МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

СОГЛАСОВАНО
Генеральный директор
ЗАО «АйТи»

_____ Бакиев О.Р.
"28" декабря 2011 г.

УТВЕРЖДАЮ
Ректор НИУ ИТМО

_____ Васильев В.Н.
"28" декабря 2011 г.

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

ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА ПОТОКОВОЙ ОБРАБОТКИ СВЕРХБОЛЬШИХ ОБЪЕМОВ ДАННЫХ И ИЗВЛЕЧЕНИЯ ИЗ НИХ ЗНАНИЙ НА ОСНОВЕ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ (МИТП-Д)

Руководство программиста

ЛИСТ УТВЕРЖДЕНИЯ

RU.СНАБ.80066-06 33 06-ЛУ

Представители
Организации-разработчика

Руководитель разработки,
профессор НИУ ИТМО

_____ Бухановский А.В.
"28" декабря 2011 г.

Ответственный исполнитель,
с.н.с. НИУ ИТМО

_____ Луценко А.Е.
"28" декабря 2011 г.

Нормоконтролер
ведущий инженер НИУ ИТМО

_____ Позднякова Л.Г.
"29" декабря 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 33 06-ЛУ

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

ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА ПОТОКОВОЙ ОБРАБОТКИ СВЕРХБОЛЬШИХ ОБЪЕМОВ ДАННЫХ И ИЗВЛЕЧЕНИЯ ИЗ НИХ ЗНАНИЙ НА ОСНОВЕ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ (МИТП-Д)

Руководство программиста

RU.СНАБ.80066-06 33 06

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

Листов 37

2011

АННОТАЦИЯ

Документ содержит руководство программиста технологической платформы потоковой обработки сверхбольших объемов данных и извлечения из них знаний на основе облачных вычислений (МИТП-Д) RU.СНАБ.80066-06 01 45. Технологическая платформа МИТП-Д входит в состав многопрофильной инструментально-технологической среды (МИТП) CLAVIRE (Cloud Applications Virtual Environment) RU.СНАБ.80066-06. Она предназначена для поддержки разработки и исполнения композитных приложений сбора и обработки данных из распределенных источников, объединенных сетями общего назначения (Интернет) на основе единых стандартов взаимодействия, в целях решения специфических задач (например, анализа медиаконтента в социальных сетях). МИТП-Д 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 И УСЛОВИЯ ПРИМЕНЕНИЯ ПРОГРАММЫ

1.1. Назначение программы

Технологическая платформа потоковой обработки сверхбольших объемов данных и извлечения из них знаний на основе облачных вычислений (МИТП-Д) RU.СНАБ.80066-06 01 45 входит в состав многопрофильной инструментально-технологической среды (МИТП) CLAVIRE (Cloud Applications Virtual Environment) RU.СНАБ.80066-06. Она предназначена для поддержки разработки и исполнения композитных приложений сбора и обработки данных из распределенных источников, объединенных сетями общего назначения (Интернет) на основе единых стандартов взаимодействия, в целях решения специфических задач (например, анализа медиаконтента в социальных сетях).

МИТП-Д представляет собой комплекс программного обеспечения для разработки, настройки и эксплуатации сред распределенных вычислений обработки больших объемов данных, предназначенный для:

1) эффективного управления вычислительными, информационными и программными ресурсами среды распределенного сбора и обработки данных, 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 сбора и обработки данных в распределенных источниках.
3) обеспечения функционирования программно-аппаратного комплекса (ПАК) поддержки инфраструктуры облачных вычислений для сбора и обработки больших объемов данных.

В соответствии с ТТ к основным функциям технологической платформы МИТП-Д, обеспечивающим решение поставленных задач, относятся:

1. Поддержка разработки и исполнения композитных приложений сбора и обработки данных из распределенных источников, объединенных сетями общего назначения (Интернет) на основе единых стандартов взаимодействия, в целях решения специфических задач.
2. Возможность развертывания в существующих коллаборативных распределенных средах, включая существующие инфраструктуры Грид I поколения.

3. Динамическое управление (мониторинг состояния, запуск приложений, передача данных, распределение нагрузки, миграция данных и задач) в автоматическом режиме набором приложений для сбора и обработки данных из распределенных источников.
4. Автоматическая оптимизация по времени процесса использования доступных вычислительных ресурсов и прикладных сервисов для сбора и обработки данных из распределенных источников.
5. Представление описания композитных приложений для сбора и обработки данных из распределенных источников на основе цепочек заданий (workflow), обеспечивающих запуск, выполнение, остановку и возобновление работы цепочки заданий в ручном и автоматическом режимах.
6. Поддержка процесса установки и первоначальной конфигурации технологической платформы для сбора и обработки данных из распределенных источников, и ее составных частей на ресурсах коллаборативной распределенной среды.
7. Поддержка многопользовательского режима при решении задач сбора и обработки данных из распределенных источников.
8. Квотирование, биллинг и тарификация использования данных из распределенных источников и вычислительных сервисов их обработки.
9. Каталогизация входных данных пользователей на основе метаданных.
10. Администрирование и контроль работы с дифференцированными правами администраторов в рамках многоуровневой политики доступа к ресурсам распределенного хранения данных.
11. Модификация знаний, используемых системой, как в ручном, так и в автоматическом режимах.
12. Функционирование сервисов резервирования и отката исправлений для результатов работы вычислительных сервисов обработки данных в удаленном хранилище в составе распределенной среды хранения данных.
13. Функционирование механизмов конвертирования данных между различными прикладными сервисами обработки данных, по заданию пользователя.

В данном документе рассматриваются способы обращения к МИТП-Д с целью описания композитных приложений, построенных на основе прикладных сервисов (пакетов), доступных МИТП в ходе ее функционирования. Для описания композитных приложений применяется предметно-ориентированный язык EasyPackage.

1.2. Условия применения программы

Для развертывания компонентов МИТП-Д 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-Д должен быть установлен сервер баз данных MongoDB версии 1.6.5. В ходе установки и настройки используются стандартные конфигурации указанных программных средств, не требующие специфической модификации. После установки необходимо осуществить запуск сервера баз данных для локального использования (localhost).

Для работы компонента информационного портала RU.СНАБ.80066-06 01 31 требуется установка СУБД MySQL (версии 5.0 или выше) и поддержка web-сервером интерпретатора языка PHP (версии 5.2 или выше). Для работы компонента хранения знаний RU.СНАБ.80066-06 01 17 требуется установка СУБД Microsoft SQLServer Compact Edition (версии 3.5 или выше). Также должен быть установлен web-сервер Glassfish версии 3.0.1, обеспечивающий поддержку технологии WebServices, необходимой для функционирования варианта реализации хранилища онтологической структуры RunLib. Кроме того, на ту же вычислительную систему должен быть установлен интерпретатор онтологической структуры Pellet версии 2.2.2, необходимый для функционирования хранилища знаний.

Для работы компонента сбора данных в социальных сетях в Интернете RU.СНАБ.80066-06 01 39 необходимы развернутые библиотеки с открытым исходным кодом Apache Hadoop (версии 0.20.2 и выше).

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

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

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

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

В целях увеличения производительности и реактивности МИТП-Д отдельные компоненты могут функционировать на разных вычислительных системах в рамках общей локальной сети центра обработки данных.

