

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

СОГЛАСОВАНО

Генеральный директор  
ЗАО «АйТи»

  
Бакиев О.Р.  
"29" сентября 2011 г.

УТВЕРЖДАЮ

Ректор НИУ ИТМО

  
Васильев В.Н.  
"29" сентября 2011 г.

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

ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА СОЗДАНИЯ  
МЕЖОРГАНИЗАЦИОННЫХ (КОЛЛАБОРАТИВНЫХ) КОМПЗИТНЫХ  
ПРОГРАММНЫХ ПРИЛОЖЕНИЙ (МИТП-М)

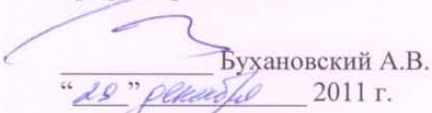
Описание применения

ЛИСТ УТВЕРЖДЕНИЯ

RU.СНАБ.80066-06 31 03-ЛУ

Представители  
Организации-разработчика

Руководитель разработки,  
профессор НИУ ИТМО

  
Бухановский А.В.  
"29" 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

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

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

---

**УТВЕРЖДЕН  
RU.СНАБ.80066-06 31 03-ЛУ**

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

**ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА СОЗДАНИЯ  
МЕЖОРГАНИЗАЦИОННЫХ (КОЛЛАБОРАТИВНЫХ) КОМПОЗИТНЫХ  
ПРОГРАММНЫХ ПРИЛОЖЕНИЙ (МИП-М)**

**Описание применения**

**RU.СНАБ.80066-06 31 03**

**ЛИСТОВ 33**

<b>Инв.№ подл.</b>	
<b>Подп. и дата</b>	
<b>Взам.инв.№</b>	
<b>Инв.№ дубл.</b>	
<b>Подп. и дата</b>	

RU.СНАБ.80066-06 31 03 **Ошибка! Источник ссылки не найден.**

## **АННОТАЦИЯ**

Документ содержит описание применения технологической платформы создания межорганизационных (коллаборативных) композитных программных приложений (МИТП-М) RU.СНАБ.80066-06 01 42. Технологическая платформа МИТП-М входит в состав многопрофильной инструментально-технологической среды (МИТП) CLAVIRE (Cloud Applications Virtual Environment) RU.СНАБ.80066-06. Она предназначена для создания распределенных центров обработки данных, реализующих сервисы в рамках концепции SaaS и AaaS на основе коллаборативных сред, включая Грид-системы первого поколения. МИТП-М разработана в ходе выполнения проекта «Создание распределенной вычислительной среды на базе облачной архитектуры для построения и эксплуатации высокопроизводительных композитных приложений» (Договор № 21057 от 15 июля 2010 г., шифр 2010-218-01-209) в рамках реализации постановления Правительства РФ № 218 «О мерах государственной поддержки развития кооперации российских высших учебных заведений и организаций, реализующих комплексные проекты по созданию высокотехнологичного производства».

**СОДЕРЖАНИЕ**

1. НАЗНАЧЕНИЕ ПРОГРАММЫ .....	4
1.1. Функциональное назначение .....	4
1.2. Область применения .....	4
1.3. Основные характеристики .....	5
1.4. Ограничения, накладываемые на область применения .....	6
2. УСЛОВИЯ ПРИМЕНЕНИЯ .....	7
2.1. Условия развертывания программы .....	7
2.2. Необходимые технические средства управляющей подсистемы МИТП-М .....	8
2.3. Необходимые технические средства подсистемы вычислительной инфраструктуры МИТП-М .....	9
3. ОПИСАНИЕ ЗАДАЧИ .....	10
3.1 Определение задачи .....	10
3.2. Методы решения задачи .....	13
3.2.1. Унифицированное описание прикладных пакетов в МИТП-М .....	13
3.2.2. Унифицированное описание композитных приложений в МИТП-М .....	17
3.2.3. Планирование и оптимизация процесса исполнения композитного приложения в среде Грид .....	19
3.2.4. Организация процесса исполнения композитного приложения в МИТП-М .....	23
3.2.5. Особенности создания и управления межорганизационными (коллаборативными) композитными программными приложениями .....	26
3.2.6. Решение типовой задачи компьютерного моделирования сложных систем с помощью коллаборативного приложения с использованием МИТП-М .....	27
4. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ .....	31
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ .....	32
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ .....	33

## **1. НАЗНАЧЕНИЕ ПРОГРАММЫ**

Технологическая платформа создания межорганизационных (коллаборативных) композитных программных приложений (МИТП-М) RU.СНАБ.80066-06 01 42 входит в состав многопрофильной инструментально-технологической среды (МИТП) CLAVIRE (Cloud Applications Virtual Environment) RU.СНАБ.80066-06. Она предназначена для создания распределенных центров обработки данных, реализующих сервисы в рамках концепции SaaS и AaaS на основе коллаборативных сред, включая Грид-системы первого поколения.

### **1.1. Функциональное назначение**

МИТП-М представляет собой комплекс программного обеспечения для разработки, настройки и эксплуатации сред распределенных вычислений коллаборативного уровня на основе Грид, предназначенный для:

- 1) эффективного управления вычислительными, информационными и программными ресурсами Грид-сред в рамках модели облачных вычислений GaaS;
- 2) создания, исполнения, управления и предоставления сервисов доступа к предметно-ориентированным высокопроизводительным композитным приложениям, функционирующим на основе облака распределенных прикладных сервисов в коллаборативной среде Грид<sup>1</sup>;
- 3) обеспечения функционирования программно-аппаратного комплекса (ПАК) поддержки инфраструктуры предметно-ориентированных облачных вычислений в различных предметных областях на основе ресурсов коллаборативной распределенной среды Грид.

### **1.2. Область применения**

Технологическая платформа МИТП-М предназначена для создания межорганизационных (коллаборативных) композитных программных приложений в уже существующих и перспективных Грид-средах. Она применима для решения следующих классов задач.

---

<sup>1</sup> 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.

RU.СНАБ.80066-06 31 03 **Ошибка! Источник ссылки не найден.**

1. Численное моделирование сложных неоднородных систем и объектов с применением набора прикладных пакетов, предоставляемых различными участниками коллаборативной среды.
2. Обеспечение удаленного доступа к уникальным сервисам и композитным приложениям в области компьютерного моделирования и обработки данных, доступным в среде Грид.
3. Поддержка работы распределенных коллективов исследователей и разработчиков над вычислительно-ресурсоемкими задачами, требующими использования коллаборативных ресурсов Грид.
4. Численное моделирование в мультидисциплинарных и многомасштабных задачах, требующее совместного применения и сопряжения пакетов, используемых в различных предметных областях разными группами специалистов.

