

В серии:

Библиотека ALT Linux

Системы управления бизнес-процессами и административными регламентами

На примере свободной программы RunaWFE

Андрей Михеев

Москва
Альт Линукс
2011

УДК 004.97
ББК 32.97
О60

Андрей Михеев

О60 Системы управления бизнес-процессами и административными регламентами на примере свободной программы RunaWFE / А. Михеев / М.: ALT Linux, 2011. — 178с.: ил. — (Библиотека ALT Linux).

ISBN 978-5-905167-04-1

Книга посвящена системам управления бизнес-процессами и административными регламентами, состоит из двух частей.

В первой части книги дана классификация типов компьютерных систем, автоматизирующих процессное управление, рассказано о связанных с ними стандартах и концепциях, Workflow-паттернах, математических теориях, борьбе международных коалиций, разрабатывающих стандарты. Дан обзор и сравнение нескольких языков управления бизнес-процессами и графических нотаций. Во второй части книги дано описание основных элементов систем управления бизнес-процессами и административными регламентами. Описание дано на примере свободной системы с открытым кодом RunaWFE.

УДК 004.91
ББК 32.9

**По вопросам приобретения обращаться:
В компанию «Альт Линукс»: (495) 662-38-83
E-mail: zakaz@altlinux.ru
Internet-магазин: <http://shop.altlinux.ru>**

Материалы, составляющие данную книгу, распространяются на условиях лицензии GNU FDL. Книга содержит следующий текст, помещаемый на первую страницу обложки: «В серии «Библиотека ALT Linux». Название: «Системы управления бизнес-процессами и административными регламентами на примере свободной программы RunaWFE». Книга не содержит неизменяемых разделов. Авторы разделов указаны в заголовках соответствующих разделов.

ALT Linux — торговая марка компании ALT Linux.

Linux — торговая марка Линуса Торвальдса. Прочие встречающиеся названия могут являться торговыми марками соответствующих владельцев.

Оглавление

1 ВВЕДЕНИЕ	5
2. ИСПОЛНИМЫЕ БИЗНЕС-ПРОЦЕССЫ И АДМИНИСТРАТИВНЫЕ РЕГЛАМЕНТЫ	5
2.1 ВВЕДЕНИЕ	5
2.2 ПРОЦЕССНЫЙ ПОДХОД К ОРГАНИЗАЦИИ УПРАВЛЕНИЯ ПРЕДПРИЯТИЕМ ...	6
2.2.1 Преимущества исполнимых процессов, причины появления СУБПиАР	6
2.2.2 СУБПиАР, потоки работ и задача интеграции масштаба предприятия	7
2.3 ОПРЕДЕЛЕНИЕ ИСПОЛНИМОГО БИЗНЕС-ПРОЦЕССА (АДМИНИСТРАТИВНОГО РЕГЛАМЕНТА)	8
2.3.1 Перспектива управления потоком	9
2.3.2 Перспектива данных	14
2.3.3 Перспектива ресурсов	14
2.3.4 Перспектива операций	16
2.4 «ВОЙНА» СТАНДАРТОВ, СВЯЗАННЫХ С СИСТЕМАМИ УПРАВЛЕНИЯ БИЗНЕС-ПРОЦЕССАМИ И АДМИНИСТРАТИВНЫМИ РЕГЛАМЕНТАМИ	16
2.4.1 Стандарты коалиции WfMC	18
2.4.2 Стандарты коалиции BPMI и консорциума OMG	19
2.4.3 Стандарт BPML	20
2.4.4 Математические основы языков определения бизнес-процессов	21
2.4.4.2 Теория сетей Петри	21
2.4.4.3 Концепция Пи-исчисление	22
2.4.5 Выводы	22
2.5 WORKFLOW ПАТТЕРНЫ	23
2.5.1 Зачем нужны Workflow-паттерны	23
2.5.2 Эволюция Workflow-паттернов	24
2.5.3 Наиболее распространенные Workflow-паттерны	24
2.5.4 Применение паттернов для сравнения языков определения бизнес-процессов и графических нотаций	28
2.6 ЯЗЫКИ ОПИСАНИЯ БИЗНЕС-ПРОЦЕССОВ	29
2.6.1 Краткое описание основных конструкций языка XPDL	30
2.6.2 Краткое описание языка BPMML	33
2.6.3 Краткое описание языка BPML	35
2.6.4 jPDL — «нестандартный» язык, ориентированный на поддержку основных WF-паттернов	36
2.6.5 Сравнение языков при помощи Workflow-паттернов	38
2.6.6 Графические нотации	44
2.6.7 Сравнение графических нотаций	47
2.7 ВОЗМОЖНОЕ РАСШИРЕНИЕ СТАНДАРТОВ НА СИТУАЦИИ, КОТОРЫЕ СУЩЕСТВУЮЩИМИ СТАНДАРТАМИ НЕ ОПИСЫВАЮТСЯ	49
2.7.1 Возможные варианты взаимодействия последовательно идущих по одному переходу точек управления с узлом-действием	50
2.7.2 Возможные варианты взаимодействия последовательно идущих по одному переходу точек управления с разделением и слиянием	51
2.7.3 Назначение заместителя для исполнителя заданий	53
2.8 ВОЗМОЖНОЕ ПРИМЕНЕНИЕ СУБПИАР ДЛЯ АВТОМАТИЗАЦИИ ДЕЯТЕЛЬНОСТИ ГОСУДАРСТВЕННОГО ВЕДОМСТВА	54