2. ХАРАКТЕРИСТИКИ ПРОГРАММЫ

2.1. Технологические функции программы

МИТП-Д призвана обеспечить выполнение следующих технологических функций:

1) использование ресурсов среды распределенного сбора и обработки данных, включая собственные (выделенные) вычислительные ресурсы центров и ресурсы внешних провайдеров (в том числе глобальных коллаборативных сред);
2) использование доступных прикладных пакетов сбора и обработки данных, установленных на ресурсах распределенной среды, для решения задач избранной предметной области;
3) формирование композитных приложений, использующих в своем составе вызовы предметно-ориентированных сервисов, реализованных с использованием доступных системе прикладных пакетов;
4) подготовка файлов входных данных и расчетных параметров, необходимых для выполнения композитных приложений;
5) автоматизированное определение целевых ресурсов, доступных для исполнения сервисов в составе композитного приложения;
6) запуск процесса выполнения сформированных композитных приложений в распределенной вычислительной среде;
7) мониторинг процесса исполнения пользовательского задания в распределенной вычислительной среде;
8) работа с пользовательскими данными в удаленном хранилище.

2.2. Порядок использования программы

Порядок использования МИТП CLAVIRE следующий.

1. Пользователь авторизуется через графический интерфейс платформы (представленный отдельным web-приложением или информационным web-порталом), что дает ему возможность доступа к соответствующим сервисам МИТП.
2. Пользователь через соответствующий web-интерфейс формирует композитное приложение на предметно-ориентированом языке EasyFlow в форме описания абстрактного потока заданий (workflow, WF).
3. 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.
4. Для подготовленного описания композитного приложения пользователь конфигурирует условия вычислений: определяет требуемые параметры WF, редактирует (при необходимости) его описание, готовит и загружает в хранилище входные данные для расчетов.
5. Пользователь запускает задачу на выполнение. При этом происходит автоматическое определение наилучшего набора сервисов, соответствующего вызовам в данном абстрактном WF (планирование). При необходимости процесс планирования может повторяться несколько раз в процессе выполнения композитного приложения.
6. Пользователь (при необходимости) осуществляет мониторинг процесса исполнения (в форме динамического отображения конкретного WF). Пользователь может на время расчетов завершить рабочую сессию с МИТП и начать ее только при необходимости использования результатов (необязательно в тот момент, когда расчеты завершены).
7. Когда расчет задачи завершен, результаты перемещаются из распределенной среды в соответствующее хранилище данных МИТП. Пользователь может получить доступ к результатам расчетов (в форме выдачи результатов соответствующих прикладных пакетов) через web-интерфейс МИТП.

Более детально порядок применения МИТП CLAVIRE рассмотрен в документе «Описание применения» RU.СНАБ.80066-06 31 01.

2.3. Режимы взаимодействия пользователя с программой

Для взаимодействия программиста (разработчика композитных приложений) с платформой МИТП предусмотрены три основных режима.

- Режим использования шаблонов, соответствующих типовому применению вызовов отдельных прикладных пакетов. В этом случае пользователь выбирает только пакет (или вид пакета по предметному описанию) и отвечает за подготовку входных данных. Выбор условий выполнения, настройка виртуального вычислительного ресурса и запуск пакета выполняются автоматически.
- Режим ручного создания композитных приложений подразумевает описание композитного приложения в окне редактирования на языке EasyFlow с непосредственным определением режимов вызова прикладных пакетов, необходимых пользователю.
- Режим использования сценариев позволяет применять заранее подготовленные описания композитных приложений с ассоциированными входными данными в форме проектов МИТП. При этом пользователь по необходимости может изменять только отдельные значения параметров запуска.

Для практической реализации перечисленных выше режимов используется предметно-ориентированное описание композитных приложений на языке EasyFlow.

2.4. Общая архитектура МИТП-Д

Платформа МИТП-Д (рис. 2.1) предназначена для решения задач обработки больших объемов данных, в связи с этим важную роль в ней играет подсистема работы с данными. Компонент хранения данных в такой конфигурации предоставляет расширенную функциональность, куда входят возможность использования распределенных хранилищ данных, оптимизация перемещения данных, оптимизация хранения данных большого объема. Перечисленные особенности необходимы для обеспечения эффективной работы платформы при исполнении WF, представляющих собой сети обработки данных.

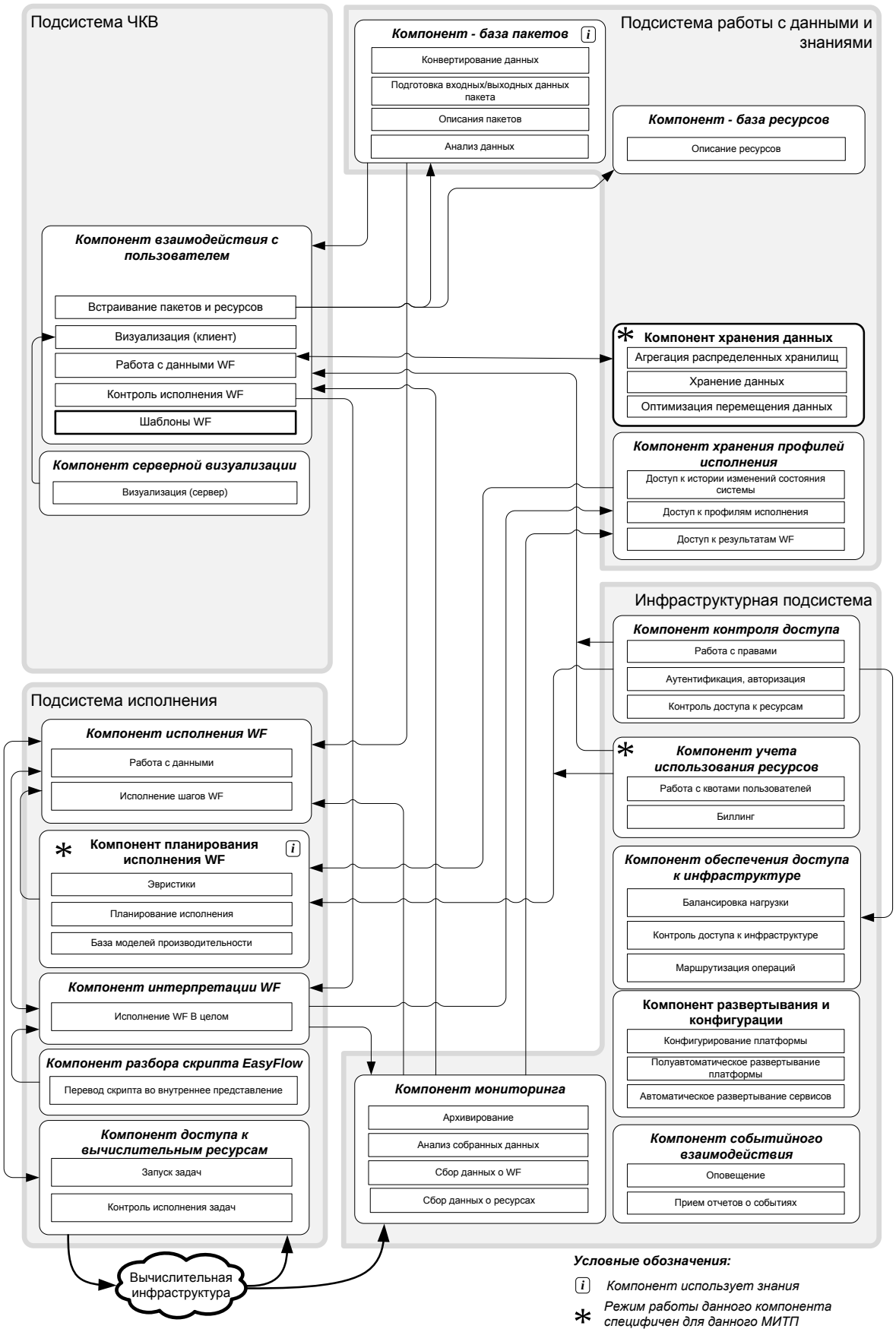

Рисунок 2.1 – Структура МИТП-Д

Процесс планирования в МИТП-Д учитывает перемещения данных, их объем, скорость передачи между ресурсами. Это позволяет производить оптимизацию выполнения WF по обработке данных, обладающих небольшой вычислительной сложностью. При планировании учитывается возможность «переноса» вычислений к данным, выраженная в том, что для запуска сервиса с большей вероятностью будет выбран ресурс, на котором необходимые данные уже присутствуют, по сравнению с ресурсом, на который данные необходимо копировать.