### **1.3. Основные характеристики**

Технологическая платформа МИТП-М 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 коллаборативных вычислительных ресурсов в рамках коллаборативных и глобальных Грид-сред.

МИТП-М является инструментальной надстройкой над Грид-средами первого поколения, которая обеспечивает высокоуровневый интерфейс к прикладным пакетам, сервисам и композитным приложениям на их основе. МИТП-М функционирует на ресурсах организации-провайдера услуг МИТП-М (в ряде случаев в качестве такой организации может выступать сам оператор Грид-среды). МИТП-М позволяет реализовать модели SaaS и AaaS в рамках «публичного» облака, создаваемого на базе Грид. В целом она обеспечивает выполнение следующих функций:

- 1) Поддержка разработки и исполнения композитных приложений на основе ресурсов коллаборативных распределенных сред – набора программно-аппаратных комплексов, принадлежащих разным владельцам, связанных через сети общего назначения (Интернет) на основе единых стандартов взаимодействия, и предназначенных для решения задач, определяемых потребностями пользователей.
- 2) Динамическое управление (мониторинг состояния, запуск приложений, передача данных, распределение нагрузки, миграция задач) в автоматическом режиме

RU.СНАБ.80066-06 31 03 **Ошибка! Источник ссылки не найден.**

набором распределенных ресурсов, доступных в коллаборативной распределенной среде.

- 3) Автоматическая оптимизация по времени процесса использования доступных вычислительных ресурсов и прикладных сервисов в коллаборативной распределенной среде..
- 4) Представление описания композитных приложений в коллаборативной распределенной среде на основе цепочек заданий (workflow), обеспечивающих запуск, выполнение, остановку и возобновление работы цепочки заданий в ручном и автоматическом режимах.
- 5) Унифицированный доступ к вычислительным ресурсам коллаборативной распределенной среды на основе интерфейсов доступа в составе программно-аппаратной архитектуры SMP, MPP, GPGPU, СВЕА.
- 6) Поддержка процесса установки и первоначальной конфигурации технологической платформы и ее составных частей на ресурсах коллаборативной распределенной среды.
- 7) Поддержка многопользовательского режима в коллаборативной распределенной среде.
- 8) Квотирование, биллинг и тарификация использования ресурсов коллаборативной распределенной среды.
- 9) Каталогизацию входных данных пользователей на основе метаданных.
- 10) Администрирование и контроль работы с дифференцированными правами администраторов в рамках многоуровневой политики доступа к ресурсам коллаборативной распределенной среды.
- 11) Модификация знаний, используемых системой, как в ручном, так и в автоматическом режимах.
- 12) Функционирование сервисов резервирования и отката изменений пользовательских данных в удаленном хранилище в составе коллаборативной распределенной среды.
- 13) Функционирование механизмов конвертирования данных между различными прикладными сервисами, по заданию пользователя.
- 14) Использование ресурсов существующих инфраструктур Грид I поколения.

#### **1.4. Ограничения, накладываемые на область применения**

Специфика использования среды Грид накладывает на применение МИТП-М следующие ограничения:

RU.СНАБ.80066-06 31 03 **Ошибка! Источник ссылки не найден.**

- 1) в состав МИТП-М не входят прикладные пакеты; они регистрируются в МИТП-М исходя из наличия в используемой среде Грид;
- 2) идентификация и аутентификация корпоративных пользователей происходит с использованием персональных сертификатов Грид;
- 3) ограничены права пользователя по целевому выбору ресурса (предоставление ресурсов регулируется политикой оператора Грид).

## **2. УСЛОВИЯ ПРИМЕНЕНИЯ**

### **2.1. Условия развертывания программы**

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

Развертывание МИТП-М 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;



RU.СНАБ.80066-06 31 03 **Ошибка! Источник ссылки не найден.**

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

Дополнительно для функционирования МИТП-М 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.

## **2.2. Необходимые технические средства управляющей подсистемы МИТП-М**

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

- тип процессоров: Intel-совместимый;

RU.СНАБ.80066-06 31 03 **Ошибка! Источник ссылки не найден.**

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

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

Для функционирования компонента развертывания и конфигурирования 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-М отдельные компоненты могут функционировать на разных вычислительных системах в рамках общей локальной сети.

### **2.3. Необходимые технические средства подсистемы вычислительной инфраструктуры МИТП-М**

Комплекс технических средств подсистемы вычислительной инфраструктуры МИТП-М включает ресурсы в составе глобальных сред распределенных вычислений.

1. *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. Минимальные требования к ресурсам в Грид для использования в МИТП-М:

- архитектура: SMP, MPP, GPGPU, СВЕА;

RU.СНАБ.80066-06 31 03 **Ошибка! Источник ссылки не найден.**

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

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

### 3. ОПИСАНИЕ ЗАДАЧИ

#### 3.1 Определение задачи

Основной задачей МИТП-М является обеспечение работы пользователя с облачными сервисами и композитными приложениями в коллаборативной распределенной вычислительной среде на основе Грид. Организация процесса создания и исполнения композитного приложения под управлением МИТП-М 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 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 – конкретные местоположения

RU.СНАБ.80066-06 31 03 **Ошибка! Источник ссылки не найден.**

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-М процесс создания распределенных корпоративных программных приложений определяется следующим алгоритмом (рис. 3.2).

1. Пользователь МИТП-М (авторизованный пользователь Грид) авторизуется в МИТП-М через портал оператора Грид (используя соответствующий сертификат), что дает ему возможность доступа к соответствующим ресурсам и сервисам в рамках Грид-среды.
2. Пользователь МИТП-М через соответствующий web-интерфейс может выбрать конкретные сервисы, а также получить (при необходимости) доступ к технической и эксплуатационной документации. Сервисы разрабатываются и предоставляются различными пользователями Грид-среды (пользователь МИТП-М может использовать как свои, так и «чужие» сервисы). За каталогизацию и аннотирование сервисов отвечает оператор среды Грид; МИТП-М обеспечивает доступ к соответствующей информационной системе.
3. Выбрав необходимые сервисы, пользователь средствами МИТП-М конструирует соответствующее композитное приложение в форме описания WF. При этом может использоваться как графическая, так и текстовая форма представления композитных приложений. В ряде случаев пользователь может не конструировать приложение, а использовать готовые WF, разработанные другими пользователями, из публичного репозитория МИТП-М.
4. Для подготовленного описания композитного приложения пользователь конфигурирует условия вычислений: определяет требуемые параметры WF, редактирует (при необходимости) его описание, готовит и загружает в хранилище МИТП-М входные данные для расчетов.

