

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 И ПОДДЕРЖКИ
ИНФРАСТРУКТУРЫ ЭКСТРЕННЫХ ВЫЧИСЛЕНИЙ В РАСПРЕДЕЛЕННЫХ
СРЕДАХ С ДИНАМИЧЕСКИМИ ВЫЧИСЛИТЕЛЬНЫМИ РЕСУРСАМИ
(МИП-Э)

Описание применения

ЛИСТ УТВЕРЖДЕНИЯ

RU.СНАБ.80066-06 31 05-ЛУ

Представители
Организации-разработчика

Руководитель разработки,
профессор НИУ ИТМО

Бухановский А.В.

“29” сентября 2011 г.

Ответственный исполнитель,
в.п.с. НИУ ИТМО

Луценко А.Е.

“29” сентября 2011 г.

Нормоконтролер
ведущий инженер НИУ ИТМО

Позднякова Л.Г.

“29” сентября 2011 г.

2011

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

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

УТВЕРЖДЕН
RU.СНАБ.80066-06 31 05-ЛУ

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

**ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА СОЗДАНИЯ И ПОДДЕРЖКИ
ИНФРАСТРУКТУРЫ ЭКСТРЕННЫХ ВЫЧИСЛЕНИЙ В РАСПРЕДЕЛЕННЫХ
СРЕДАХ С ДИНАМИЧЕСКИМИ ВЫЧИСЛИТЕЛЬНЫМИ РЕСУРСАМИ
(МИП-Э)**

Описание применения

RU.СНАБ.80066-06 31 05

ЛИСТОВ 36

2011

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

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

АННОТАЦИЯ

Документ содержит описание применения технологической платформы создания и поддержки инфраструктуры экстренных вычислений (urgent computing) в распределенных средах с динамическими вычислительными ресурсами (МИТП-Э) RU.СНАБ.80066-06 01 44. Технологическая платформа МИТП-Э входит в состав многопрофильной инструментально-технологической среды (МИТП) CLAVIRE (Cloud Applications Virtual Environment) RU.СНАБ.80066-06. Она предназначена для поддержки инфраструктуры экстренных вычислений в целях обеспечения процессов поддержки принятия решений в экстремальных ситуациях. МИТП-Э 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	7
2.1. Условия развертывания программы	8
2.2. Необходимые технические средства управляющей подсистемы МИТП-Э	9
2.3. Необходимые технические средства подсистемы вычислительной инфраструктуры МИТП-Э	10
2.4. Необходимые технические средства подсистемы визуализации МИТП-Э	12
3. ОПИСАНИЕ ЗАДАЧИ	12
3.1. Определение задачи	12
3.2. Методы решения задачи	16
3.2.1. Унифицированное описание прикладных пакетов в МИТП-Э	16
3.2.2. Унифицированное описание композитных приложений в МИТП-Э	19
3.2.3. Описание процессов экстренных вычислений в МИТП-Э	21
3.2.4. Планирование и выполнение экстренных вычислений в МИТП-Э	24
3.2.5. Особенности создания и управления инфраструктуры экстренных вычислений (urgent computing) в распределенных средах с динамическими вычислительными ресурсами	28
3.2.6. Решение типовой задачи предупреждения критических ситуаций с использованием МИТП-Э	30
4. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ	34
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ	36
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ	37

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

1. НАЗНАЧЕНИЕ ПРОГРАММЫ

Технологическая платформа создания и поддержки инфраструктуры экстренных вычислений (urgent computing, UC) в распределенных средах с динамическими вычислительными ресурсами (МИТП-Э) RU.СНАБ.80066-06 01 44 входит в состав многопрофильной инструментально-технологической среды (МИТП) CLAVIRE (Cloud Applications Virtual Environment) RU.СНАБ.80066-06. Она предназначена для поддержки инфраструктуры экстренных вычислений в целях обеспечения процессов поддержки принятия решений в экстремальных ситуациях.

1.1. Функциональное назначение

МИТП-Э представляет собой комплекс программного обеспечения для разработки, настройки и эксплуатации сред экстренных вычислений с целью поддержки принятия решений в критических ситуациях, предназначенный для:

- 1) эффективного управления вычислительными, информационными и программными ресурсами среды экстренных вычислений, включая собственные (выделенные) вычислительные ресурсы центров и ресурсы внешних провайдеров (в том числе глобальных коллаборативных сред);
- 2) создания, исполнения, управления и предоставления сервисов доступа к предметно-ориентированным высокопроизводительным композитным приложениям для поддержки принятия решений в критических ситуациях, функционирующим на основе облака прикладных сервисов в среде экстренных вычислений¹;
- 3) поддержки динамической организации и реорганизации распределенной среды экстренных вычислений путем выбора и подключения вычислительных ресурсов коллаборативной распределенной среды, необходимых для решения пользовательской задачи за заданное время, в условиях неопределенности и неполноты информации о режимах их функционирования исполнения.

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-Э предназначена для создания программной инфраструктуры ситуационных центров, реализующих процессы УС для поддержки принятия решений в критических ситуациях. Под УС понимается процесс организации высокопроизводительных вычислений в целях компьютерного моделирования для поддержки принятия решений в экстремальных ситуациях, в условиях ограниченного времени решений. Для функционирования приложений УС используются как собственные (корпоративные) ресурсы ситуационных центров, так и привлекаемые ресурсы внешних провайдеров (выделенные суперкомпьютеры, Грид-среды, среды облачных вычислений первого поколения). МИТП-Э применима для решения следующих классов задач.

1. Поддержка функционирования распределенных систем раннего предупреждения (Early Warning Systems, EWS) для различных видов критических ситуаций.
2. Обеспечение удаленного доступа к уникальным сервисам и композитным приложениям, необходимым для модельного воспроизведения сценариев развития различных видов критических ситуаций.
3. Обеспечение информационной поддержки процессов принятия решений в ситуационных центрах на основе модельного подхода (SDA, Simulation Driven Approach) в условиях ограниченного времени принятия решений.
4. Выполнение анализа процессов возникновения и развития критических ситуаций на основе компьютерного моделирования в условиях неопределенности и неполноты исходных данных.

1.3. Основные характеристики

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

Специфическая задача МИТП-Э состоит в том, чтобы динамически формировать и контролировать пул ресурсов (данных, прикладных пакетов, вычислительных мощностей) в распределенной среде, достаточный для решения задачи пользователя (в форме задаваемого им сценария) в течение заданного времени. Управление данными ресурсами осуществляется низкоуровневыми средствами МИТП-Э без участия пользователя; после завершения задачи используемые ресурсы освобождаются автоматически. Задача

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

оптимизации нагрузки в распределенной среде в рамках концепции УС решается как обратная: сформировать такой пул ресурсов, который обеспечивал бы решение задачи за время не выше заданного. Это требует анализа вариантов распараллеливания вычислений на всех уровнях иерархии (от ядер и процессоров – до выбора отдельных целевых систем в распределенной среде).

МИТП-Э обеспечивает выполнение следующих функций:

- 1) Поддержка разработки и исполнения композитных приложений на основе динамически организуемых ресурсов распределенной среды экстренных вычислений – набора программно-аппаратных комплексов, принадлежащих разным владельцам, и динамически связываемых через сети общего назначения (Интернет) на основе единых стандартов взаимодействия для решения специфических ресурсоемких задач внешнего пользователя с ограничением на предельное время исполнения.
- 2) Возможность развертывания в существующих коллаборативных распределенных средах, включая существующие инфраструктуры Грид I поколения.
- 3) Поддержка динамической организации и реорганизации распределенной среды экстренных вычислений путем выбора и подключения вычислительных ресурсов коллаборативной распределенной среды, необходимых для решения пользовательской задачи за заданное время, в условиях неопределенности и неполноты информации о режимах их функционирования исполнения.
- 4) Динамическое управление (мониторинг состояния, запуск приложений, передача данных, распределение нагрузки, миграция задач) в автоматическом режиме набором распределенных ресурсов, доступных в распределенной среде экстренных вычислений.
- 5) Автоматическая оптимизация по времени процесса использования доступных вычислительных ресурсов и прикладных сервисов в распределенной среде экстренных вычислений.
- 6) Представление описания композитных приложений в распределенной среде экстренных вычислений на основе цепочек заданий (workflow), обеспечивающих запуск, выполнение, остановку и возобновление работы цепочки заданий в ручном и автоматическом режимах.
- 7) Унифицированный доступ к вычислительным ресурсам распределенной среды экстренных вычислений на основе интерфейсов доступа в составе программно-аппаратной архитектуры SMP, MPP, GPGPU, CBEA.