3. СВОБОДНАЯ СИСТЕМА УПРАВЛЕНИЯ БИЗНЕС-ПРОЦЕССАМИ И АДМИНИСТРАТИВНЫМИ РЕГЛАМЕНТАМИ С ОТКРЫТЫМ КОДОМ	57
3.1 ВВЕДЕНИЕ	57
3.2 ОСНОВНЫЕ КОМПОНЕНТЫ СИСТЕМЫ	57
3.3 ГДЕ СКАЧАТЬ ИСХОДНЫЕ ФАЙЛЫ И КАК УСТАНОВИТЬ СИСТЕМУ	59
3.3.1 Установка системы при помощи специализированных дистрибутивов для операционной системы Linux	60
3.3.2 Установка системы при помощи специализированного дистрибутива для операционной системы Windows	61
3.4 РАБОТА С СИСТЕМОЙ	67
3.4.1 Вход в систему	67
3.4.2 Меню системы	68
3.4.3 Список заданий	70
3.4.4 Проигрыватель форм	71
3.4.5 Определения процессов и запуск процессов	72
3.4.6 Запущенные процессы	78
3.4.7 Управление пользователями и группами пользователей	82
3.4.8 Пункт меню «Система»	87
3.4.9 Бот-станции	87
3.4.10 Подпроцессы	96
3.4.11 Описание системы безопасности	96
3.4.12 Работа с заданиями	98
3.4.13 Система заместителей	100
3.5 ГРАФИЧЕСКИЙ РЕДАКТОР БИЗНЕС-ПРОЦЕССОВ	102
3.5.1 Создание нового проекта процессов	102
3.5.2 Создание ролей	107
3.5.3 Создание графических форм	110
3.5.4 Создание файла-архива бизнес-процесса	111
3.5.5 Загрузка бизнес-процесса в систему	112
3.5.6 Выполнение процесса	113
3.5.7 Создание переменных	120
3.6 КЛИЕНТ-ОПОВЕЩАТЕЛЬ О ПОСТУПИВШИХ ЗАДАНИЯХ	133
3.6.1 Иконка клиента-оповещателя о поступивших заданиях	133
3.6.2 Оповещение о поступивших заданиях	134
3.6.3 Интерфейс клиента-оповещателя о поступивших заданиях	135
3.7 ПРИМЕРЫ ВЫПОЛНЕНИЯ ПРОСТЕЙШИХ БИЗНЕС-ПРОЦЕССОВ	135
3.7.1 Описание выполнения бизнес-процесса «Сверхурочные»	138
3.7.2 Описание выполнения бизнес-процесса «Отпуск»	143
3.7.3 Описание исполнения бизнес-процесса «Командировка»	146
3.7.4 Описание демо-процесса «Загрузить, Скачать файл»	147
3.7.5 Описание исполнения бизнес-процесса «Отчет»	149
3.7.6 Описание исполнения бизнес-процесса «Создание текстового файла»	151
3.7.7 Описание исполнения бизнес-процесса «Таймер»	152
3.7.8 Описание исполнения бизнес-процесса с подпроцессом	154
3.7.9 Описание исполнения бизнес-процесса с мульти-действием	155
3.8 ПРОЕКТ RUNAWFE	157
4. СПИСОК ЛИТЕРАТУРЫ	158

1. Введение

В первой части книги рассматриваются бизнес-процессы и административные регламенты, выполняющиеся в компьютерных системах. Они позволяют исключить из действий сотрудников рутинные операции, неэффективные процедуры, связанные с поиском и передачей информации, существенно повысить скорость взаимодействия сотрудников. При использовании системы управления бизнес-процессами и административными регламентами работники предприятия могут выполнять поступившие задачи, не отвлекаясь на получение от других работников необходимой для выполнения задания информации, передачу результатов своего труда другим работникам, изучение должностных инструкций. Вся необходимая информация возникает перед работником на экране компьютера.