Особенностью МИТП-Д является то, что при учете использования ресурсов важен объем хранимых и переданных по сети данных.

Необходимо отметить, что в МИТП-Д отсутствует компонент информационного портала. Отсутствуют компоненты диалога поддержки принятия решений и хранения знаний, так как построение WF сведено к выбору из набора типовых шаблонов.

2.5. Основные компоненты МИТП-Д

В состав МИТП-Д входят все компоненты ядра МИТП (см. документ «Руководство системного программиста» МИТП RU.СНАБ.80066-06 32 01). Кроме того, в состав МИТП-Д включены следующие компоненты (с указанием зависимостей между ними и компонентами ядра МИТП):

- RU.СНАБ.80066-06 01 22. Компонент серверной визуализации CLAVIRE/CSNV. ***Зависимости:*** CLAVIRE/Storage.
- RU.СНАБ.80066-06 01 32. Компонент профилей исполнения CLAVIRE/Provenance. ***Зависимости:*** CLAVIRE/Eventing, CLAVIRE/Monitoring.
- RU.СНАБ.80066-06 01 34. Компонент учета использования ресурсов CLAVIRE/Billing. ***Зависимости:*** CLAVIRE/Eventing, CLAVIRE/Monitoring.
- RU.СНАБ.80066-06 01 27. Компонент обеспечения доступа к инфраструктуре CLAVIRE/InfraAccess. ***Зависимости:*** нет.
- RU.СНАБ.80066-06 01 19. Компонент разбора скрипта EasyFlow CLAVIRE/EasyFlow. ***Зависимости:*** нет.

Кроме того, в состав МИТП-Д включен ряд дополнительных компонентов:

- RU.СНАБ.80066-06 01 60. Библиотека для встраивания пакетов длительного исполнения CLAVIRE/LRWFLib, ориентированная на поддержку прикладных сервисов (пакетов) длительного исполнения.

- RU.СНАБ.80066-06 01 62. Компонент поддержки пользователя при разработке скриптов на EasyFlow CLAVIRE/UI/EasyFlowEditorServices, обеспечивающий поддержку пользователя в процессе ввода скриптов на языке EasyFlow.
- RU.СНАБ.80066-06 13 64. Компонент средств мониторинга и настройки основных компонентов платформы CLAVIRE/AdminTool, предназначенный для администрирования и контроля работы платформы. ***Зависимости:*** CLAVIRE/ResourceBase, CLAVIRE/PackageBase, CLAVIRE/Billing, CLAVIRE/GateKeeper, CLAVIRE/Portal.

3. ОБРАЩЕНИЕ К ПРОГРАММЕ

В ходе обращения к МИТП CLAVIRE программист формирует описание композитного приложения в форме абстрактного WF на языке EasyFlow. При оперировании характеристиками пакетов используется описание прикладных пакетов на языке EasyPackage (см. «Руководство программиста» для МИТП RU.СНАБ.80066-06 33 02). В данном разделе приводится общая характеристика языка EasyFlow, а также описываются операции, имеющие особое значение для технологической платформы МИТП-Д.

Взаимодействие программиста с МИТП CLAVIRE осуществляется посредством интерактивной работы с интерфейсом компонента CLAVIRE/Ginger, обеспечивающего возможность построения композитного приложения, передачи пользовательских входных данных, запуска построенного приложения и получения результата. Обращение к МИТП, основные операции по работе с интерфейсом CLAVIRE/Ginger, способы запуска композитных приложений (передача управления МИТП) описаны в документе «Руководство оператора» для МИТП-Д RU.СНАБ.80066-06 34 06.

3.1. Алгоритм построения композитного приложения

Приоритетный подход к построению композитного приложения в МИТП-Д – построение на основе шаблона. Схема соответствующего алгоритма приведена на рис. 3.1. Рассмотрев этот алгоритм, следует отметить ряд особенностей реализации.

1) Редактирование WF для систем, построенных на базе МИТП-Д, заключается в настройке параметров и задании входных данных для композитного приложения. Ввиду специфики приложений, ориентированных на распределенную обработку данных, типичными являются параметры поиска данных в хранилище, доступном

сервису. С другой стороны, настройка данных может заключаться в конфигурации входных потоков сервисов.

2) Система позволяет планировать исполнение по двум критериям: стоимости и времени. Однако более целесообразна оценка по стоимости, определяемой на основе входных параметров задачи. Стоимость распределенных сервисов анализа данных в типовом сценарии работы с приложением, построенном на базе МИТП-Д, определяется исходя из объема обработанных данных. Тем не менее в ряде случаев оценка объема обрабатываемых данных представляется нетривиальной задачей. В таких ситуациях оценка стоимости может проводиться на основании правил и тарифов, сформированных владельцами данных.
3) В процессе использования хранилища данных в составе МИТП-Д значительную роль играет возможность работы с потоками данных. При этом наиболее важны выходные потоки удаленных сервисов обработки данных.