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

- 8) Поддержка процесса установки и первоначальной конфигурации технологической платформы и ее составных частей на ресурсах, доступных для динамической организации распределенной среды экстренных вычислений.
- 9) Каталогизация входных данных пользователей на основе метаданных.
- 10) Администрирование и контроль работы с дифференцированными правами администраторов в рамках единой политики доступа к ресурсам в распределенной среде экстренных вычислений.
- 11) Модификация знаний, используемых системой, как в ручном, так и в автоматическом режимах.
- 12) Функционирование сервисов резервирования и отката исправлений для пользовательских данных в удаленном хранилище в составе распределенной среды экстренных вычислений.
- 13) Функционирование механизмов конвертирования данных между различными прикладными сервисами, по заданию пользователя.
- 14) Функционирование сервисов, обеспечивающих интерфейс с многофункциональными и предметно-ориентированными системами визуализации результатов расчетов в системах виртуальной реальности типа 3D-Wall.

1.4. Ограничения, накладываемые на область применения

Специфика использования распределенной среды УС накладывает на применение МИТП-Э следующие ограничения:

- 1) в состав МИТП-Э не входят прикладные пакеты; они регистрируются в МИТП-Э исходя из специфики деятельности конкретного ситуационного центра;
- 2) идентификация и аутентификация корпоративных пользователей происходит на основе единого сертификата УС, как для Грид-систем, так и для собственных ресурсов (что допускается не во всех существующих Грид-средах);
- 3) пользователь не имеет возможности самостоятельно выбирать целевые ресурсы (планирование осуществляется внутренними механизмами МИТП-Э);
- 4) возможность обеспечения расчетного времени принятия решений зависит от доступности ресурсов коллаборативной среды и политики реализации УС (включая права на принудительное завершение задач других пользователей для УС-приложений).

2. УСЛОВИЯ ПРИМЕНЕНИЯ

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

2.1. Условия развертывания программы

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

Развертывание МИТП-Э 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 информационного портала RU.СНАБ.80066-06 01 31;
- компонент доступа к вычислительным ресурсам RU.СНАБ.80066-06 01 38.

Для развертывания компонентов МИТП-Э 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

RU.СНАБ.80066-06 31 05 Ошибка! Источник ссылки не найден.

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-Э должен быть установлен сервер баз данных: MongoDB версии 1.6.5. В ходе установки и настройки используются стандартные конфигурации указанных программных средств, не требующие отдельной настройки. После установки необходимо осуществить запуск сервера баз данных для локального использования (localhost). СУБД MongoDB используется компонентами CLAVIRE/Ginger RU.СНАБ.80066-06 01 21 – для хранения данных о пользовательских проектах; CLAVIRE/Eventing RU.СНАБ.80066-06 01 23 – для журналирования произошедших в системе событий; CLAVIRE/Monitoring RU.СНАБ.80066-06 01 24 – в качестве хранилища актуальных данных о платформе; CLAVIRE/GateKeeper RU.СНАБ.80066-06 01 26 – для хранения учетных данных пользователей; CLAVIRE/InfraAccess RU.СНАБ.80066-06 01 27 – для хранения данных о зарегистрированных компонентах; CLAVIRE/Provenance RU.СНАБ.80066-06 01 32 – для хранения профилей исполнения композитных приложений; CLAVIRE/Billing RU.СНАБ.80066-06 01 34 – для хранения пользовательских счетов, тарифов и истории операций; CLAVIRE/Storage RU.СНАБ.80066-06 01 37 – для хранения сервисной информации, используемой центральным модулем хранения данных, а также для хранения метаданных, соответствующей объектам хранения.

Для работы компонента информационного портала RU.СНАБ.80066-06 01 31 требуется установка СУБД MySQL (версии 5.0 или выше) и поддержка web-сервером интерпретатора языка PHP (версии 5.2 или выше).

2.2. Необходимые технические средства управляющей подсистемы МИТП-Э

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

- тип процессоров: Intel-совместимый;
- количество ядер – не менее 4;
- количество процессоров – не менее 2;

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

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

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

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

2.3. Необходимые технические средства подсистемы вычислительной инфраструктуры МИТП-Э

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

1. *Вычислительные кластеры.* Доступные высокопроизводительные ресурсы ситуационного центра и внешних провайдеров; предназначены для установки (в т.ч. автоматической) и последующего использования прикладных сервисов МИТП-Э. Ввиду того что МИТП-Э 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.

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

2. *Корпоративная или глобальная облачная инфраструктура.* Виртуальная вычислительная инфраструктура, конфигурируемая по запросу со стороны МИТП-Э или пользователя системы (при организации доступа к уже сконфигурированным виртуальным ресурсам). Со стороны МИТП-Э работа с облачной инфраструктурой происходит не только на уровне абстрактных прикладных сервисов (реализуются подбор и конфигурация существующих статических ресурсов), но и на уровне абстрактных вычислительных ресурсов (осуществляется динамическая конфигурация ресурса в соответствии с предъявляемыми требованиями).
3. *Грид-инфраструктура первого поколения.* Данный класс ресурсов реализует концепцию виртуальных организаций. При этом специфика доступа к таким ресурсам определяется технологическими особенностями сервисной среды Грид, а также высокой изменчивостью структуры и характеристик этих ресурсов. Тем не менее распределенный характер и высокая суммарная производительность позволяют эффективно задействовать данный класс ресурсов при решении ряда вычислительных задач.
4. *Прочие виды корпоративных ресурсов ситуационного центра* (рабочие станции, серверные ЭВМ и пр.). Интеграция широкого спектра разнородных ресурсов позволяет сформировать инфраструктуру, обеспечивающую исполнение заданий, оптимизированных для различных архитектур. Минимальные требования к корпоративным ресурсам для использования в МИТП-Э:
 - 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.
5. *Специализированные хранилища данных.* МИТП-Э обеспечивает унифицированный доступ как к локальным, так и к распределенным хранилищам и источникам данных при условии их нахождения в локальной сети ситуационного центра совместно с управляющей подсистемой МИТП-Э. Использование инфраструктуры

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

внешних провайдеров для хранения данных МИТП-Э нецелесообразно по соображениям: (а) безопасности и (б) надежности.

2.4. Необходимые технические средства подсистемы визуализации МИТП-Э

МИТП-Э должна обеспечивать интерпретацию результатов расчетов, полученных в процессе моделирования критических ситуаций. Для этого используются специализированные средства трехмерной и интерактивной визуализации, среды виртуальной реальности и пр.

В целом МИТП-Э не накладывает дополнительных ограничений на характеристики системы визуализации (они нивелируются применением инструментального средства RU.СНАБ.80066-06 01 63), кроме требований к управляющей вычислительной системе:

- тип процессоров: Intel-совместимый;
- количество ядер – не менее 2;
- количество процессоров – не менее 1;
- тактовая частота каждого процессора – не ниже 2.0 ГГц;
- оперативная память (на ядро) – не менее 1.0 ГБ;
- дисковая подсистема – не менее 500 ГБ;
- видеокарта NVIDIA Quadro FX3800 (или аналог);
- аппаратная поддержка OpenGL 3.3 и GL_EXT_quad_buffer.