Использование бизнес-процессов и административных регламентов также позволяет оперативно перестраивать бизнес-процессы организации и административные регламенты ведомства. Во многих случаях исполнителей заданий можно даже не информировать об изменении бизнес-процесса, так как это не отразится на характере их работы. То есть получается легче и быстрее изменять выполнение процессов. Таким образом предприятие или ведомство может более эффективно реагировать на изменение внутренних или внешних условий.

Во второй части дано описание основных элементов систем управления бизнес-процессами и административными регламентами. Описание дано на примере свободной системы с открытым кодом RunaWFE. RunaWFE свободно распространяется на условиях открытой лицензии LGPL. Система бесплатная. Скачать дистрибутивы и исходный код ее можно через интернет с портала разработчиков свободного программного обеспечения [sourceforge.net](http://sourceforge.net/projects/runawfe) по адресу: <http://sourceforge.net/projects/runawfe>.

Система RunaWFE включена в репозиторий свободных программ Sisyphus (<http://sisyphus.ru>). На основе RunaWFE и дистрибутива ALT Linux компания Альт Линукс выпустила специализированный дистрибутив ALT Linux Workflow.

Адрес сайта проекта RunaWFE: <http://wf.runa.ru/rus>

2. Исполнимые бизнес-процессы и административные регламенты

2.1 Введение

Первые компьютерные системы, автоматизирующие управление бизнес-процессами появились давно, более пятнадцати лет назад. Тем не менее и сегодня ситуация в данном классе программных продуктов остается непростой и развивается очень динамично. Многими вопросами, относящимися к этим системам, активно занимаются самые разные организации: ведущие софтверные компании, международные консорциумы, комитеты по стандартизации, а также ученые-математики (специалисты по теории графов и алгебрам процессов).

Для исполнимых в компьютерной среде бизнес-процессов и административных регламентов требуется строгое определение бизнес-процесса (административного регламента), такое, чтобы его легко было перевести в представление, понимаемое компьютером. Для этого удобно использовать математические понятия. Такое определение будет дано в настоящей части книги.

Как правило, для переноса описания бизнес-процесса или административного регламента в конкретную исполнимую среду используются языки описания бизнес-процессов. В настоящее время существует большое количество языков описания бизнес-процессов, некоторые из них являются международными стандартами. Между международными коалициями, разрабатывающими эти языки, а также другие связанные с процессным управлением стандарты, идет серьезная конкурентная борьба, которая получила название «война стандартов».

В настоящей части книги будет дана классификация типов компьютерных систем, автоматизирующих управление бизнес-процессами предприятия или административными регламентами ведомства, рассказано о связанных с ними стандартах и концепциях, Workflow-паттернах, математических теориях, борьбе международных коалиций, разрабатывающих относящиеся к процессному управлению стандарты. Будет дан обзор и сравнение нескольких языков управления бизнес-процессами и графических нотаций.

2.2 Процессный подход к организации управления предприятием.

2.2.1 Преимущества исполнимых процессов, причины появления СУБПиАР

В настоящее время процессный подход к организации управления предприятием признан наиболее перспективным. В соответствии с ним деятельность промышленного предприятия или государственного ведомства представляется в виде множества бизнес-процессов (административных регламентов). Быстрая интеграция работы разнородных компьютерных систем и труда сотрудников из различных департаментов в единый бизнес-процесс (административных регламент) позволяет добиться результата за короткий промежуток времени, что является одним из серьезных преимуществ.

Вследствие этого у предприятий и ведомств возникает потребность в гибких компьютерных системах, реализующих процессный подход к управлению. Основанные на нем компьютерные системы получили название системы управления бизнес-процессами и административными регламентами. В данной книге будем использовать для таких систем сокращенный термин — СУБПиАР. Важной характеристикой этих систем является возможность быстрой разработки и изменения бизнес-процессов предприятия (административных регламентов ведомства) без обновления специализированного кода, с использованием лишь графической среды разработки.

Внедрение СУБПиАР также приведет к появлению единого для всех менеджеров предприятия или сотрудников ведомства языка описания бизнес-

процессов, основанного на графических диаграммах. После того как сотрудники организации освоят этот язык, они смогут быстро читать существующие бизнес-процессы, разбираться в состояниях выполняющихся бизнес-процессов и административных регламентов, а также производить быструю сборку из разнородных элементов (труда сотрудников и работы компьютерных систем предприятия) новых качественных бизнес-процессов. Для ускорения процедуры создания новых процессов или регламентов полезно создать на предприятии библиотеку типичных шаблонов.