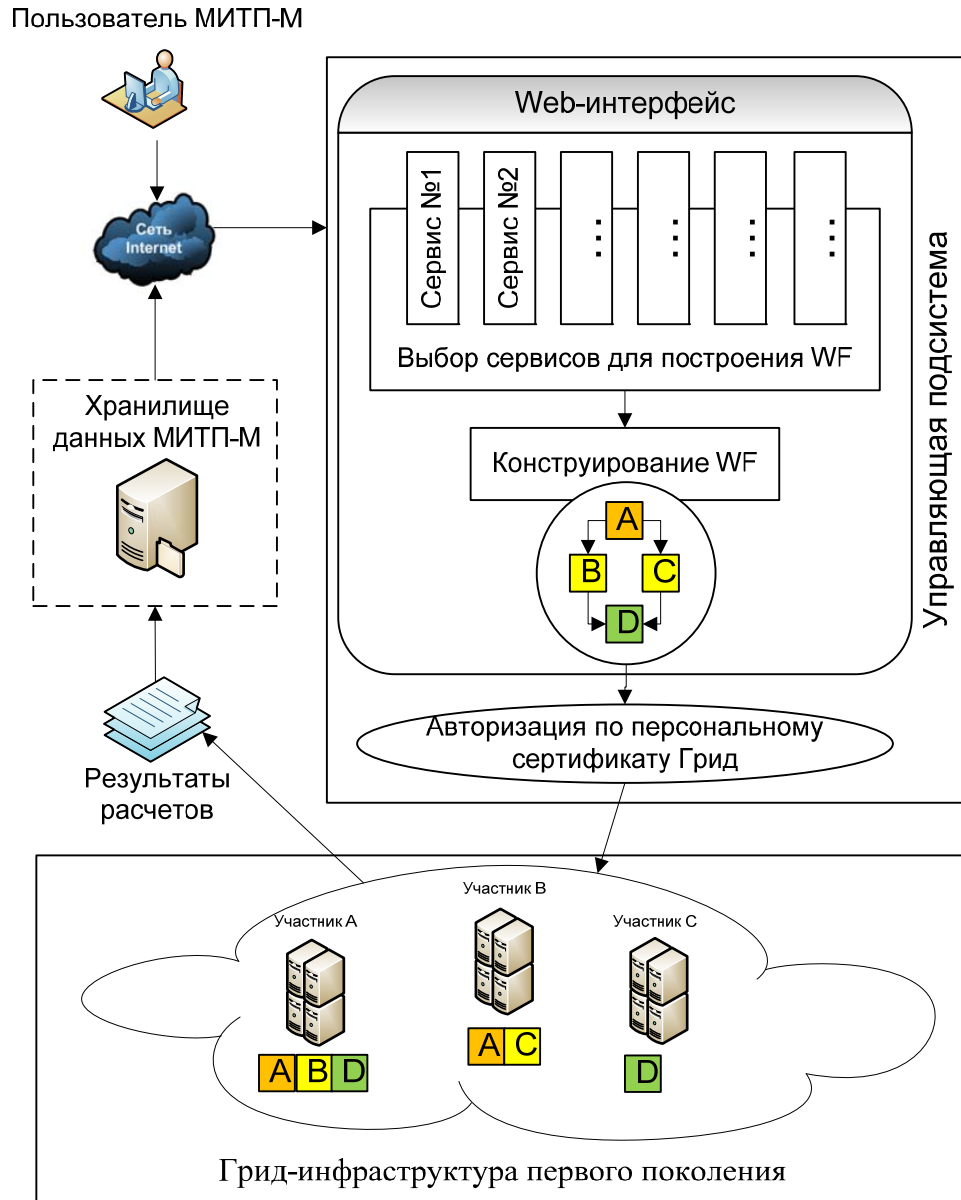


Рисунок 3.1 Принцип функционирования МИТП-М

5. Пользователь определяет режим исполнения задачи в МИТП-М (утверждает предлагаемые ему варианты) в соответствии с требованиями к временным характеристикам расчета и текущими возможностями среды Грид.
6. Пользователь запускает задачу на выполнение в среде МИТП-М. Использование вычислительных ресурсов и сервисов производится с учетом соответствующего персонального Грид-сертификата пользователя, который выдается и контролируется службой авторизации оператора Грид-среды (провайдер МИТП-М только передает соответствующие данные авторизационному центру). Реальное распределение задач по ресурсам выдается пользователю в виде справочной информации.

RU.SNAB.80066-06 31 03 **Ошибка! Источник ссылки не найден.**

7. Пользователь (при необходимости) осуществляет мониторинг процесса исполнения (в форме динамического отображения WF); при этом прогнозируется (в вероятностной форме) время завершения вычислений. Пользователь может на время расчетов завершить рабочую сессию (закрыть интерфейс доступа к МИТП-М) и начать ее только при необходимости использования результатов (необязательно в тот момент, когда расчеты завершены).
8. Когда расчет задачи завершен, результаты помещаются из среды Грид в соответствующее хранилище данных МИТП-М; пользователю отправляется соответствующее уведомление. Пользователь может получить доступ к результатам расчетов (в форме выдачи результатов соответствующих прикладных пакетов) через интерфейс МИТП-М.

Таким образом, основная задача МИТП-М 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-М. 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 среде Грид;
- 3) планирования исполнения композитных приложений на доступных ресурсах среды Грид;
- 4) организации всего процесса исполнения композитного приложения в Грид под управлением МИТП-М.

#### ***3.2.1. Унифицированное описание прикладных пакетов в МИТП-М***

В МИТП-М простейшая форма WF представляет собой описание исполнения одного вычислительного пакета на ресурсе корпоративной вычислительной среды, с загрузкой входных и получением выходных данных. Однако унифицированное описание этого действия осложнено тем, что разные вычислительные пакеты применяют свою стратегию работы с данными (использование конфигурационного файла, командной строки аргументов, переменных окружения, проектов, хранящихся в структуре

RU.СНАБ.80066-06 31 03 **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-М 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 03 **Ошибка! Источник ссылки не найден.**