3. ОПИСАНИЕ ЗАДАЧИ

3.1. Определение задачи

Основной задачей МИТП-Э является обеспечение работы пользователя с облачными сервисами и композитными приложениями в среде экстренных вычислений в условиях ограничений на время получения расчетного результата. В целом организация процесса создания и исполнения композитного приложения под управлением МИТП-Э в рамках концепции iPSE сводится к последовательной формализации наборов описаний в терминах потоков заданий (workflow, WF). На первом этапе процесса проектирования композитного приложения создается мета-WF (MWF). Пользователь может осуществлять выбор классов сервисов, которые доступны в облачной среде, и уточнять их по мере ввода дополнительной информации. Указанные пользователем классы сервисов будут использоваться на следующем этапе для подбора конкретных сервисов. На основе MWF

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

создается поток заданий, в котором уже зафиксированы конкретные реализации вычислительных сервисов, однако еще ничего не известно об условиях их выполнения (AWF). Следующим этапом процесса проектирования является построение расписания и создание сценария выполнения в терминах конкретного WF (CWF), который представляет собой поток заданий с полностью определенными блоками. Для блоков действий указаны сервисы и узлы для исполнения, а для блоков данных – конкретные местоположения необходимых данных. Более полное описание процесса создания и исполнения композитного приложения, общего для всех технологических платформ в составе МИТП CLAVIRE приведено в документе RU.СНАБ.80066-06 31 01.

Применительно к МИТП-Э процесс создания и использования приложений UC определяется следующим алгоритмом (рис. 3.1).

1. При прогнозе возникновения экстремальной ситуации пользователи – группа принятия решений ситуационного центра – авторизуется в МИТП-Э через web-портал МИТП-Э (используя соответствующий сертификат МИТП), что дает им возможность доступа к соответствующим ресурсам и сервисам.
2. Члены группы принятия решений (или эксперт-оператор, по согласованию) через соответствующий web-интерфейс могут выбрать конкретные шаблоны расчетных сценариев – композитных приложений в форме WF, а также получить (при необходимости) доступ к технической и эксплуатационной документации. За подбор, каталогизацию и аннотирование сервисов отвечают эксперты ситуационного центра.
3. Для подготовленного описания композитного приложения пользователь конфигурирует условия вычислений: определяет требуемые параметры WF, редактирует (при необходимости) его описание, готовит и загружает в хранилище МИТП-Э входные данные для расчетов. В ряде случаев такие данные могут предоставляться через МИТП-Э автоматизировано (например, через интерфейсы с провайдерами оперативных данных).
4. Группа принятия решений определяет допустимое время вычислений (исходя из времени принятия решений или из уже запущенных сценариев). После этого МИТП-Э предлагает различные варианты организации вычислений, в том числе связанные с привлечением ресурсов Грид и облачных сред.

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

5. Оператор-эксперт запускает задачу на выполнение в среде МИТП-Э. Использование вычислительных ресурсов и сервисов производится с учетом единого сертификата МИТП-Э, с которым в средах внешних провайдеров распределенных вычислений ассоциированы сертификаты «суперпользователей» МИТП-Э. При этом МИТП-Э останавливает выполнение задач других пользователей вычислительных сред (вне процесса UC), если ресурсов недостаточно для достижения заданного времени вычислений.

RU.СНАБ.80066-06 31 05 Ошибка! Источник ссылки не найден.

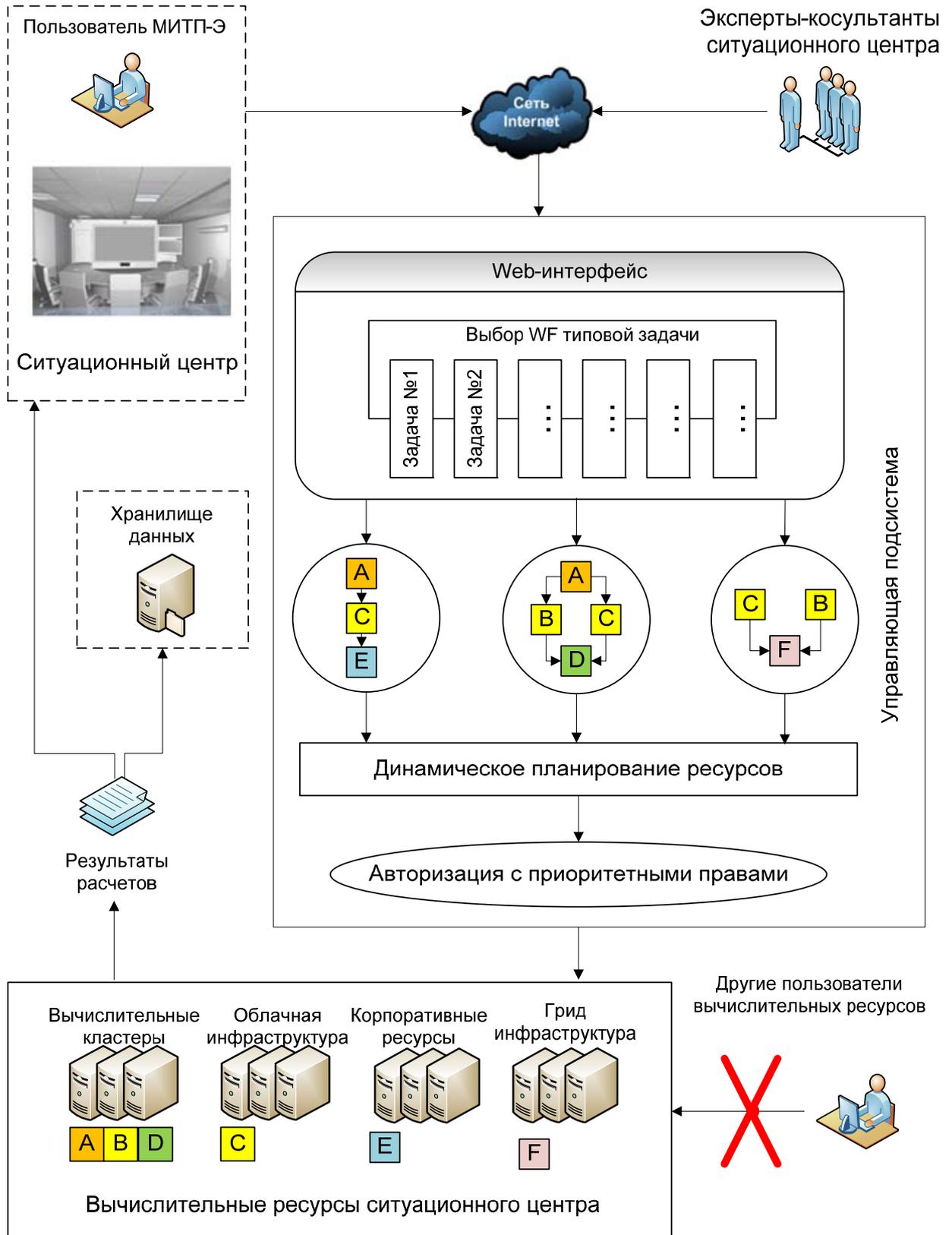


Рисунок 3.1 Принцип функционирования МИТП-Э

6. Оператор-эксперт осуществляет мониторинг процесса исполнения (в форме динамического отображения WF); при этом прогнозируется (в

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

вероятностной форме) время завершения вычислений. Во время расчетов пользователем могут быть построены и запущены другие расчетные сценарии, а также внесены изменения в текущий сценарий (для отдельных классов WF).

7. Когда расчет задачи завершен, результаты помещаются в соответствующее хранилище данных МИТП-Э ситуационного центра. Группа принятия решений может получить доступ к результатам расчетов (в форме выдачи результатов соответствующих пакетов в составе сервисов) через интерфейс МИТП-Э. МИТП-Э также обеспечивает работу специализированных средств визуализации и виртуальной реальности. При этом процесс формирования самих динамических сцен выполняется на стороне ситуационного центра (поскольку он связан со спецификой ПАК виртуальной реальности).

