_table.data"],
    rand_seed_file = sweep RandomSeedGeneration.Result.Outputs.values
);

step ProbabilityCalculation runs scilab after BroachingSimulation (
    script_name = "SHIPXDS_broaching.sce",
    input_folder = BroachingSimulation.Result.sweep_outs["experiment.log"],
    output_file = "broaching",
    output_ext = "png"
);

```

Видно, что композитным приложением последовательно выполняется шесть шагов, соответствующих определению тяги винта, необходимой для возникновения брочинга (ForceValGeneration, PropTableSimulation, PropTableCompilation) и собственно расчету и анализу поведения судна в режиме брочинга (RandomSeedGeneration, BroachingSimulation, ProbabilityCalculation). Для простоты данный WF не использует входных файлов – все параметры задаются непосредственно в скрипте.

На рис. 3.6 приведено отображение данного WF в графической форме в интерфейсе МИТП-К.

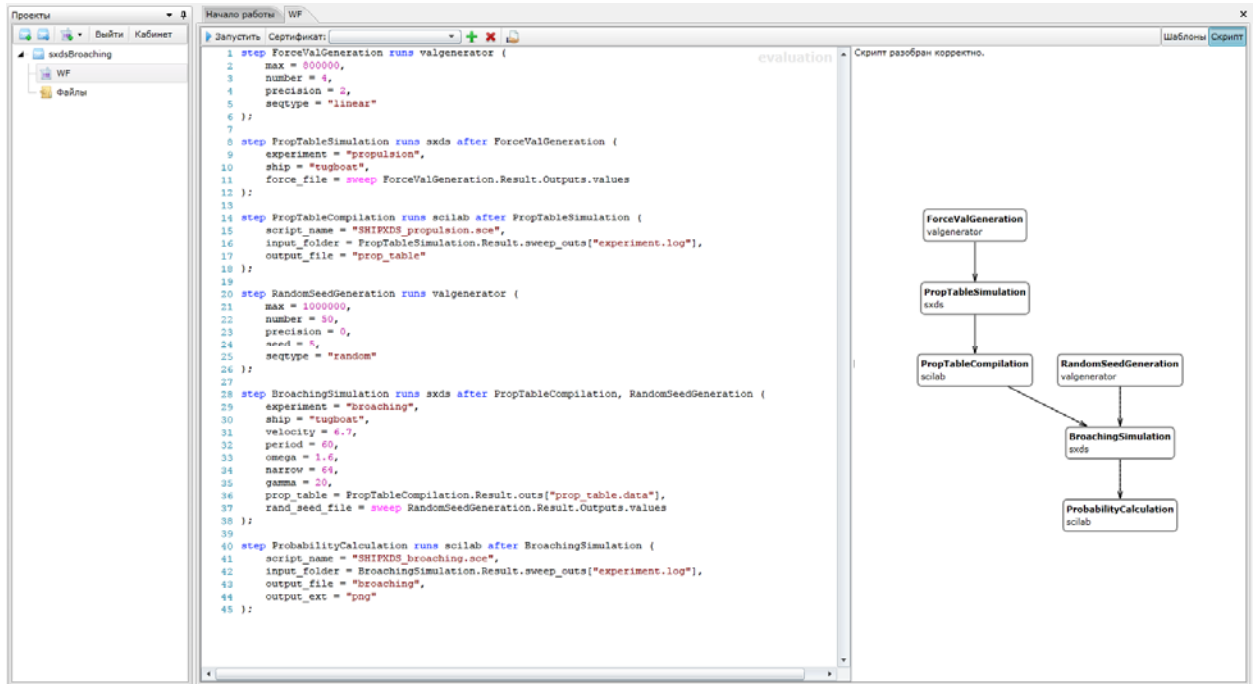


Рисунок 3.6 – Отображение WF моделирования брочинга в МИТП-К

Из рис. 3.6 видно, что приведенный в листинге 3.1 WF не является последовательной цепочкой операций: моделирование полей нерегулярного морского волнения и расчета обобщенной функции тяги винта может проводиться параллельно, на различных вычислительных ресурсах.