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.



```

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

### 3.2.2. Унифицированное описание композитных приложений в МИТП-М

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-М, позволяет упростить процедуру задания композитных приложений. Он предоставляет конечному пользователю гибкие возможности по заданию форм WF, в рамках которых выполняются различные прикладные пакеты, происходит генерация выходных данных, их получение, конвертация и обработка.

Характерной чертой языка является полное абстрагирование от особенностей распределенной вычислительной среды, в которой работает пользователь. Фактически EasyFlow – это высокоуровневый язык описания AWF. Такой подход позволяет описывать саму решаемую задачу, а не способ ее исполнения на конкретной вычислительной архитектуре. На рис. 3.3 приведен пример описания простого AWF, представляющего собой скрипт. Тело скрипта состоит из описания вызовов прикладных пакетов – шагов, которые задаются с помощью директивы *step* и представляют собой узлы графа WF. Для описания каждого шага необходимо задать его имя (в примере это *Step1*, *Step2*, *Step3*), название запускаемого пакета (*EmptyPackage*, *Package1* и *Package2*) и перечень предметных параметров этого пакета (см. раздел 3.2.1).

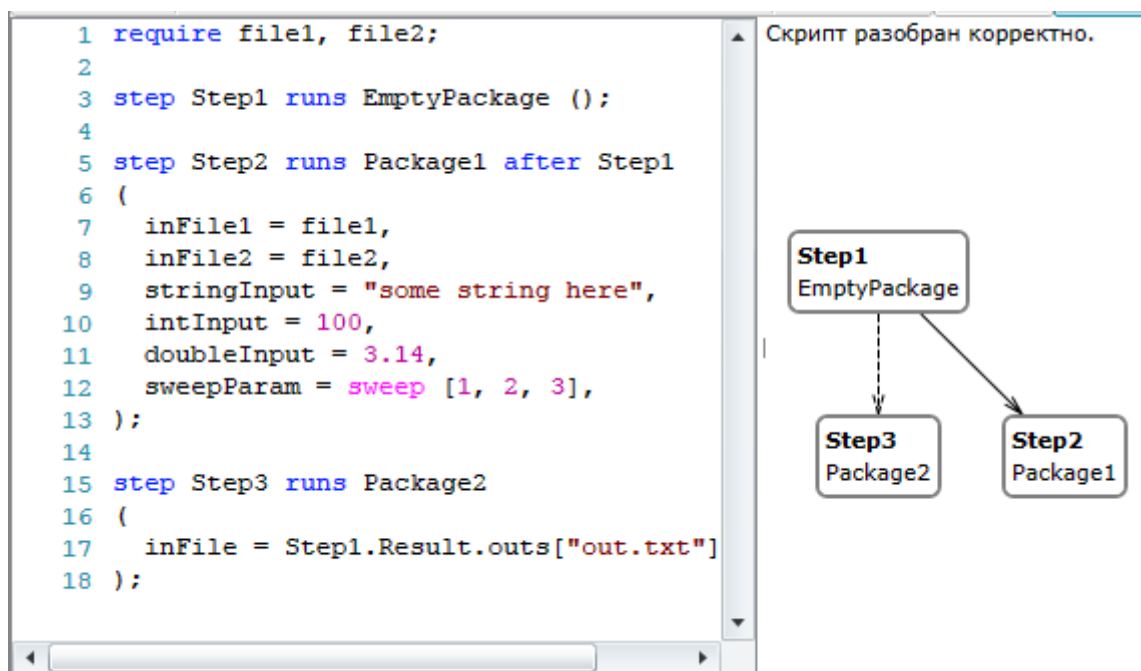


Рисунок 3.3 – Пример описания композитного приложения на языке EasyFlow и его графическое представление в МИТП-М

RU.СНАБ.80066-06 31 03 **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 03 **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 среде Грид***

Для запуска приложений в МИТП-М 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 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-М, 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 03 **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-М 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 03 **Ошибка! Источник ссылки не найден.**

приложения. Однако для работы в среде Грид необходимо учитывать стохастическую изменчивость характеристик вычислительных ресурсов, связанную с коммунальным характером использования инфраструктуры. Для этого вводится вероятностная модель времени исполнения прикладного сервиса, которая может быть представлена выражением:

$$T_{ij} = t_{ij} + \kappa_{ij} T_{0j} / S(\Xi_j, P, \Omega_i). \quad (3.5)$$

Здесь индексы  $i, j$  обозначают целевой ресурс и тип сервиса. Величина  $t$  в (3.5) является характеристикой накладных расходов на запуск сервиса, задаваемой распределением  $\mathfrak{Z}_t(T)$ , и может быть представлена следующим образом:

$$t = t_u(\Omega, \Xi) + t_r(\Omega) + t_w(\Xi), \quad (3.6)$$

где  $t_u$  – накладные расходы на подготовку задания к исполнению в распределенной среде  $\Omega$  с данными  $\Xi$ ,  $t_r(\Omega)$  – внутренние накладные расходы распределенной среды,  $t_w(\Xi)$  – совокупное время подготовки и передачи данных в среде Грид. Таким образом, процедура планирования (3.1) в Грид приобретает вероятностный характер благодаря параметризации величины издержек (3.6). Потому выбор оптимального ресурса для выполнения блока осуществляется на основе сопоставления характерных квантилей распределения  $\mathfrak{Z}_t(T)$ .

Для определения времени работы WF в целом в МИТП-М 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, задаваемые (3.5)–(3.6). 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 (рис. 3.4):

- 1) формализация композитного приложения: формирование структуры абстрактного WF исходя из пользовательского описания, состава данных  $\Xi$  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

RU.СНАБ.80066-06 31 03 **Ошибка! Источник ссылки не найден.**

представления знаний, ассоциированных с элементами WF), а также определение накладных расходов, связанных с членами выражения (3.6);

- 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;

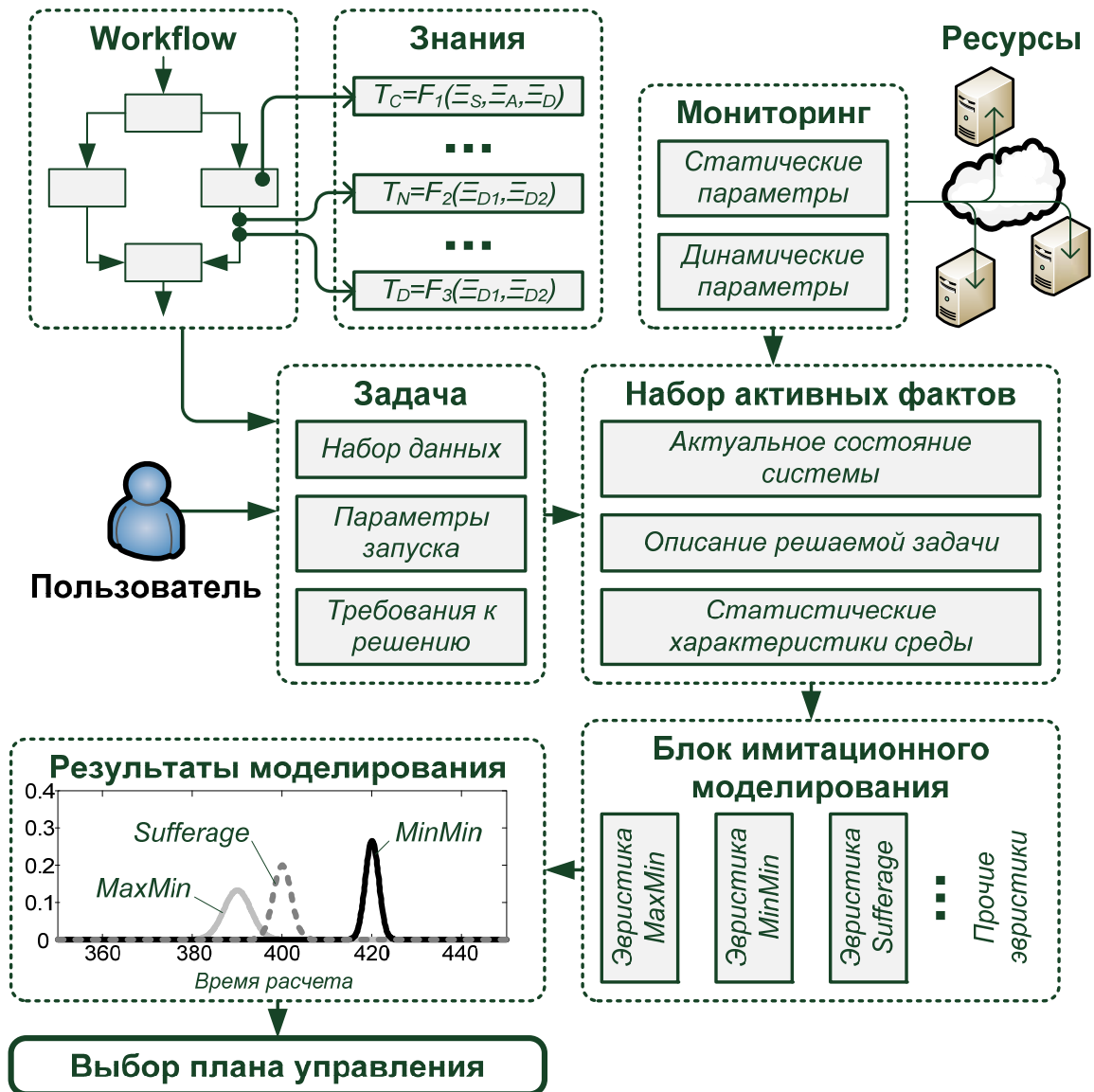


Рисунок 3.4 – Процедура планирования исполнения композитного приложения в МИТП-М

- 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

RU.СНАБ.80066-06 31 03 **Ошибка! Источник ссылки не найден.**