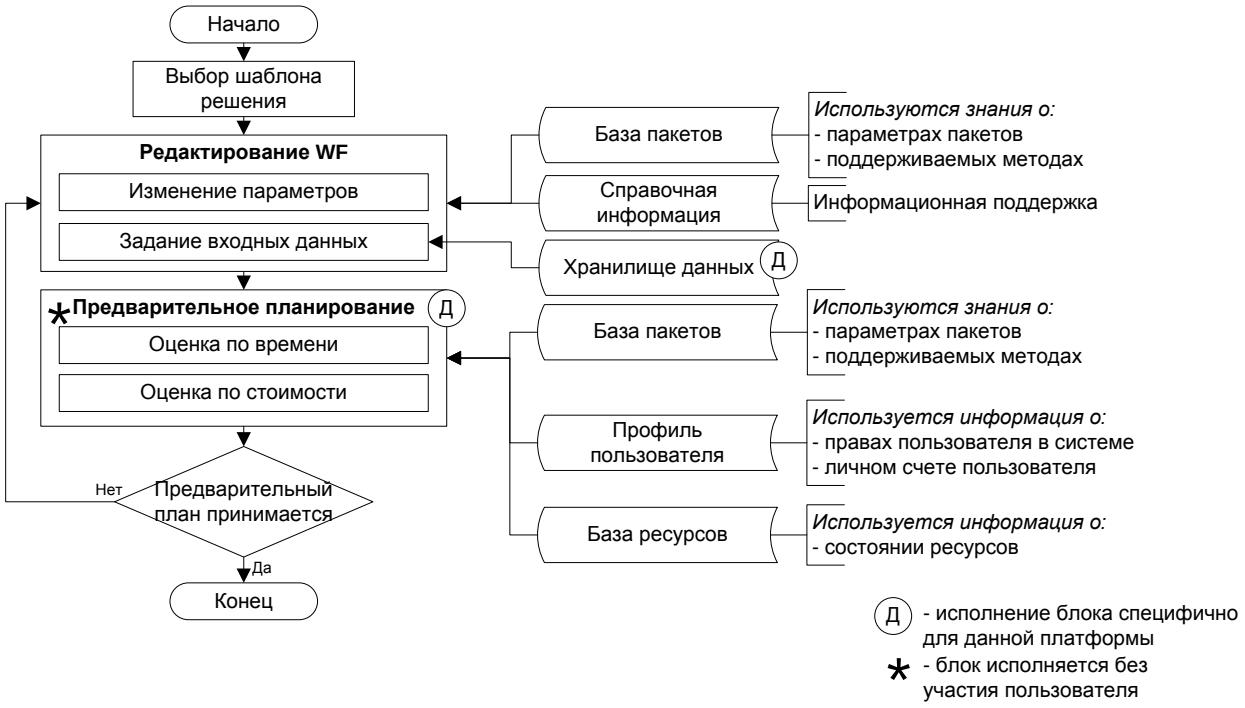

Рисунок 3.1 – Алгоритм создания типового композитного приложения в МИТП-Д

Построенное композитное приложение может быть исполнено с применением алгоритма интерпретации и исполнения WF в составе МИТП-Д (рис. 3.2). При этом существует ряд особенностей, характерных, в первую очередь, для реализации данного процесса в рамках МИТП-Д.

1) Композитное приложение должно исполняться на подготовленной сервисной инфраструктуре из распределенных сервисов обработки данных. Такая инфраструктура формируется перед запуском основного цикла вычислений.
2) В цикле контроля исполнения осуществляется обработка обновлений состояния сервисной инфраструктуры. При этом, например, в случае изменения состава и параметров инфраструктуры может быть произведена повторная планировка исполнения композитного приложения. В этом случае выполнение сервисов приостанавливается и после повторного планирования сервисная инфраструктура обновляется для данного приложения.

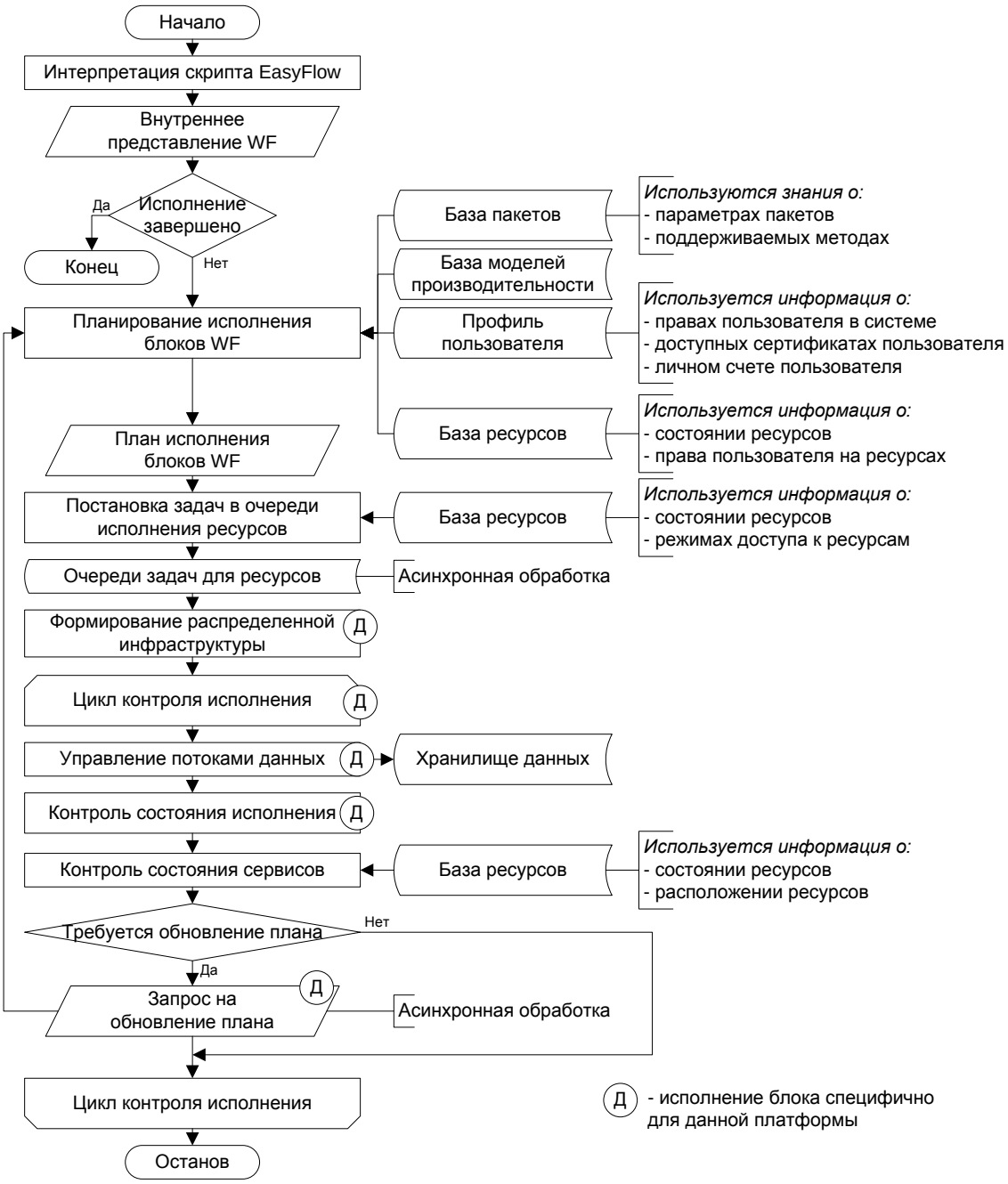

Рисунок 3.2 – Алгоритм исполнения композитного приложения в МИТП-Д

3) В ходе выполнения указанного цикла контролируется состояние сервисов в составе построенной сервисной инфраструктуры (активность сервисов) и состояние исполнения (степень завершенности задания).
4) Сервисная инфраструктура осуществляет управление потоками данных, поступающих со стороны распределенных сервисов сбора и обработки данных.

3.2. Описание композитных приложений на языке EasyFlow

Язык описания композитных приложений в форме WF EasyFlow (далее – Язык или EasyFlow) предназначен для описания инструкций по выполнению композитных приложений в распределенной вычислительной среде. Он предоставляет конечному пользователю гибкие возможности по заданию различных форм потоков вычислений (WF), в рамках которых происходит выполнение различных прикладных пакетов, генерация выходных данных, их получение, конвертация и обработка.

Основной целью Языка является полное абстрагирование от особенностей распределенной вычислительной среды, в которой работает пользователь. Это делает описание задания полностью независимым от той вычислительной среды, которая доступна на данный момент. Иными словами, EasyFlow – это высокоуровневый язык описания *абстрактных* workflow (AWF), такой подход позволяет описывать саму решаемую задачу, а не способ ее исполнения на конкретной вычислительной архитектуре.

3.2.1. Основные элементы языка

В данном разделе описаны элементы языка EasyFlow с использованием нотации ANTLR.

Язык EasyFlow имеет следующие основные характеристики:

- скрипт представляет собой текстовый файл (с расширением .flow) в кодировке UTF-8;
- EasyFlow является языком, зависимым от регистра (т. е. myVariable и MyVariable представляют собой различные идентификаторы);
- типизация переменных – динамическая, слабая, с ограниченными возможностями по указанию типов;
- EasyFlow является платформо-независимым (с точки зрения различия операционных систем и вычислительных платформ);
- EasyFlow является интерпретируемым.