Таким образом, основная задача МИТП-Э 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-Э. 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) обобщенного задания условий выполнения УС-приложений в МИТП-Э;
- 4) планирования и исполнения УС-приложений в МИТП-Э в условиях ограниченного времени принятия решений.

3.2.1. Унифицированное описание прикладных пакетов в МИТП-Э

В МИТП-Э 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

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

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

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

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

RU.СНАБ.80066-06 31 05 **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 "in"
10     filename "arg.txt"
11     place "/"
12     extractor IntegerFileExtractor.new("in")
13     assembler ObjectToSAssembler.new("in")
14   }
15   meta param {
16     name "in"
17     required
18     type int
19     validator lambda { |val, ctx| val > 0 and val < 10000 }
20     validation_error_msg "num have to be in [0; 10000]"
21   }
22   meta param {
23     name "abs_plus_3"
24     required
25     type int
26     evaluator { |ctx| ctx.in.abs + 3 }
27   }
28   cmdline lambda { |ctx| "{0} arg.txt out.txt" }
29 }
30 outputs {
31   auto file {
32     name "output_num"
33     required
34     filename "out.txt"
35     place "/"
36     extractor IntegerFileExtractor.new("out")
37   }
38   auto param {
39     name "out"
40     required
41     display_as "Output number"
42     type int
43   }
44 }
45 prepare_package

```

Рисунок. 3.2 – Фрагмент описания прикладного пакета на языке EasyPackage

Контекст работы представляет собой набор уже вычисленных значений параметров (*ctx*). Файловые параметры дополнительно имеют следующий набор полей (строки 8–14): имя файла, путь до файла, процедура извлечения данных из файла (*extractor*), процедура сборки файла (*assembler*). Процедура сборки файла позволяет создавать входной файл, основываясь на значениях входных параметров. На практике используются стандартные процедуры, например, сборка файла по шаблону (библиотека ERB). Процедура извлечения данных из файла, как правило, применяется для выходных файлов пакета с целью определения значений выходных параметров (строка 12). Базовый набор процедур извлечения значений из файлов и их сборки из параметров можно дополнять за счет

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

написания своих процедур в секции расширений. Последним в файле описания является раздел параметров исполнения, который позволяет при работе с пакетом не учитывать неоднородность ресурсов (различных ОС, архитектур). К параметрам данного раздела относятся: скрипт запуска пакета (точнее, процедура его сборки), командная строка, переменные окружения.

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

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

3.2.2. Унифицированное описание композитных приложений в МИТП-Э

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-Э, 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).

RU.СНАБ.80066-06 31 05 Ошибка! Источник ссылки не найден.

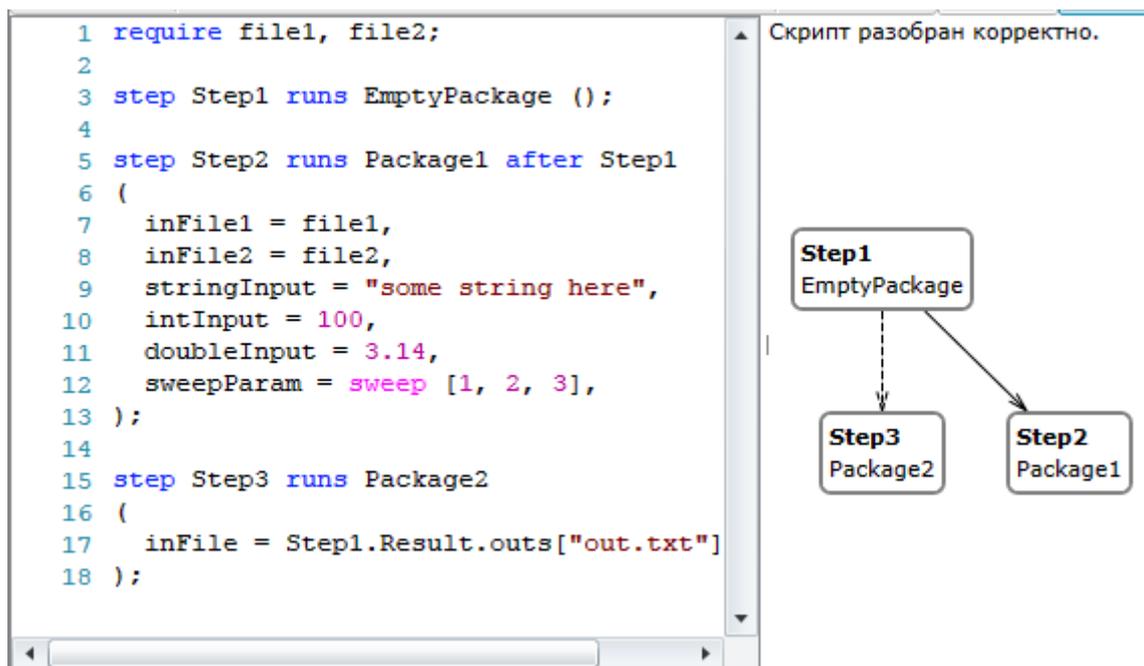


Рисунок 3.3 – Пример описания композитного приложения на языке EasyFlow и его графическое представление в МИТП-Э

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

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

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

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

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

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

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

3.2.3. Описание процессов экстренных вычислений в МИП-Э

Для описания процессов UC в диалекте языка EasyFlow в МИП-Э поддерживается ряд модификаций по сравнению с версией документа RU.СНАБ.80066-06 33 01.

Поддержка WF длительного исполнения (LRWF, Long Running Workflow). Предполагается, что композитное приложение может работать как в пакетном, так и в потоковом режиме, который характеризуется одним или всеми перечисленными ниже особенностями.

1. Время завершения LR-блока заранее не определено. Его завершение может быть вызвано определенным сочетанием событий или данных, поступивших извне.

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

2. LR-блок может ожидать поступления данных извне (например, в виде потока) и реагировать на них определенным образом (например, изменять ход алгоритма или обрабатывать периодически поступающие данные). Далее такие блоки называются потребителями (*consumers*).
3. LR-блок может генерировать данные, которые могут потребляться другими LR-блоками. Такие блоки называются продюсерами (*producers*).
4. LR-блок способен влиять на исполнение других блоков (как обычных, так и LR) с использованием механизма подписки на события.

Стоит отметить, что один и тот же LR-блок может быть одновременно и потребителем, и продюсером (пример: динамическая система визуализации, которая ожидает действий пользователя по управлению сценой и одновременно генерирует выходной поток в виде набора кадров визуализации).

Для реализации описанных выше особенностей работы системы в язык описания EasyFlow внесены следующие дополнения.

1. Введен модификатор, обозначающий, что блок должен быть запущен как LR-блок (*~step*).
2. Введены инструкции подписки на события, генерируемые другими блоками (инструкция *on*), и выполнение действий с блоком при наступлении этих событий.
3. Введены входные параметры типа «поток» по аналогии с файловыми параметрами, которые позволяют связывать потоковые выходы одних LR-блоков с потоковыми входами других (для этого используется оператор '*<-*').
4. Введены механизмы указания ресурсов, предназначенных для запуска конкретного блока, это необходимо, ввиду того что LR-блоки обычно работают с конкретными объектами, внешними по отношению к системе.

В листинге 3.1 в качестве примера представлен пример скрипта на языке EasyFlow, описывающий LRWF, состоящий из двух узлов. Первый узел (*Camera*) периодически генерирует пакеты входных данных, а второй (*Model*) выполняет по ним модельный расчет. В отличие от узлов пакетного запуска, данные узлы помечены ключевым словом *~step*, которое говорит о том, что они являются узлами длительного исполнения (строки 1 и 7). В скрипте задана коммуникационная связь между узлами (строка 13): выходной параметр *init_agent_pos* узла *Camera* связан с одноименным входным параметром узла *Model*.

Листинг 3.1 – Фрагмент описания LRWF на EasyFlow