На рис. 3.7 приведен результат запуска композитного приложения на исполнение в корпоративной вычислительной среде из 10 вычислительных систем, выделенных данному пользователю. Видны два слоя, на которых выполнялось распараллеливание. На верхнем слое параллельно выполняются как разные пакеты (scilab, wavegenerator, sxds), так и экземпляры пакета sxds с разными данными (для определения скорости хода). На нижнем слое идет распараллеливание по статистическому ансамблю, соответствующему различным сценариям возникновения брочинга (пакет sxds).

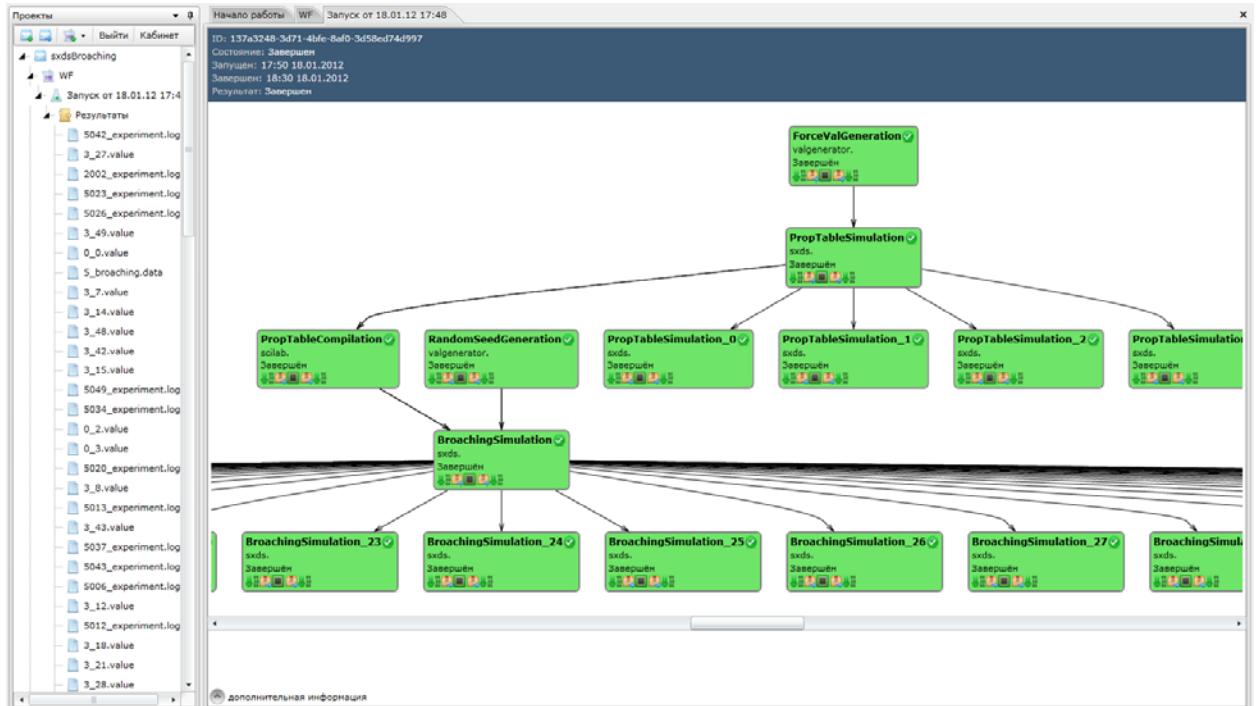


Рисунок 3.7 – Результат исполнения WF моделирования брочинга в МИТП-К

Результатом работы WF является следующий набор файлов (см. наименования в скрипте WF в листинге 3.1).

1. Набор текстовых файлов с расширением `.value`, содержащих входные параметры (целые числа) для генератора случайных чисел, на основе которого формируется начальный профиль морской поверхности.