Переводы строк и пробелы. В EasyFlow переводы строк и пробелы являются разделителями различных конструкций языка. Для определения работы с переводами строк и пробелами в грамматику языка внесены правила, приведенные в листинге 3.1.

Листинг 3.1. Правила обработки пробельных символов

```
wsn
  : WS
```

```
  | NEWLINE
  ;

wsnp
  : wsn+
  ;

wsns
  : wsn*
  ;

WS
  : ' ' | '\t' ;

NEWLINE
  : '\r'? '\n'
  ;
```

Пробельными символами считаются: перевод строки, пробел и знак табуляции. Переводы строк могут быть следующими: «\r\n» (Windows-style) и «\n» (Unix-style).

Для удобства определены правила wsns (любое количество пробельных символов) и wsnp (один и более пробельных символов), которые активно используются во всей грамматике языка.

Комментарии. В тексте скрипта могут встречаться однострочные комментарии после символов «//» и до конца строки, которые игнорируются парсером. Многострочные комментарии заключаются внутри символов «/*» и «*/».

Служебные символы: «{», «}», «[», «]», «(», «)», «=», «;» (точка с запятой), «:» (двоеточие), «.» (точка), «,» (запятая), «~» (тильда), «<-» (стрелка влево). Определение терминалов для этих символов приведено в листинге 3.2.

Листинг 3.2. Определение терминалов для служебных символов

```
LCUR
    : '{'
    ;

RCUR
    : '}'
    ;

LSQ
    : '['
    ;
RSQ
    : ']'
    ;

LPAREN
    : '('
    ;
```

```
RPAREN
    : ')'
    ;

ASSIGN
    : '='
    ;

SEMICOLON
    : ';'
    ;

COLON
    : ':'
    ;
DOT
    : '.'
    ;

COMMA
    : ','
    ;

TILDA
    : '~'
    ;

ARROW
    : '<-'
    ;
```

Идентификаторы языка EasyFlow определяются правилами, приведенными в листинге 3.3. Они могут начинаться на букву английского алфавита или знак подчеркивания и содержать любое количество букв английского алфавита, цифр и знаков подчеркивания.

Листинг 3.3. Определение идентификаторов

```
ID
    : ID_BASE
    ;

fragment ID_BASE
    : ('a'..'z'|'A'..'Z'|'_' ) ('a'..'z'|'A'..'Z'|'0'..'9'|'_')*
    ;
```

Идентификаторы могут быть простыми (ID) и для указания констант (ID_CONST), которые префиксируются символом «@».

Строки в EasyFlow – это последовательности символов, заключенные в двойные кавычки вида «"». Внутри кавычек могут располагаться как обычные символы, так и

escape-последовательности символов для ввода специальных символов, список которых приведен ниже:

«\"» – символ «"» (двойные кавычки);

«\b» – символ backspace;

«\t» – символ табуляции;

«\n» – символ перевода строки;

«\f» – символ разрыва раздела;

«\r» – символ возврата каретки;

«\\»– символ «\»;

«\'» – символ «'» (одинарные кавычки).

Строки могут содержать коды символов в шестнадцатеричном (вида «\uXXXX», где X – шестнадцатеричная цифра) и восьмеричном (вида «\XXX», где X – восьмеричная цифра) представлениях.

Листинг 3.4. Определение терминалов для строк

```
STRING
    :  '"' ( (~('\\' | '"'))| ESC_SEQ)* '"'
    ;

fragment HEX_DIGIT : ('0'..'9'|'a'..'f'|'A'..'F') ;

fragment ESC_SEQ
    :   '\\' ('b'|'t'|'n'|'f'|'r'|'\"'|'\''|'\\')
    |   UNICODE_ESC
    |   OCTAL_ESC
    ;

fragment OCTAL_ESC
    :   '\\' ('0'..'3') ('0'..'7') ('0'..'7')
    |   '\\' ('0'..'7') ('0'..'7')
    |   '\\' ('0'..'7')
    ;

fragment UNICODE_ESC
  : '\\' 'u' HEX_DIGIT HEX_DIGIT HEX_DIGIT HEX_DIGIT
  ;

fragment RULETTER
  : '\u0400'..'\u04FF'
```

Целые числа в EasyFlow представляются стандартным образом в десятичной системе счисления и могут иметь знак. Определение целых чисел приведено в листинге 3.5.

Листинг 3.5. Определение терминалов для целых чисел

```
INT : ('+' | '-')? DEC_DIGIT+
    ;

fragment DEC_DIGIT : ('0'..'9') ;
```

Дробные числа в EasyFlow представляются стандартным образом в десятичной системе счисления и могут иметь знак, целую часть, мантиссу и показатель степени (примеры: 0.14, 0.15e+14, .34e10, 5.e–14). Определение дробных чисел приведено в листинге 3.6.

Листинг 3.6. Определение терминалов для дробных чисел

```
DOUBLE
    :   DEC_DIGIT+ '.' DEC_DIGIT* EXPONENT?
    |   '.' DEC_DIGIT+ EXPONENT?
    |   DEC_DIGIT+ EXPONENT
    ;

fragment DEC_DIGIT : ('0'..'9') ;

fragment EXPONENT : ('e'|'E') ('+'|'-')? ('0'..'9')+ ;
```

Булевы константы в EasyFlow представляют собой ключевые слова: true (истина) и false (ложь). Определение булевых констант приведено в листинге 3.7.

Листинг 3.7. Определение терминалов булевых констант

```
BOOL
    :  ('true') | ('false')
    ;
```

Выражения в EasyFlow в текущей версии являются простыми и могут представлять собой:

строку;

список (см. ниже);

целое число;

дробное число;

булеву константу;

константный идентификатор (ID_CONST, см. выше);

выражение доступа (см. ниже).

Определение правила для выражения приведено в листинге 3.8.

Листинг 3.8. Определение правила для выражения

```
expression
  : STRING
  | INT
  | DOUBLE
  | BOOL
  | list
  | ID_CONST
  | varIdentifier
  ;
```

Выражения доступа. В EasyFlow для описания доступа к различным сложным объектам (именам пакетов, структурам) используются выражения доступа (varIdentifier), которые представляют собой идентификаторы, разделенные точками и, возможно, имеющие индексаторы, заключенные в квадратные скобки (примеры: object.Field, object.Array[13], ORCA.List[“Hello”].str). Определение правил для выражений доступа приведено в листинге 3.9.

Листинг 3.9. Определение правил для выражения доступа

```
varIdentifier
  : simpleVarIdentifier( wsns DOT wsns simpleVarIdentifier )*
  ;

simpleVarIdentifier
  : ID (wsns varIndexer)?
  ;

varIndexer
  : LSQ wsns expression wsns RSQ
  ;
```

Списки в EasyFlow представляют собой последовательности выражений, разделенных запятыми и заключенных в квадратные скобки (примеры: [1, 2, 3], [a, [c, d]]). Списки могут быть вложенными и содержать значения различных типов. Определение правил для списков приведено в листинге 3.10.

Листинг 3.10. Определение правил для списков

```
list
  : LSQ wsns expressionList? RSQ
  ;

expressionList
  : expression wsns
  (COMMA wsns expression wsns)*
  ;
```

Ключевые слова. EasyFlow для служебных целей резервирует следующие ключевые слова:

flow – для префиксирования атрибутов WF (см. ниже);

require – для определения требуемых входных файлов;

step – описание начала шага;

~step — описание долгоживущего шага;

sweep – оператор перебора по значениям;

runs – дескриптор запускаемого пакета;

after – указатель явной передачи управления между шагами;

on – указатель передачи управления между шагами по событиям;

true / false – истина/ложь;

pre, code, post — определение секций пре- и постобработки.