```

1 ~step Camera runs crowdg (
2   delay = 30,
3   agent_count = 10000,
4   agent_r = 0.4,
5   tracing = true
6 );
7 ~step Model runs crowdm (
8   isLr = true,
9   needPress = false,
10  time_step = 0.2,
11  agent_count = 10000,
12  tracing = true,
13  init_agent_pos <- Camera. init_agent_pos
14 );

```

Для того чтобы указать, что WF выполняется в UC-режиме, используется механизм атрибутов языка EasyFlow. Атрибут – это пара ключ–значение, которая может модифицировать поведение отдельного шага WF или WF в целом. Атрибуты для шагов задаются в квадратных скобках перед шагом (см. листинг 3.2).

Листинг 3.2 – Явное указание UC-режима исполнения блока WF

```

[priority=@high]
[comment="This is the comment"]
step A runs Package
(
  ...
);

```

В приведенном примере для шага А указывается, что его приоритет (priority) при выполнении должен быть высоким (@high). В описание шага входит и другой атрибут: комментарий, который может отображаться в пользовательском интерфейсе. Таким образом, шаг может иметь неограниченное количество атрибутов с тем ограничением, что все атрибуты должны быть уникальными.

Для работы в UC-режиме в МИТП-Э в языке EasyFlow предусмотрены три атрибута: mode, maxDuration и disabled. Первый является атрибутом уровня WF и задает режим исполнения. Для включения urgent-режима требуется установить атрибут mode в значение @urgent. Для указания планировщику максимального времени выполнения WF используется атрибут WF maxDuration, который может принимать значения времени в формате «ЧЧ:ММ:СС». Помимо того, существует атрибут disabled, который управляет активностью отдельных шагов и может принимать два значения: true (шаг активен) и false (шаг неактивен). В первом случае шаг выполняется при интерпретации, во втором – шаг и все зависимые от него шаги исключаются из выполнения.

Таким образом, рассмотренные модификации совокупно с технологиями разделов 3.2.1–3.2.2 обеспечивают унифицированное описание сервисов и композитных приложений в МИТП-Э.

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

3.2.4. Планирование и выполнение экстренных вычислений в МИТП-Э

Для выполнения экстренных вычислений в МИТП-Э могут использоваться различные стратегии распределения вычислительных ресурсов исходя из степени неполноты знаний о характеристиках UC-WF, времени исполнения заданий других пользователей МИТП-Э и приложениях, запускаемых другими пользователями распределенных сред, ресурсы которых используются МИТП-Э. Они могут различаться по следующим параметрам, которые и определяют алгоритмы построения начального плана выполнения, а также алгоритмы и критерии оптимизации текущих планов:

- 1) выбор наилучших либо наихудших (для соответствующих задач планируемого WF в соответствии с полученными оценками) ресурсов. Такой подход позволяет сохранить наиболее мощные ресурсы свободными для возможности их загрузки другими задачами экстренных WF либо, напротив, минимизировать время выполнения WF;
- 2) возможность учета других параллельно выполняющихся (неэкстренных) задач при оценке полного времени выполнения планируемого WF. Это позволяет минимизировать возможные потери, возникающие при прерывании неэкстренных задач (выполняющиеся экстренные задачи других WF принимаются в расчет в любом случае в силу равенства их приоритета по отношению к задачам планируемого WF).

В рамках общего алгоритма планирования в МИТП (см. документ RU.СНАБ.80066-06-31 01) для задач экстренных вычислений используются следующие стратегии.

1. **«Жадный» алгоритм планирования** (Greedy) минимизирует время выполнения WF, выбирая для каждой задачи такой набор вычислительных узлов, при использовании которого полное время ее выполнения будет минимальным. Такой подход позволяет обеспечить быстрое решение важных задач. Критерием оптимальности плана выполнения для этого алгоритма является лишь *минимальное* время выполнения. Потому нет необходимости перебирать различные варианты планов, что сокращает накладные расходы на планирование выполнения WF в МИТП-Э.
2. **Алгоритм предпочтения в первую очередь лучших ресурсов** (Best First) позволяет вписать время выполнения WF в заданные границы, при этом стремясь удержать его как можно ближе к минимально допустимому. При использовании этого алгоритма первоначальный план выполнения WF, относительно которого производится оптимизация, строится по методике «жадного» алгоритма, описанного выше – сначала всем задачам ставятся в соответствие такие наборы узлов, на которых полное время выполнения этих задач будет минимально. Далее в порядке, обратном полученному в результате топологической сортировки задач, производится последовательное

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

назначение задачам WF таких узлов, на которых полное время выполнения этих задач было бы все большим. Этот процесс продолжается до тех пор, пока общее время выполнения плана меньше минимально допустимого для данного WF, либо до тех пор, пока не будет выполнено одно из условий останова.

3. *Алгоритм предпочтения в первую очередь лучших свободных ресурсов* (Best Free First) представляет собой модифицированный алгоритм Best First, однако с той разницей, что в рамках этого алгоритма учитываются запущенные задачи, имеющие не только равный, но и меньший приоритет. За счет этого при оптимизации плана алгоритм в первую очередь рассматривает свободные ресурсы, что дает возможность уменьшить ущерб другим приложениям при запуске WF.
4. *Алгоритм предпочтения в первую очередь худших ресурсов* (Busiest First) позволяет вписать время выполнения WF в заданные границы, при этом стремясь удержать его как можно ближе к максимально допустимому и нанести наименьший ущерб уже выполняющимся задачам, имеющим низкий приоритет. При использовании этого алгоритма первоначальный план выполнения WF, относительно которого производится оптимизация, строится таким образом, что полное время выполнения каждой задачи будет максимальным. Далее в порядке, полученном в результате топологической сортировки задач, производится назначение задачам WF таких сочетаний узлов, на которых полное время выполнения этих задач было бы всё меньшим. В результате общее время выполнения WF последовательно уменьшается. Этот процесс продолжается до тех пор, пока общее время выполнения плана больше, чем максимально допустимое для данного UC-WF, либо до тех пор, пока не будет выполнено одно из условий останова.
5. *Алгоритм предпочтения в первую очередь худших свободных ресурсов* (Busiest Free First) представляет собой модифицированный алгоритм Busiest First, однако с той разницей, что в рамках этого алгоритма учитываются задачи, имеющие не только равный, но и меньший приоритет. За счет этого при оптимизации плана алгоритм в первую очередь рассматривает узлы с уже запущенными задачами.

Выбор стратегии в условиях неопределенности обусловлен сопоставлением интервальных оценок времени выполнения. В том случае, если все интервалы пересекаются, предпочтение отдается стратегии с наименьшим ущербом приложениям других пользователей. Таким образом, данный подход позволяет последовательно (в ходе исполнения блоков WF) осуществлять выбор необходимых ресурсов. Для этого необходимо само описание WF на языке EasyFlow, который может быть

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

параметризован набором входных параметров и файлов, а также параметров исполнения. За разбор скрипта WF и 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 выбранном ресурсе (согласуясь с планом), после чего обрабатывает выходные данные с помощью компонента хранения данных CLAVIRE/Storage RU.СНАБ.80066-06 01 25.