эвристика с минимальным средним временем исполнения и ограничением на разброс в сторону увеличения времени исполнения. 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 критерии их ранжирования. Таким образом, рассмотренный выше подход позволяет совокупно учесть стохастическую изменчивость характеристик исполнения сервисов в Грид и исходные знания о производительности прикладных сервисов в ходе планирования исполнения композитного приложения. В МИТП-М он реализован в компоненте планирования исполнения WF CLAVIRE/Scheduler RU.СНАБ.80066-06 01 28.

#### ***3.2.4. Организация процесса исполнения композитного приложения в МИТП-М***

Рассмотренные выше (разделы 3.2.1–3.2.3) подходы к описанию пакетов, композитных приложений и интерфейсов работы с ними позволяют организовать процесс исполнения приложения в рамках МИТП-М. Композитное приложение представляется в виде скрипта описания WF на языке EasyFlow, который может быть параметризован набором входных параметров и файлов, а также параметров исполнения, т.е. один и тот же WF может быть исполнен для разного набора входных данных, а также в различных условиях исполнения. За разбор скрипта WF и исполнение WF в целом отвечает компонент интерпретации WF CLAVIRE/FlowSystem RU.СНАБ.80066-06 01 20. После получения скрипта он «разбирает» его и преобразует во внутреннее представление (предварительно проверив его корректность). Обработка представления WF производится непрерывно, согласно событийной модели функционирования, т.е. интерпретация WF происходит в рамках цикла обработки поступающих событий. При запуске отдельной задачи происходит интерпретация параметров узла WF и формируется описание запуска задачи, после чего сформированное описание передается в очередь компоненту исполнения WF CLAVIRE/Executor RU.СНАБ.80066-06 01 29. Далее компонент CLAVIRE/Executor подготавливает данные для пакета и производит запуск пакета в среде Грид, после чего обрабатывает выходные данные с помощью компонента хранения данных CLAVIRE/Storage RU.СНАБ.80066-06 01 25.



RU.СНАБ.80066-06 31 03 **Ошибка! Источник ссылки не найден.**

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(a) 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.

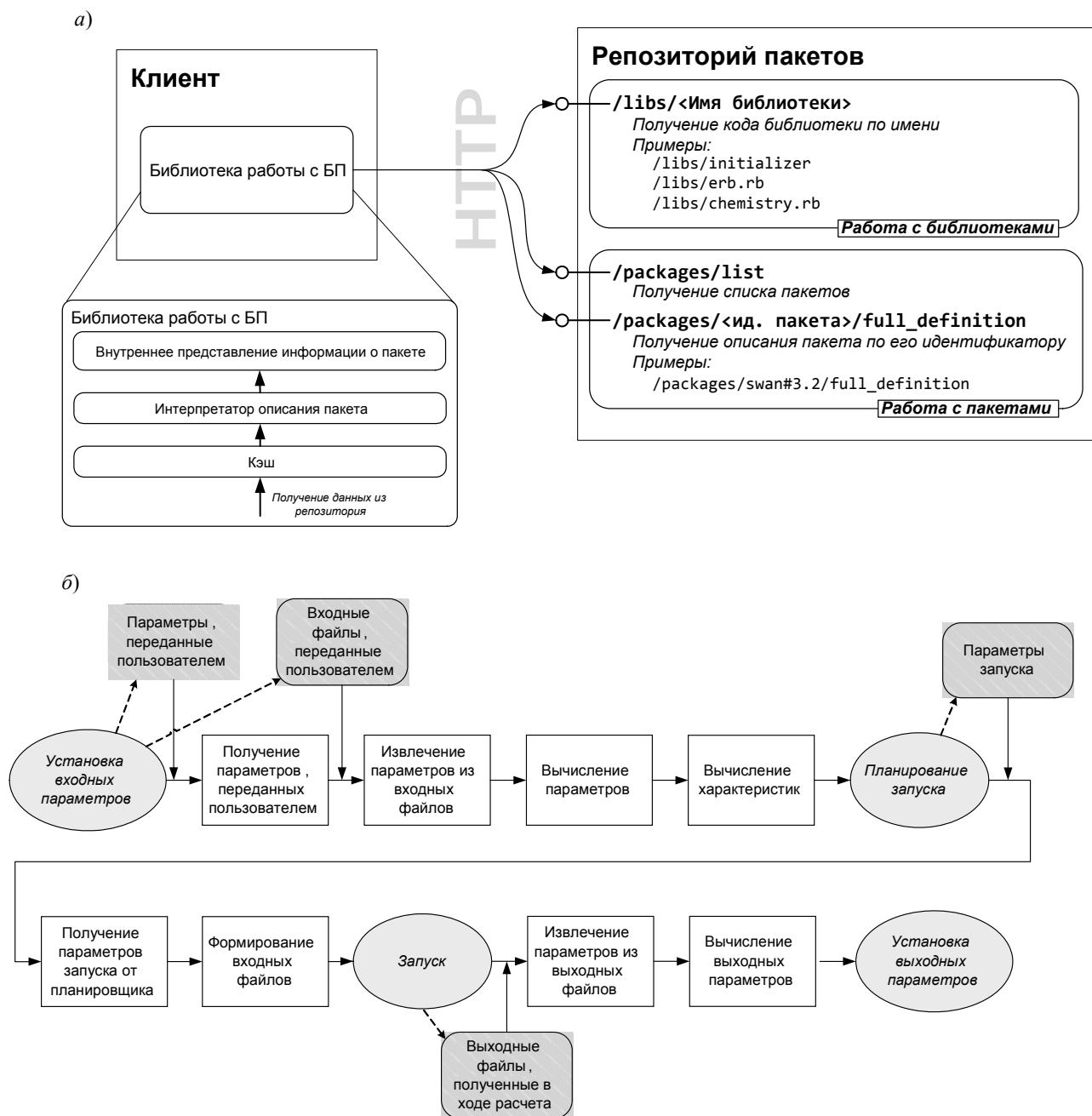


Рисунок 3.5 – Интерпретация и исполнение WF: (а) структура CLAVIRE/PackageBase ; (б) процесс исполнения композитного приложения под управлением МИТП-М

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) представлен сценарий исполнения композитного приложения под управлением МИТП-М, 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).

Специфика управления ресурсами Грид в МИТП-М реализуется посредством механизма *провайдера ресурсов*. Специфика Грид-систем состоит в том, что они обладают собственной системой управления запуском задания (СРКЗ «Пилот» на рис. 3.6). Клиент в данной системе, как и инструменты работы с GridFTP, и средства управления сертификатами для системы безопасности Грид, доступны не для всех операционных систем. Поэтому в качестве «моста» к ресурсам Грид используется виртуальная машина со всеми необходимыми инструментами работы с Грид. Провайдер вычислительных ресурсов Грид осуществляет управление путем подключения к указанной виртуальной машине по протоколу SSH и запуска необходимых программ. Общая схема взаимодействия с инфраструктурой Грид показана на рис. 3.6.

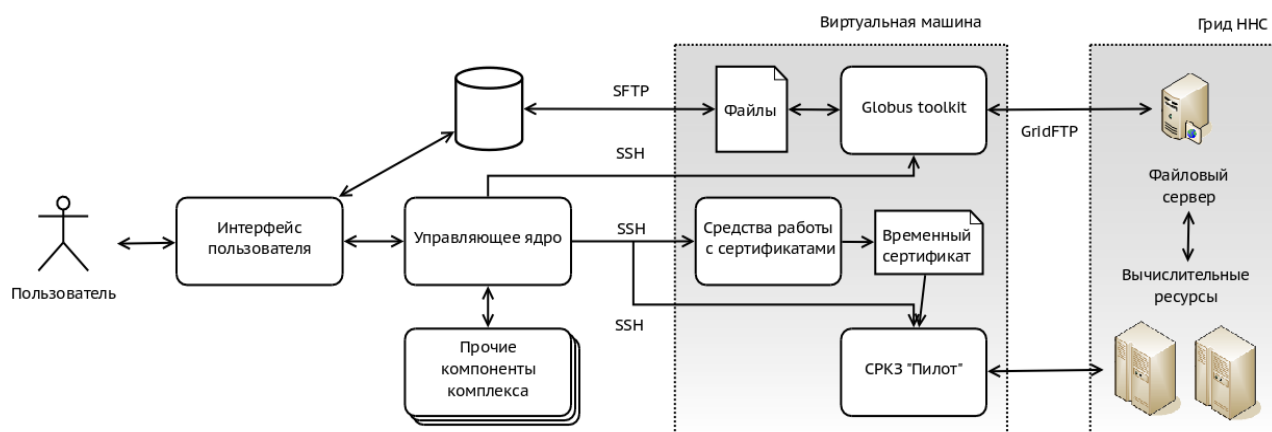


Рисунок 3.6 – Схема организации взаимодействия МИТП-М с Грид-средой<sup>2</sup>

Таким образом, рассмотренные в разделах 3.2.1–3.2.4 методы и технологии обеспечивают решение основной задачи МИТП-М – поддержку разработки и исполнения композитных приложений на основе ресурсов коллаборативных распределенных сред – набора программно-аппаратных комплексов, принадлежащих разным владельцам, связанных через сети общего назначения (Интернет) на основе единых стандартов взаимодействия (например, Грид) и предназначенных для решения задач, определяемых потребностями пользователей.

### **3.2.5. Особенности создания и управления межорганизационными (коллаборативными) композитными программными приложениями**

Отличительными особенностями создания и управления межорганизационными композитными программными приложениями являются.