Определение терминалов для ключевых слов приведено в листинге 3.11.

Листинг 3.11. Определение терминалов для ключевых слов

```
FLOW
    : 'flow'
    ;

REQUIRE
    : 'require'
    ;

STEP
    : 'step'
    ;

SWEEP
    : 'sweep'
    ;

RUNS
    : 'runs'
    ;

AFTER
    : 'after'
    ;

ON
    : 'on'
    ;

BOOL
    :  ('true') | ('false')
    ;

POST
  : 'post';
```

```
fragment CODE
  : 'code';

PRE
  : 'pre';

fragment END
  : 'end';
```

Предопределенные константы. В EasyFlow возможно использовать предопределенные средой константы. Для этого используются идентификаторы с префиксом в виде символа «@».

Листинг 3.12. Определение терминала для предопределенных констант

```
ID_CONST
    : '@' ID_BASE
    ;
```

3.2.2. Построение workflow

Строение тела скрипта. Тело скрипта (правило program) представляет собой набор выражений верхнего уровня (правило topClause) – см. листинг 3.13.

Листинг 3.13. Правила program и topClause

```
wsn
  : WS
  | NEWLINE
  ;

wsnp
  : wsn+
  ;

wsns
  : wsn*
  ;

program
  : wsns ( topClause wsns )*
  ;


topClause
  : flowAttributeClause
  | stepClause
  | requireClause
  ;
```

Как видно из листинга, выражением верхнего уровня могут быть: атрибут WF (flowAttributeClause), определение шага (stepClause) и определение требования файла (requireClause). Об этих элементах см. ниже.

Атрибуты WF представляют собой задание настройки для исполнения WF и имеют синтаксис, приведенный в листинге 3.14. Пример задания настройки для WF приведен в листинге 3.15.

Листинг 3.14. Определение правила для атрибутов WF

```
flowAttributeClause

  : LSQ wsns FLOW wsns COLON wsns ID wsns ASSIGN wsns expression RSQ
  ;
```

Листинг 3.15. Примеры атрибутов WF

```
[flow:priority = @urgent]
[flow:author = “Paul McCartney”]
[flow:name = “Molecular geometry optimization”]
[flow:mode = @raw]
```

Тело скрипта может содержать любое количество атрибутов WF, заданных в форме, приведенной выше. Эти атрибуты задают либо некую информацию о WF (например, данные об авторстве или названии WF), либо системные настройки, связанные с запуском.

Директива требования файлов (require). Чтобы указать, какие файлы требуются WF, в EasyFlow введена директива *require*. Она представляет собой ключевое слово «require» и следующий за ним список разделенных запятыми идентификаторов, обозначающих файлы. В дальнейшем эти идентификаторы будут ассоциированы с конкретными файлами, использующимися в запусках. В рамках одного скрипта директива требования файлов может появляться неограниченное число раз.

Определение правила для директивы *require* приведено в листинге 3.16. Примеры директив приведены в листинге 3.17.

Листинг 3.16. Определение директивы require

```
requireClause
  : REQUIRE wsnp idList wsns SEMICOLON
  ;
idList
  : ID (wsns COMMA wsns ID)*
```

```
  ;
```

Листинг 3.17. Примеры использования директивы require

```
require file1;
require jpg, png, gif;
```

Директива определения шага (step). Одной из основных функций EasyFlow является описание запусков вычислительных пакетов безотносительно к распределенной вычислительной архитектуре. Для этого используется директива определения шага *step*, задание правил для которой приведено в листинге 3.18.

Листинг 3.18. Определение правил для директивы step

```
stepClause
  :
    attributes
    (TILDA)? STEP wsnp id
    RUNS wsnp compoundName
    (wsnp AFTER wsnp idList)?  // Run conditions end
    wsns LPAREN
        wsns (APP wsns COLON wsns)? namedParametersList?
        ((EXEC wsns COLON wsns) namedParametersList?)?
    RPAREN
    (wsns 'pre' WS+ CODE_BLOCK)?
    (wsns 'post' WS+ CODE_BLOCK)?
    (wsns SEMICOLON)?
 ;

compoundName
  : ID ( wsns DOT wsns ID )*

attributes
  : (attribute wsns)*
  ;

attribute
  : LSQ wsns ID wsns ASSIGN wsns expression wsns RSQ
  ;

namedParametersList
  : namedParameter wsns
    (
    (
      ( COMMA wsns namedParameter wsns )+
      ( COMMA wsns )? // comma at the end
    )
    | (COMMA wsns)?
    )
  ;

namedParameter
  : COLON?ID wsns ASSIGN wsns (SWEEP wsns)? expression
  ;
```

Директива *step* задает параметры и связи с другими шагами для выполняемого вычислительного пакета. Полный формат описания шага включает: атрибуты шага, имя шага, запускаемый пакет, условия получения управления (директива *after*), список входных параметров и файлов и описание секции постобработки. Пример полного определения шага приведен в листинге 3.19.

Листинг 3.19. Пример полного определения выполнения шага

```
require file1, file2;

step AnotherStep runs EmptyPackage ();

[priority = @high]
step StepName runs Package.Method after AnotherStep
(
  inFile1 = file1,
  inFile2 = file2,
  stringInput = "some string here",
  intInput = 100,
  doubleInput = 3.14,
  sweepParam = sweep [1, 2, 3],
  listParam = [AnotherStep.outs["out.txt"]]
)
post code ruby
  i = 1
  list = StepName.Result.outs
  list.reverse
code end

~step LongRunningStep runs LRPackage
(
  inStream <- StepName.Result.outs[“output.txt”]
);
```

3.2.3. Способы структурирования программы

В предыдущем разделе был описан синтаксис языка EasyFlow. Данный раздел посвящен особенностям создания WF с его помощью, а также технике выстраивания выполнения шагов относительно друг друга. В следующих пунктах подробно разбираются различные возможности описательной стороны EasyFlow.

Номенклатура названий вычислительных пакетов. Для определения шага требуется указать пакет, который предполагается запустить. Для этого используется нотация составного имени пакета, которая позволяет указывать не только имя запускаемого пакета, но и конкретный модуль, метод или режим, который требуется запустить. Для этого используется запись т.н. объекта запуска, которая представляет

собой строку идентификаторов, разделенных точками. Ниже приведены примеры названий объектов запуска и их описания:

ORCA.DFT – пакет ORCA в режиме работы по методу DFT;

ScienceVis.ThreeDGraph.Mesh – запуск научного визуализатора, вызов его модуля ThreeDGraph и построение графика методом Mesh.

Такая нотация позволяет очень гибко описывать встраиваемые в комплекс пакеты и делать вызов отдельных часто используемых режимов и модулей более прозрачным.

Порядок передачи управления между шагами. Так как WF представляет собой ориентированный граф, в EasyFlow введены возможности по организации его структуры, а именно механизмы определения порядка выполнения шагов. Они делятся на два вида: зависимости по управлению и зависимости по данным.

Зависимости по управлению представляют собой явные указания на то, что шаг A должен начать свое исполнение после шага B с помощью директивы *after*, как это показано в листинге 3.20.