Для подготовки пакета к запуску и обработки его результатов используется компонент CLAVIRE/PackageBase RU.СНАБ.80066-06 01 35, который позволяет сопоставить абстрактные и фактические правила работы с каждым пакетом в составе WF. На рис. 3.4(а) приведена структура CLAVIRE/PackageBase (см. подробнее документ RU.СНАБ.80066-06 13 35). 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.

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

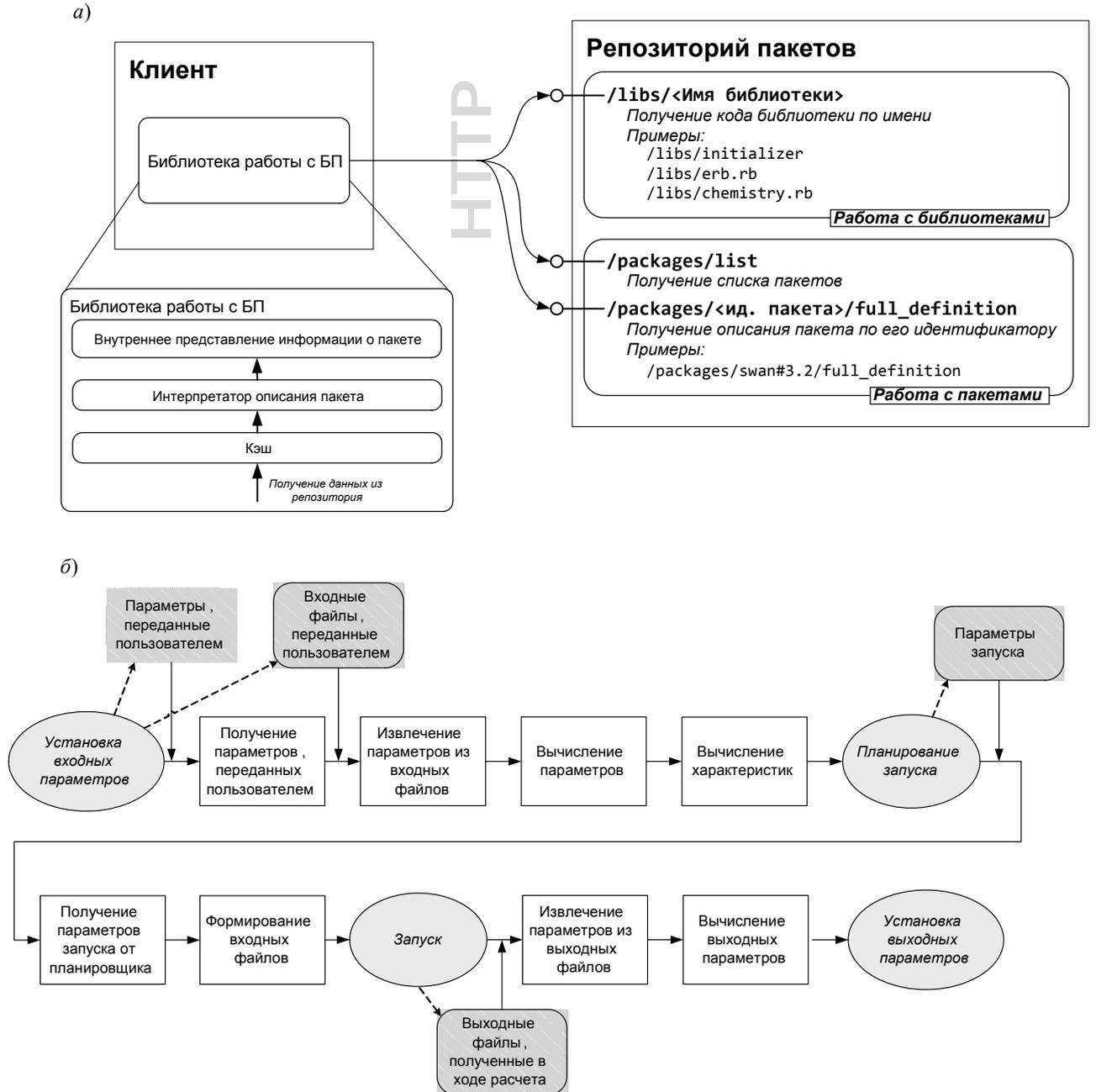


Рисунок 3.4 – Интерпретация и исполнение WF: (а) структура CLAVIRE/PackageBase ; (б) процесс исполнения композитного приложения под управлением МИТП-Э

На рис. 3.4(б) представлен сценарий исполнения композитного приложения под управлением МИТП-Э, 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

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

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

Таким образом, рассмотренные в разделах 3.2.1–3.2.4 методы и технологии обеспечивают решение основной задачи МИТП-Э – поддержку разработки и исполнения композитных приложений на основе динамически организуемых ресурсов распределенной среды экстренных вычислений – набора программно-аппаратных комплексов, принадлежащих разным владельцам, и динамически связываемых через сети общего назначения (Интернет) на основе единых стандартов взаимодействия для решения специфических ресурсоемких задач внешнего пользователя с ограничением на предельное время исполнения.

3.2.5. Особенности создания и управления инфраструктуры экстренных вычислений (*urgent computing*) в распределенных средах с динамическими вычислительными ресурсами

Организация инфраструктуры экстренных вычислений (UC) посредством МИТП-Э имеет следующие особенности:

Исходное ограничение на время принятия решения. Для организации UC ключевым параметром является время получения результатов расчетов. Обеспечение расчетов за время не выше критического является прерогативой самой системы; оно осуществляется за счет выбора вычислительных ресурсов и условий функционирования вычислительных пакетов. В том случае, если при заданных параметрах расчета время получения результатов недостижимо при любых конфигурациях вычислительных средств, система должна рекомендовать пользователю допустимые модификации (например, за счет снижения точности вычислений или использования альтернативных моделей с меньшей ресурсоемкостью).

Автоматизация процесса управления вычислениями. Под управлением МИТП-Э функционируют вычислительные системы принципиально различной архитектуры (выделенные суперкомпьютеры, Грид-среды разного назначения, облачные среды и пр.). Вследствие невозможности «ручной» работы пользователя с различными видами систем в условиях ограниченного времени принятия решений все процессы мониторинга, управления задачами и обмена данными должны быть, по возможности, автоматизированы и скрыты от пользователя. В том случае, когда это невозможно или нецелесообразно, эти процедуры должны быть унифицированы для всех видов вычислительных ресурсов.

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

Организация динамической архитектуры УС в многопользовательской среде.

Специфика МИТП-Э состоит в том, что она использует вычислительные среды коллективного пользования для решения задач, критичных ко времени исполнения и качеству результата. Это требует применения механизмов, которые обеспечивают планирование выполнения УС-расчетов в многопользовательской среде с минимальным (по возможности) ущербом для задач пользователей с более низким приоритетом².

Единая среда «погружения» необходима для совмещения функций *виртуального* ситуационного центра (на основе распределенных ресурсов) и собственно центра принятия решений (локализованной структуры, которая эксплуатирует сервисы МИТП-Э). Решение такой задачи возможно разными способами; однако в рамках МИТП-Э оно реализуется посредством многопользовательских технологий виртуальной реальности типа 3DWall. При этом требуется разработка специализированных средств когнитивной визуализации, обеспечивающих эффективную работу пользователя с применением данной технологии

Высокоуровневый интерфейс общения с системой является отличительной чертой МИТП-Э. Это связано с тем, что, в отличие от предметных систем поддержки принятия решений, пользователь МИТП-Э имеет возможность самостоятельно конструировать (модифицировать) описание процесса принятия решений в различных предметных областях. В рамках системы эта возможность реализована посредством специального предметно-ориентированного языка (Domain Specific Language, DSL).

Поддержка композитных приложений. Важной чертой МИТП-Э является возможность построения композитных приложений из готовых модулей – отдельных прикладных пакетов и источников данных. Такой механизм реализован на основе унифицированных интерфейсов DSL EasyFlow и EasyPackage. Используемый подход не требует модификации кода прикладных пакетов при встраивании их в МИТП-Э; напротив, он позволяет сопрягать пакеты, написанные разными авторами на различных языках программирования, функционирующие под управлением разных операционных систем и использующие разные механизмы распараллеливания.

Дружественное окружение. МИТП-Э обеспечивает весь жизненный цикл проектирования, разработки и использования систем поддержки принятия решений на основе УС. Она поддерживает работу распределенного сообщества пользователей.

² В ряде существующих систем запуск УС-задач приводит к остановке вычислений всех других пользователей.

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

Хранение проектов и данных пользователей осуществляется в централизованном хранилище. Все взаимодействие с системой осуществляется через web-интерфейс, что обеспечивает возможность организации распределенной среды поддержки принятия решений.

С точки зрения потребностей конкретного пользователя, особенности работы с МИТП-Э сводятся к следующему

- 1) Создание и поддержка типовых WF различной сложности специалистами прикладной области.
- 2) Пользователи используют для собственных расчетов базу типовых WF, модифицируя основные входные параметры системы, тем самым оперативно выполняя запуски композитных приложений.
- 3) Пользователю предоставлены максимальные возможности по управлению исполнением WF: определение статуса экстренности вычислений («urgent»), как на уровне всего WF, так и его конкретных шагов; возможность использования приоритетов выполнения шагов; фиксация диапазона времени расчета нижней и верхней границами; возможность исключения шагов из расчета в готовых типовых сценариях, через графическое представление WF.
- 4) Использования в создании композитных приложений возможностей long running WF, для реализации большого класса WF, в том числе ориентированных на реактивное взаимодействие шагов.

3.2.6. Решение типовой задачи предупреждения критических ситуаций с использованием МИТП-Э

В качестве характерного композитного приложения для МИТП-Э рассмотрим WF ансамблевого прогноза морских наводнений в Санкт-Петербурге. Для решения данной задачи осуществляется генерация ансамбля полей скорости ветра исходя из метеопрогнозов HIRLAM. Для этого на поля ветра накладывается маска, значения которой сгенерированы по нормальному закону распределения с указанной мерой разброса значений. Полученный ансамбль прогнозов HIRLAM используется для расчета ветрового волнения, а затем – для выполнения 48-часового гидродинамического прогноза полей уровня и течений Балтийского моря. По ансамблевому прогнозу строятся порядковые статистики, описывающие разброс прогнозируемых значений уровня в контрольных точках, по которым определяется факт наступления наводнения.

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

В листинге 3.3 приведен скрипт описания соответствующего композитного приложения на языке на EasyFlow.

Листинг 3.3 – Код композитного приложения ансамблевого прогноза морских наводнений в Санкт-Петербурге