- 1) Реализация WF в большей мере отходит на сторону конечного пользователя – шаблоны не являются основными композитными приложениями.
- 2) Пользователи платформы самостоятельно разрабатывают сложные индивидуальные WF, в связи с различной спецификой решаемых задач в Грид.
- 3) Доступность прикладных пакетов через Грид позволяет разрабатывать WF с использованием уникальных сервисов, размещенных на специфических ресурсах.
- 4) WF зачастую работают с разнородными данными, что затрудняет процесс создания взаимодействия пакетов на уровне описания композитного приложения. В этой связи, возникает необходимость в реализации дополнительных механизмов

<sup>2</sup> На примере Грид Национальной нанотехнологической сети.

RU.СНАБ.80066-06 31 03 **Ошибка! Источник ссылки не найден.**

конвертации (т.е. extractor-ов и assembler-ов), или более того, в создании новых вспомогательных пакетов конвертирования и обработки однотипных данных.

- 5) При редактировании пользовательского WF существует возможность управлять выполнением конкретных шагов, определяя группы ресурсов выполнения для каждого шага WF или только для некоторых из них, а также варьировать их приоритеты (если это не противоречит политике Грид-среды)

Пользователями МИТП-М могут быть только члены уже существующих Грид-сетей, на основе которых развернута платформа. При авторизации в платформе пользователь предоставляет персональный Грид-сертификат, который передается затем системе запуска заданий в Грид. В том случае, если пользователь не обладает сертификатом, или сертификат невалиден (например, просрочен, или превышена квота), пользователь может только разрабатывать композитные приложения в МИТП-К, однако в их исполнении на ресурсах МИТП ему будет отказано.

### ***3.2.6. Решение типовой задачи компьютерного моделирования сложных систем с помощью коллаборативного приложения с использованием МИТП-М***

В качестве характерного композитного приложения для МИТП-М рассмотрим WF, реализующий задачу оценки эффективности различных стратегий вакцинации для противодействия развитию вирусной эпидемии. Задача решается на основе аппарата комплексных сетей: вначале по заданным вероятностным характеристикам моделируется контактная эпидемиологическая сеть заданного объема, а затем на ее основе проводится моделирование распространения инфекции при различных вариантах цензурирования сети. Результатом расчетов являются интервальные характеристики (минимальные и максимальные порядковые статистики, медиана, размах выборки) при заданной стратегии вакцинации. На основании этих характеристик возможно исследовать эффективность различных стратегий вакцинации в целях обеспечения поддержки принятия решений при планировании соответствующих санитарно-эпидемиологических мероприятий.

Решение задачи заключается в моделировании структуры контактной сети, реализации определенной стратегии по противодействию распространению вирусной инфекции и моделировании вирусного процесса. Моделирование структуры сети осуществляется при помощи расширенной конфигурационной модели. Она обеспечивает возможность создавать сеть, состоящую из нескольких классов вершин, различающихся по типу распределения степеней. Стратегия по противодействию эпидемии (вакцинация)

RU.СНАБ.80066-06 31 03 **Ошибка! Источник ссылки не найден.**

сводится к извлечению из сети определенного числа случайных узлов или узлов с наибольшей степенью. Для моделирования процесса распространения вируса используется модель Далея-Кендалла. Все множество вершин условно разделяется на три группы: незараженные, зараженные-распространители и невосприимчивые. На каждом шаге алгоритма моделирования производится расчет числа новообразованных зараженных и невосприимчивых к инфекции узлов. Если зараженный узел контактирует с незараженным, то с вероятностью  $\lambda$  второй узел также становится зараженным. Если зараженный узел контактирует с зараженным или невосприимчивым, то он с вероятностью  $\alpha$  становится невосприимчивым.

Все пакеты, используемые в данном WF, размещены в среде ГридННС.

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

### Листинг 3.1 – Код композитного приложения оценки эффективности стратегий

#### вакцинации