Листинг 3.20. Пример зависимостей по управлению

```
step A1 runs Pkg0 ();

step A2 runs Pkg1 ();

step B runs Pkg2 after A2 ();

step C runs Pkg3 after A2 ();

step D runs Pkg4 after B, C, A1 ();
```

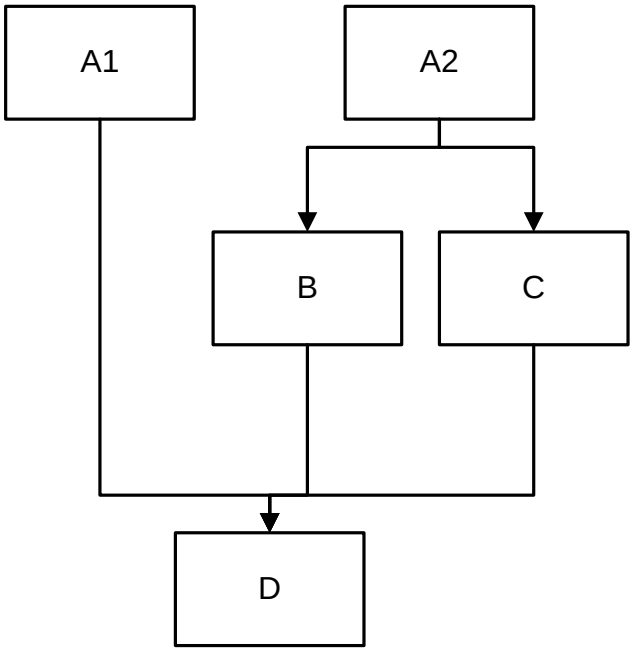

Рисунок 3.3 – Иллюстрация к листингу 3.20

Как видно из листинга, шаги A1 и A2 выполняются первыми, так как не зависят ни от каких других шагов. Шаги B и C выполняются после шага A1, а шаг D выполняется по завершении шагов B, C и A2. Эта запись может быть представлена в виде графа на рис. 3.3.

Зависимости по данным представляют собой неявные указания на зависимости между шагами, которые анализируются при интерпретации скрипта EasyFlow. Данный тип зависимости определяет порядок выполнения шагов: зависимые шаги могут быть запущены только после готовности всех требуемых для их работы данных. Такие зависимости могут присутствовать в описываемом WF одновременно с зависимостями по управлению, что позволяет очень гибко настраивать порядок выполнения шагов. Пример зависимостей по данным приведен в листинге 3.21, а соответствующий этому описанию граф – на рис. 3.4.

Листинг 3.21. Пример зависимостей по данным и управлению

```
step A1 runs Pkg0 ();

step A2 runs Pkg1 ();

step B runs Pkg2
(
  inFile = A1.outs["out.txt"]
);

step C runs Pkg3 after A2 ();

step D runs Pkg4 after C, A2 ();
```

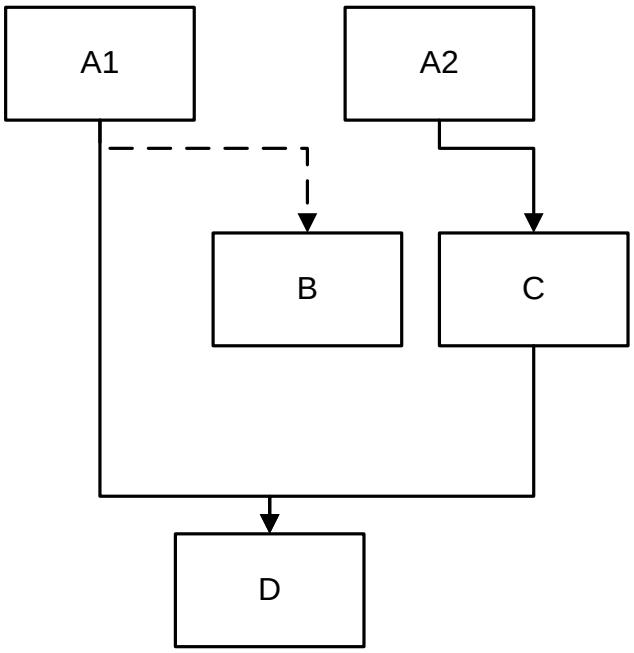

Рисунок 3.4 – Иллюстрация к листингу 3.21

Данный пример отличается от листинга 3.20 тем, что в описании скрипта шаг B зависит от шага A1 по данным, т.к. явно использует его результаты (A1.outs["out.txt"]).

Автоматический перебор параметров. Для удобства пользователя при проведении экспериментов в EasyFlow введена возможность автоматического варьирования 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 делает N шагов. Здесь N – количество элементов в декартовом произведении списков варьирования для различных параметров.

Пример варьирования по двум параметрам (iterations и precision) представлен в листинге 3.22.

Листинг 3.22. Пример варьирования параметров

```
step SweepExample runs SomePackage
(
  width = 100,
  height = 200,
  precision = [0.1, 0.01]
  iterations = sweep [100, 200, 300]
);
```

В данном примере требуется запустить пакет SomePackage, варьируя два параметра (iterations и precision) и закрепив два других (width и height). При запуске будет создано

шесть шагов с наборами пар (precision, iterations): (0.1, 100), (0.1, 200), (0.1, 300), (0.01, 100), (0.01, 200), (0.01, 300).

3.3. Специализированные элементы языка для платформы МИТП-Д

В данном разделе перечислены элементы, обладающие особой значимостью в рамках задач, соответствующих платформе МИТП-Д.

3.3.1. Запуск workflow длительного выполнения (LRWF)

Для преодоления ограничений, связанных с пакетным исполнением WF, в платформе CLAVIRE используется специальное расширение модели WF — WF длительного исполнения (Long Running WF), обеспечивающее работу WF в интерактивном виде, долгое время (или бесконечно) с возможностью управления и динамического изменения WF, а также возможностью обмена информацией между шагами во время исполнения.

В качестве абстракции для коммуникационной связи узлов LRWF применяется парадигма реактивного программирования, ориентированная на потоки данных и распространение изменений. Каждый узел WF имеет входные и выходные параметры (длительного исполнения). Связи можно устанавливать между выходным параметром одного узла и входным – другого узла. Когда изменяется значение выходного параметра, все завязанные на него входные параметры других узлов изменяют свое значение. В результате этого процесс, входной параметр которого изменился, учитывает новое значение в своей работе и может пересчитать значения своих выходных параметров. Таким образом, происходит распространение изменений по всем связанным узлам WF.

С точки зрения подсистемы исполнения вычислительный пакет длительного исполнения – это обычный узел пакетного запуска. Поэтому для такого пакета сохраняются все возможности по описанию пакетного режима запуска и приобретаются дополнительные. В таком описании пакет помечается ключевым словом, которое говорит о том, что данный пакет поддерживает режим длительного исполнения.

Для задания WF длительного исполнения в виде скрипта в язык EasyFlow были внедрены синтаксические конструкции, которые позволяют помечать узел как узел длительного исполнения; задавать коммуникационные зависимости путем связи выходных и входных параметров.

Листинг 3.23. Фрагмент описания WF длительного исполнения на EasyFlow

```
~step Camera runs crowdg (
 delay = 30,
 agent_count = 10000,
 agent_r = 0.4,
 tracing = true
);
 ~step Model runs crowdm (
 isLr = true,
 needPress = false,
 time_step = 0.2,
 agent_count = 10000,
 tracing = true,
 init_agent_pos <- Camera.init_agent_pos
);
```

.