```

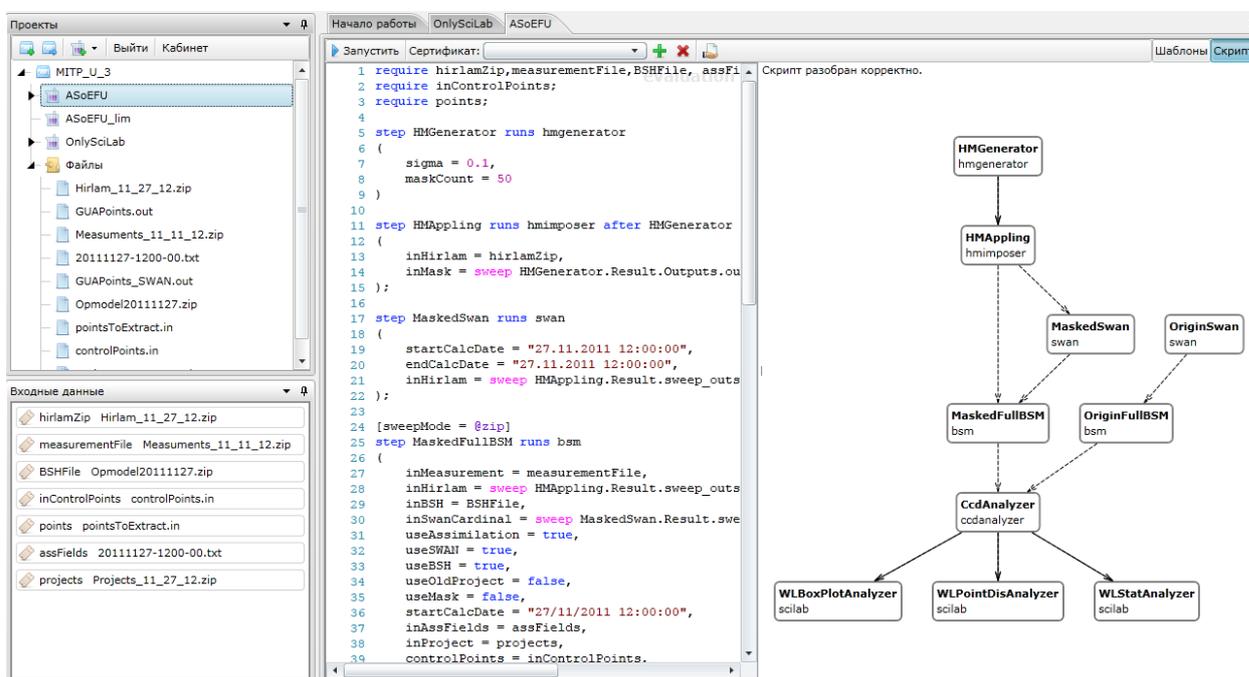
require hirlamZip, measurementFile, BSHFile, assFields, projects;
require inControlPoints, points;

step HMGenerator runs hmgenerator(
  sigma = 0.1,
  maskNum = 10
)
step HMAppling runs hmimposer after HMGenerator (
  inHirlam = hirlamZip,
  inMaskZip = sweep [HMGenerator.Result.masks]
);
step MaskedSwan runs swan(
  startCalcDate = "27.11.2011 12:00:00",
  endCalcDate = "27.11.2011 12:00:00",
  inHirlam = sweep HMAppling.Result.sweep_outs["MaskedHirlam.zip"]
);
[sweepMode = @zip]
step MaskedFullBSM runs bsm(
  inMeasurement = measurementFile,
  inHirlam = sweep HMAppling.Result.sweep_outs["MaskedHirlam.zip"],
  inBSH = BSHFile,
  inSwanCardinal = sweep MaskedSwan.Result.sweep_outs["swanCardinal.zip"],
  useAssimilation = true,
  useSWAN = true,
  useBSH = true,
  useOldProject = false,
  useMask = false,
  startCalcDate = "27/11/2011 12:00:00",
  inAssFields = assFields,
  inProject = projects,
  controlPoints = inControlPoints,
  deleteDirs = true
);
step OriginSwan runs swan(
  startCalcDate = "27.11.2011 12:00:00",
  endCalcDate = "27.11.2011 12:00:00",
  inHirlam = hirlamZip
);
step OriginFullBSM runs bsm(
  inMeasurement = measurementFile,
  inHirlam = hirlamZip,
  inBSH = BSHFile,
  inSwanCardinal = OriginSwan.Result.outs["swanCardinal.zip"],
  useAssimilation = true,
  useSWAN = true,
  useBSH = true,
  useOldProject = false,
  useMask = false,
  startCalcDate = "27/11/2011 12:00:00",
  inAssFields = assFields,
  inProject = projects,
  controlPoints = inControlPoints,
  deleteDirs = true
);
step CcdAnalyzer runs ccdanalyzer (
  OriginCcd = OriginFullBSM.Result.outs["ccd.zip"],
  CcdZip = MaskedFullBSM.Result.sweep_outs["ccd.zip"],
  pointsToExtract = points
);