```
require inNetwork;
require exclConfFile;
require spreadConfFile;
require lrConfigFile;

//Generate network
step NetworkGeneration runs cnm (
    in_format = "short",
    inDataFile = inNetwork
);

//Exclude several nodes
step NodesExcluder runs excluder after NetworkGeneration (
    inDataFile = sweep [NetworkGeneration.Result.outs["output.dat"],
        NetworkGeneration.Result.outs["output.dat"],
        NetworkGeneration.Result.outs["output.dat"],
        NetworkGeneration.Result.outs["output.dat"]],
    inConfigFile = exclConfFile
);

//Information spread Modelling
step InfSpreadModelling runs ism after NodesExcluder (
    inDataFile = sweep NodesExcluder.Result.sweep_outs["output.dat"],
    inConfigFile = spreadConfFile
);

//Calculate min-max-aver Statistics
step SpreadStatistics runs spreadstats after InfSpreadModelling (
    spreadGroup = InfSpreadModelling.Result.sweep_outs["output.dat"]
);
```

Видно, что композитным приложением последовательно выполняется четыре шага – генерация сети (Network Generation), удаление узлов (NodesExcluder), расчет

RU.СНАБ.80066-06 31 03 **Ошибка! Источник ссылки не найден.**

распространения инфекции (SpreadModeling) и расчет интервальных оценок (SpreadStatistics).

На рис. 3.7 приведен результат запуска композитного приложения на исполнение в среде Грид. Видны два слоя, на которых выполнялось распараллеливание. На верхнем слое параллельно выполняются задачи расчета распространения инфекции по всей сети, а также задачи выполнения вакцинации (удаления узлов и ребер в сети). На нижнем слое идет распараллеливание по статистическому ансамблю, соответствующему различным сценариям распространения инфекции (пакет ism).

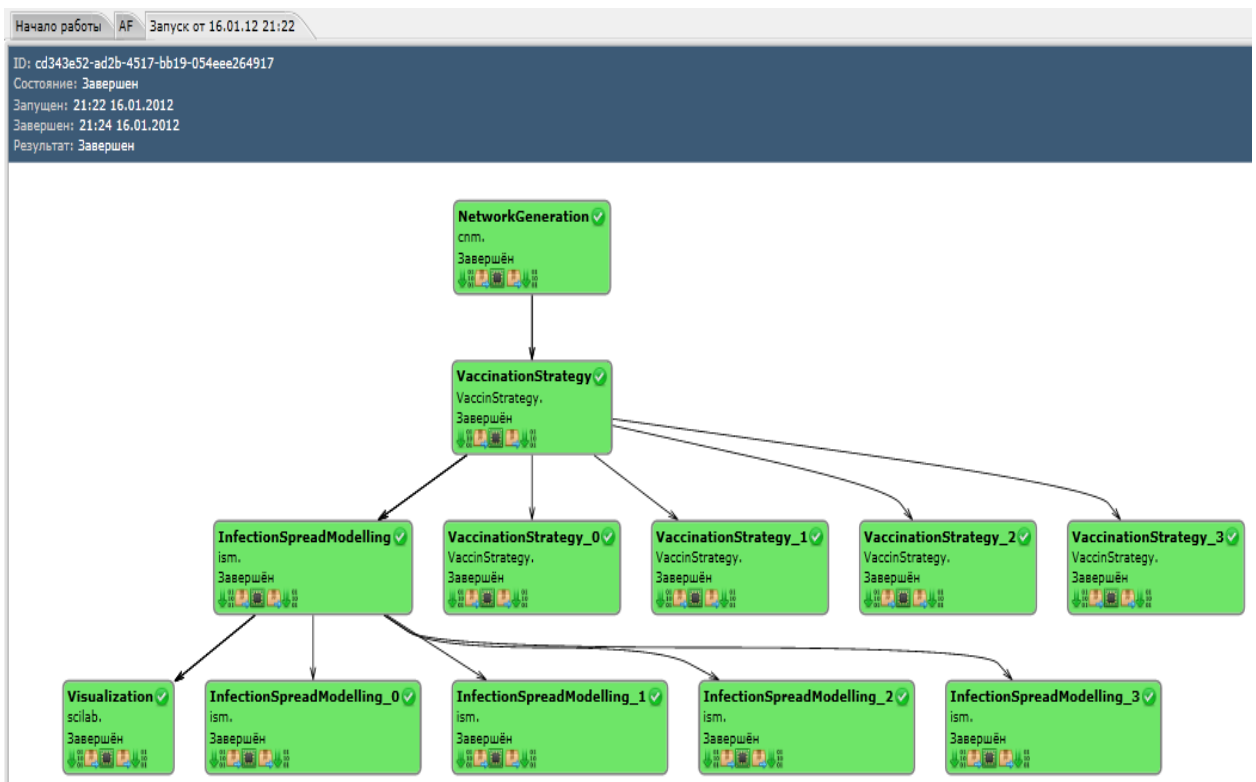
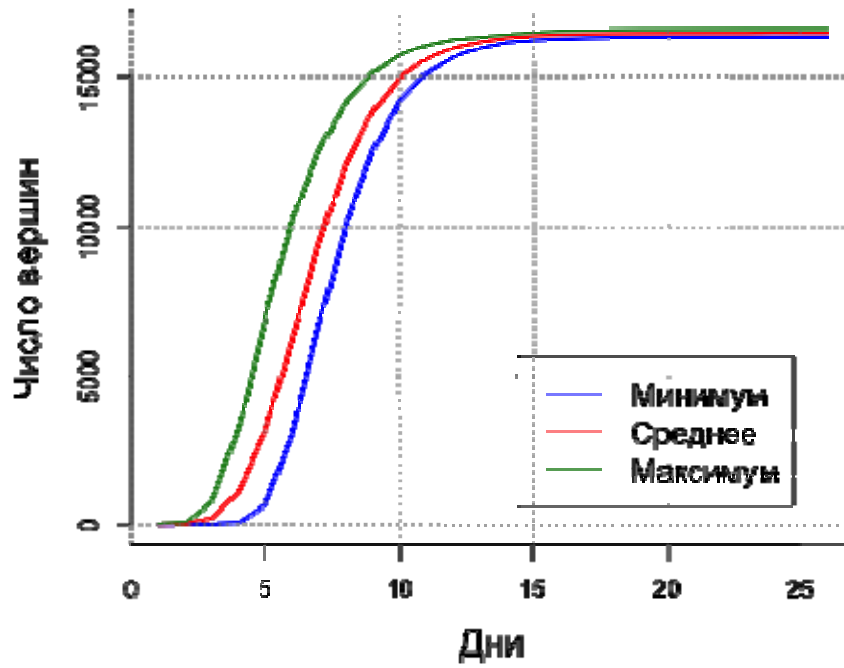
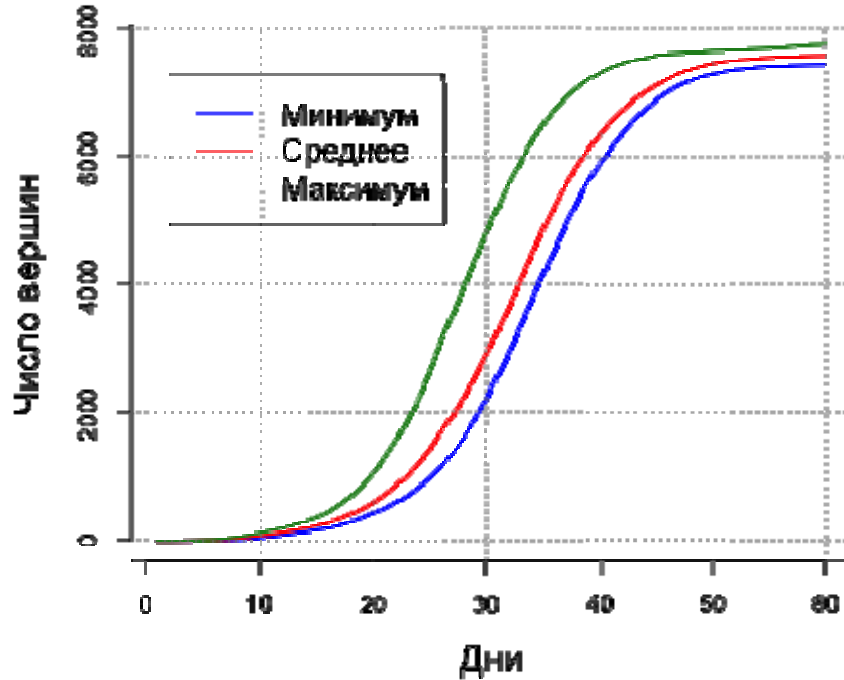