В листинге 3.23 представлен пример скрипта на языке EasyFlow, описывающий WF длительного исполнения. В отличие от узлов пакетного запуска данные узлы помечены ключевым словом ~step, которое говорит о том, что это узлы длительного исполнения (строки 1 и 7). К тому же в скрипте задана коммуникационная связь между узлами (строка 13): выходной параметр init_agent_pos узла Camera связан с одноименным входным параметром узла Model.

4. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

Для работы с МИТП не требуется специальных видов входных данных. В ходе работы входными данными могут являться любые входные файлы, соответствующие по формату запускаемым прикладным сервисам (в текстовом, цифровом, графическом виде), а также данные, вводимые пользователем с клавиатуры по запросу сервиса. В случае несоответствия данных условиям их использования будет выдано системное сообщение. Для их приведения к общему формату используется технология описания на основе языка EasyPackage.

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

5. СООБЩЕНИЯ

В данном разделе определяются основные сообщения программисту, информирующие об ошибке, возникшей в процессе работы с МИТП. Сообщения программисту могут выдаваться двумя способами.

1. *Посредством пользовательского интерфейса*. В этом случае программист получает сообщение в виде всплывающего окна, информирующего о возникновении исключительной ситуации (рис. 5.1).

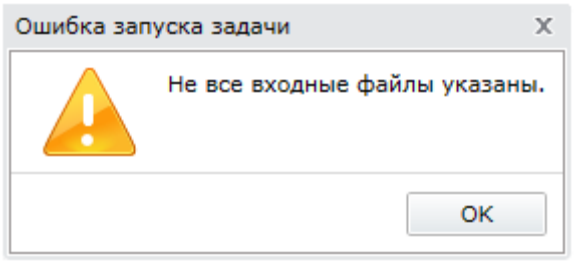

Рис. 5.1. Сообщение об исключительной ситуации (пример)

2. *В журнал событий МИТП*. В журнале событий приводится развернутая информация, необходимая для устранения ошибки. Доступ к журналу сообщений МИТП и ее компонентов программист может получить с использованием возможностей компонента средств мониторинга и настройки основных компонентов платформы CLAVIRE/AdminTool. Действуя в соответствии с документом «Методика использования компонента мониторинга и настройки основных компонентов платформы CLAVIRE/AdminTool» RU.СНАБ.80066-06 И3 64, программист может получить доступ к просмотру внутренних журналов платформы и ее компонентов (рис. 5.2).

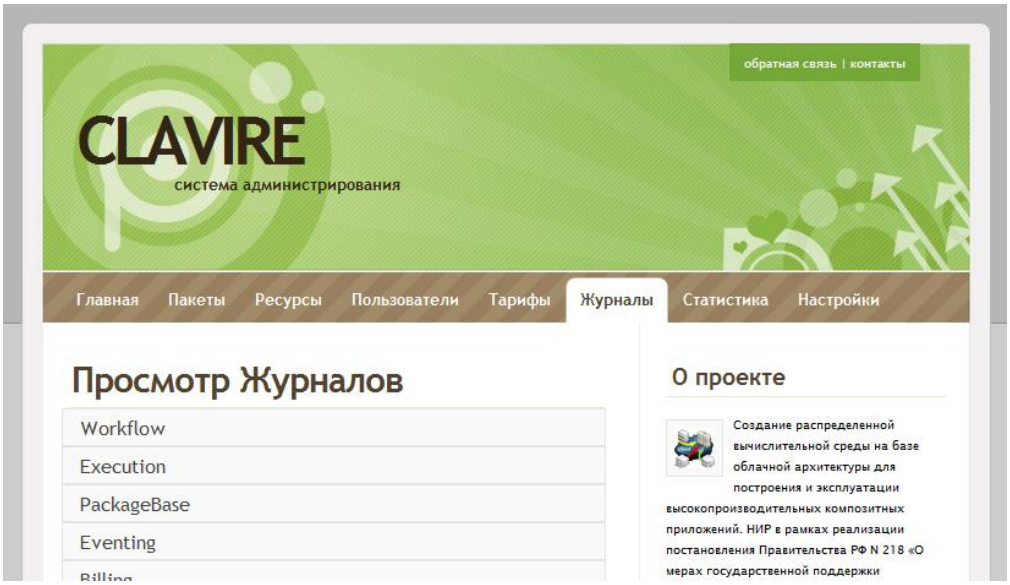

Рисунок 5.2 – Журналы работы компонентов платформы

При выборе компонента появляется окно просмотра журнала этого компонента (рис. 5.3). Особое внимание стоит обращать на строчки, выделенные красным, – это сообщения об ошибках в работе комплекса.

Рисунок 5.3 – Журнал системных сообщений МИТП

Перечень наиболее важных сообщений, выдаваемых компонентами МИТП, приведен в табл. 5.1.

Таблица 5.1

Основные сообщения программисту и оператору

Сообщение	Типовые действия по выявлению и устранению ошибки
Ошибки входа в систему	
Введенная пара логин / пароль неверна	Проверить введенные логин и пароль на корректность, проверить состояние клавиши CapsLock
Ваша учетная запись отключена	Обратиться к администратору комплекса по поводу отключения учетной записи
Система находится на профилактике, попробуйте позднее	Повторить попытку входа через 3–5 минут
Сервер возвратил ошибку: NotFound	Обратиться в службу поддержки комплекса за устранением ошибки
Нет соединения с сетью	Проверить наличие соединения с Интернетом и попробовать снова выполнить операцию
Истекло время ожидания ответа от сервера	Проверить наличие соединения с Интернетом и попробовать снова выполнить операцию. В случае повторения обратиться в службу поддержки комплекса
Соединение принудительно прервано сервером	Попробовать снова выполнить операцию. В случае повторения обратиться в службу поддержки комплекса
Ошибки работы с проектами и задачами	
Не удалось сохранить проект	Проверить наличие соединения с Интернетом и попробовать снова выполнить операцию. В случае повторения обратиться в службу поддержки комплекса
Не все входные данные задачи указаны	Указать все требуемые для задачи файлы и повторить операции
Невозможно запустить задачу: недостаточно свободных ресурсов	Подождать 3–5 минут и попробовать выполнить операцию снова. В случае повторения обратиться в службу поддержки комплекса
Системные ошибки	
Нехватка места на жестком диске	Проверить корректность работы программных компонентов, установленных на указанной ЭВМ, на предмет бесконтрольного заполнения свободного пространства на жестком диске, осуществить переконфигурацию программных модулей, принять меры к увеличению свободного пространства на жестком диске
Нехватка оперативной памяти	Проверить корректность работы программных компонентов, установленных на указанной ЭВМ, на предмет бесконтрольного заполнения пространства оперативной памяти, осуществить переконфигурацию программных модулей, принять меры к увеличению объема оперативной памяти

Ошибки при разработке программ для обработки сверхбольших объемов данных	
Ошибка определения структуры LRWF	Проверить корректность структуры LRWF в сформированном скрипте EasyFlow
Ошибка запуска LRWF	Проверить доступность ресурсов в режиме длительных вычислений. Проверить наличие прав на запуск WF длительного исполнения
Ошибка краулинга	Проверить настройку сервиса сбора данных. Проверить корректность описания сбора данных в сткрипте EasyFlow

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

WF	Поток заданий, workflow
AWF	Абстрактный WF
CWF	Конкретный WF
ННС	Национальная нанотехнологическая сеть
MWF	Мета-WF

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Лист регистрации изменений									
Изм.	**Номера листов (страниц)**				**Всего листов (страниц) в докум.**	**№ документа**	**Входящий № сопроводительного докум. и дата**	**Подп.**	**Дата**
	изменённых	**заменённых**	**новых**	**аннулированных**					