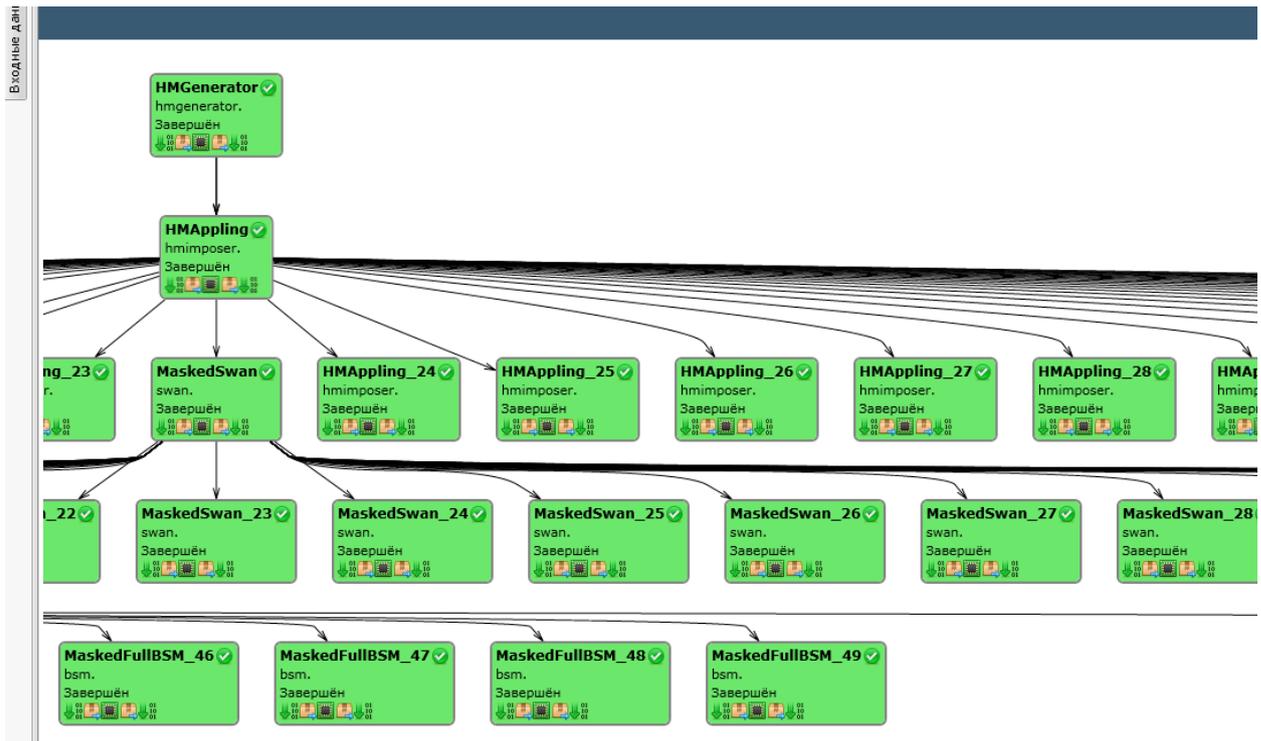
```

```
step WaterLevelVisualization runs sciLab after CcdAnalyzer(
);
```

На рис. 3.5(a) приведено окно интерпретации композитного приложения в МИТП-Э с указанием всех шагов выполнения моделирования, а на рис. 3.5(б) – результат исполнения WF в среде экстренных вычислений. Видно, что для снижения времени вычислений МИТП-Э проводит распараллеливание вычислений на трех уровнях. Выполнение последующих блоков начинается сразу при поступлении первой порции данных от предыдущих, не дожидаясь окончания всего этапа.



(a)



(б)

Рисунок 3.5 – Разработка и исполнение композитного приложения ансамблевого прогноза наводнений в МИТП-Э: (а) окно интерпретации WF, (б) результат исполнения WF

В качестве демонстрации результатов расчетов МИТП-Э на рис. 3.6 приведены результаты расчетов ансамблей уровня воды в пункте Горный институт для прогнозируемой наводненческой ситуации 28 ноября 2011 г. (прогноз без закрытия затворов Комплекса защитных сооружений). Видно, что распределения получаемого ансамбля уровня воды отличны от нормального, причем максимальная плотность распределения смещена в область занижения прогноза. Наибольший разброс значений уровня воды наблюдается при существенных подъемах и спадах при наложении друг на друга отраженных и прямых волн. Прогноз более чувствителен к неопределенности исходных данных в плане предсказания скорости наступления наводнения, чем его пика.

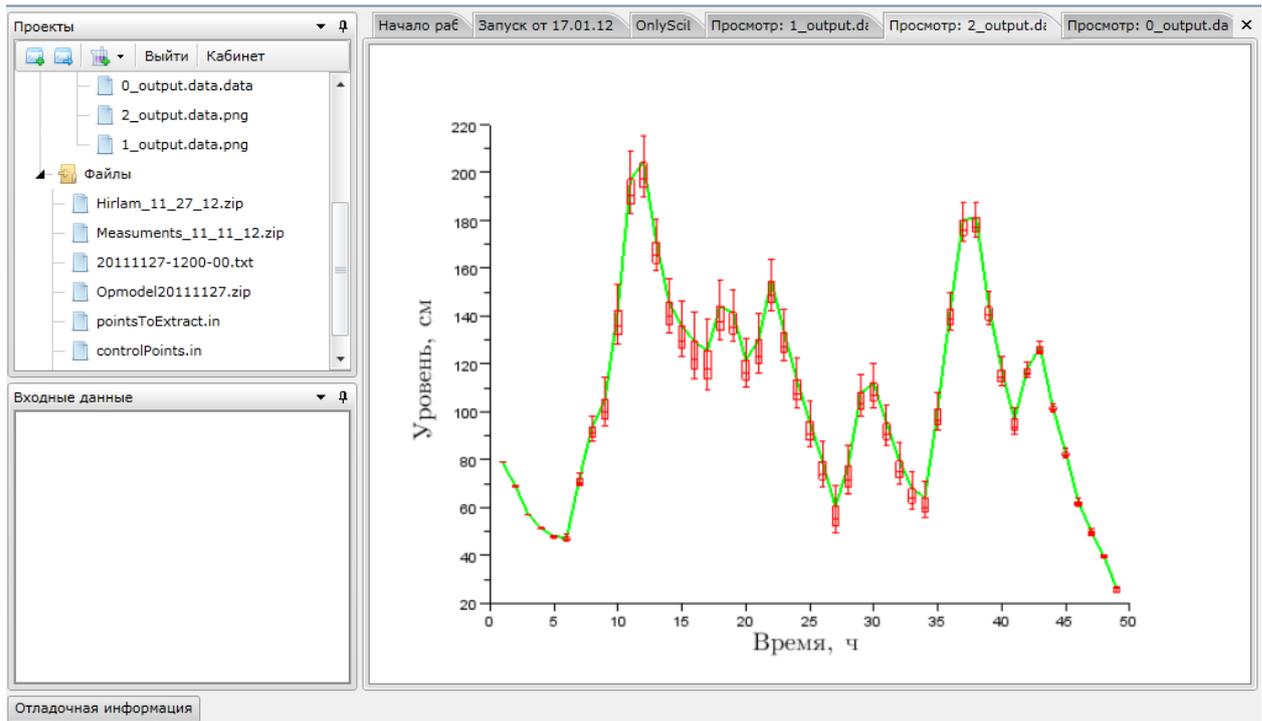


Рисунок 3.6 – Результаты ансамблевого прогноза наводнения 28 ноября 2011 г. (при открытых затворах Комплекса защитных сооружений) в МИТП-Э

Подробное описание рассмотренной задачи и композитного приложения приведено в документе RU.СНАБ.80066-06 13 55.

4. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

Как видно из примера в разделе 3.2.5, для работы с МИТП-Э не требуются специальные виды 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, или в скрипте композитного приложения. В случае несоответствия данных условиям их использования будет выдано соответствующее системное сообщение. Для их приведения к общему формату используется технология описания на основе языка EasyPackage (раздел 3.2.1).

В качестве выходных данных МИТП-Э 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

RU.СНАБ.80066-06 31 05 **Ошибка! Источник ссылки не найден.**

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-Э	Технологическая платформа создания и поддержки инфраструктуры экстренных вычислений в распределенных средах с динамическими вычислительными ресурсами (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
EWS	Early Warning System, система раннего предупреждения
iPSE	Intelligent Problem Solving Environment, концепция
LRWF	Long Running Workflow, WF длительного исполнения
MWF	Мета-WF
SaaS	Software as a Service, модель облачных вычислений
SDA	Simulation Driven Approach, подход на основе компьютерного моделирования
UC	Urgent Computing, экстренные вычисления
WF	Поток заданий, workflow