Рисунок 3.7 – Результат исполнения WF оценки эффективности стратегий вакцинации в МИТП-М

На рис. 3.8 результаты визуализации интервальных оценок различных стратегий вакцинации иллюстрируют расчеты МИТП-М.



(а)



(б)

Рисунок 3.8 – Количество инфицированных узлов в сети  
(комментарии по тексту)

RU.СНАБ.80066-06 31 03 **Ошибка! Источник ссылки не найден.**

Для эксперимента были выбраны следующие данные: количество участников – 50 000; средняя степень вершины – 30; первая стратегия (рис. 3.8(а)) заключается в вакцинации 5 % случайным образом выбранных участников; вторая стратегия (рис. 3.8(б)) заключается в вакцинации 5 % участников, имеющих наибольшее число связей с другими участниками. Видно, что при использовании первой стратегии вакцинации эпидемия прошла быстрее (примерно 15 дней при использовании первой стратегии и 60 дней – при использовании второй), но число заболевших гораздо больше, чем при использовании второй стратегии (максимальное число заболевших в первом случае около 17 000, а во втором – всего около 8000).

Подробное описание рассмотренной задачи и композитного приложения приведено в документе RU.СНАБ.80066-06 13 51.

#### **4. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ**

Как видно из примера в разделе 3.2.5, для работы с МИТП-М 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. 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-М 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-М	Технологическая платформа создания межорганизационных (коллоборативных) композитных программных приложений (на основе CLAVIRE)
ПАК	Программно-аппаратный комплекс
ОС	Операционная система
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
GaaS	Grid as a Service, модель облачных вычислений
MWF	Мета-WF
SaaS	Software as a Service, модель облачных вычислений
WF	Поток заданий, workflow