2. Набор текстовых файлов с расширением `.value`, содержащих значения силы тяги винта судна (дробные числа), по которым строится таблица соответствия силы тяги винта и скорости судна.
3. Набор текстовых файлов `experiment.log`, содержащих результаты численных экспериментов.
4. Текстовый файл `prop_table.data`, который содержит таблицу соответствия силы тяги винта и скорости судна.
5. Текстовый файл `broaching.data`, который содержит рассчитанное значение вероятности возникновения брочинга.

Результаты расчетов по завершении исполнения WF могут быть визуализированы посредством внешних средств визуализации, подключенных к компоненту хранения данных CLAVIRE/Storage RU.СНАБ.80066-06 01 25. В качестве примера на рис. 3.8(а) приведен пример визуализации динамической сцены брочинга, а на рис. 3.8(б) – траектории и углы крена при различных сценариях возникновения брочинга.

RU.СНАБ.80066-06 31 02 **Ошибка! Источник ссылки не найден.**

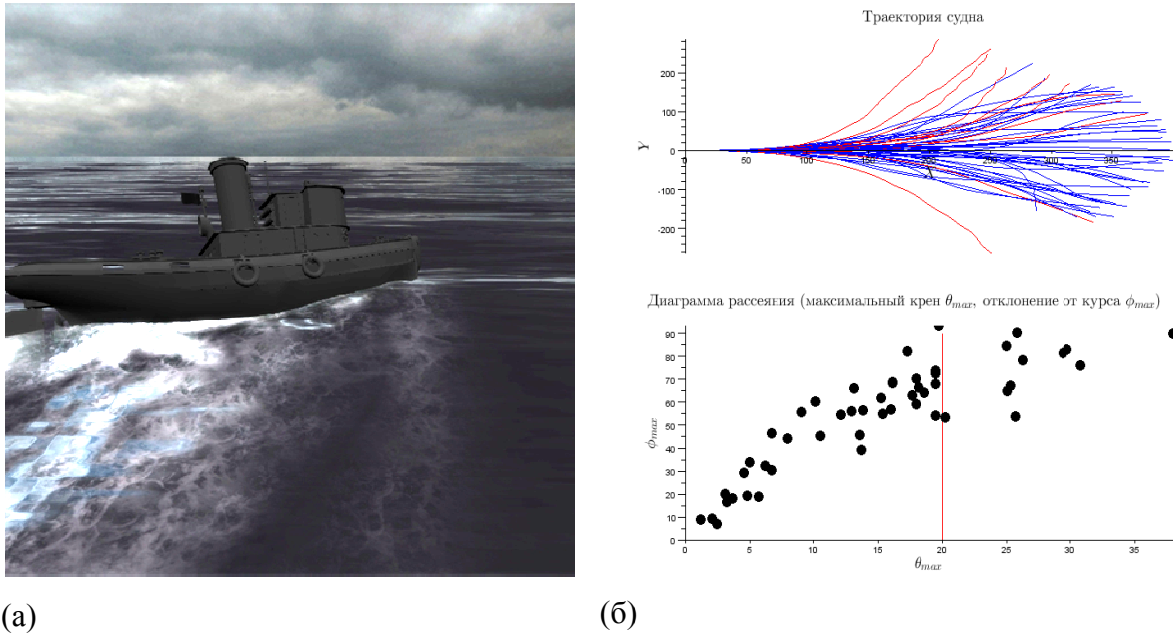


Рисунок 3.8 – Результаты визуализации броучинга: (а) динамическая сцена; (б) количественные характеристики – траектория движения судна и динамический угол крена

Подробное описание рассмотренной задачи и композитного приложения приведено в документе RU.СНАБ.80066-06 13 50.

4. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

Как видно из примера в разделе 3.2.5, для работы с МИТП-К не требуются специальные виды входных данных. В ходе работы ими 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 (см. листинг 3.1). 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
МИТП-К	Технологическая платформа создания распределенных корпоративных программных приложений для компьютерного моделирования сложных систем и поддержки принятия решений по управлению комплексными объектами (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