2.2.2 СУБПиАР, потоки работ и задача интеграции масштаба предприятия

Основная задача СУБПиАР — направлять задания исполнителям и контролировать их исполнение. Как правило, последовательность заданий определяется графом бизнес-процесса, который можно быстро изменять при помощи редактора бизнес-процессов.

Управление бизнес-процессами — активно развивающаяся область, и многие термины в ней еще не устоялись. Различные авторы прибегают к таким понятиям, как СУБПиАР, системы управления потоками работ (Workflow), системы управления документооборотом (Docflow), системы интеграции масштаба предприятия (EAI — Enterprise Application Integration) и т.п.

Мы будем использовать термин управление потоками работ (Workflow) применительно к случаям, когда исполнителями заданий бизнес-процесса являются только люди. Термин СУБПиАР мы будем рассматривать в качестве более общего по отношению к управлению потоками работ: исполнителями заданий бизнес-процесса или регламента в СУБПиАР являются как люди, так и компьютерные приложения. Причем СУБПиАР должна координировать работу всех исполнителей единообразно, не выделяя специальным образом работы, выполняемые человеком или компьютерными системами.

СУБПиАР и системы управления документооборотом

Парадигма СУБПиАР — это «поток (элементов) работ», передаваемых исполнителям в виде заданий. В ней всякую деятельность можно представить в виде элементов работы, перемещающихся по определенному маршруту между исполнителями в соответствии с заданными правилами. При этом от одного исполнителя к другому передается точка управления. Как правило, набор всех возможных маршрутов хорошо представим в виде графа.

Кроме СУБПиАР большое распространение получили системы управления документооборотом, или DocFlow-системы. Парадигму системы управления документооборотом представляет «поток документов». Здесь всякую деятельность можно представить в виде документов, путешествующих между их редакторами по определенному маршруту в соответствии с заданными правилами.

DocFlow-системы являются наследниками бумажного документооборота. Отсюда и их естественные ограничения: с документом можно совершить ограниченный набор действий: одобрить, отказать, визировать, удалить, внести

правку и т. п. Обычно системы документооборота дополняются системами хранения образов бумажных документов и системами версионного контроля. Основным преимуществом систем документооборота является возможность их быстрого внедрения на предприятии, если там уже на хорошем уровне налажен документооборот.

В системах документооборота, так же как и в СУБПиАР, существуют схемы на основе графов, которые состоят из узлов, соединенных возможными переходами. Однако по этим графам перемещаются не точки управления, а «корзины» документов. В DocFlow-системах, как правило, данные содержатся внутри документов, которые непосредственно перемещаются по схеме документооборота.

В СУБПиАР данные не перемещаются вместе с точкой управления, а содержатся в глобальных (соответствуют всему бизнес-процессу) и локальных (соответствуют одному узлу) переменных.

В настоящее время СУБПиАР и системы документооборота представляют собой системы разных типов, однако постепенно системы документооборота по функциональности приближаются к СУБПиАР. При помощи современных DocFlow-систем можно моделировать многие виды бизнес-процессов, а при помощи СУБПиАР — автоматизировать элементы документооборота.

СУБПиАР и задачи интеграции масштаба предприятия

На современном российском предприятии, как правило, уже эксплуатируется несколько разнородных автоматизированных систем, которые участвуют в каких-либо бизнес-процессах предприятия. Так как бизнес-процессы пронизывают все предприятие, в процессе исполнения бизнес-процессов в СУБПиАР ей придется взаимодействовать со всеми этими автоматизированными системами. Таким образом, задача внедрения СУБПиАР оказывается частным случаем задачи интеграции компьютерных приложений масштаба предприятия. Иными словами, при внедрении СУБПиАР на предприятии должно появиться приложение, обеспечивающее ее интеграцию с уже имеющимися системами. В простейшем случае это приложение должно представлять собой компонент, содержащий набор коннекторов к различным системам и базам данных.

СУБПиАР является центральной частью современных систем масштаба предприятия. Если в корпоративной информационной системе (КИС) отсутствует СУБПиАР, то логика бизнес-процессов оказывается рассеянной по различным элементам системы — базам данных, отдельным приложениям и т.д., и такие системы будет крайне сложно сопровождать и развивать дальше.

2.3 Определение исполнимого бизнес-процесса (административного регламента)

В данной книге мы будем рассматривать термины бизнес-процесс и административный регламент как синонимы. Традиционно термин бизнес-процесс используется в случае промышленного предприятия, а административный регламент — государственной организации.

Для бизнес-процессов (регламентов), которые могут быть исполнены в компьютерной среде, необходимо дать более строгое определение, такое, которое легко можно перевести в представление, понимаемое компьютером. Для такого определения удобно использовать математические понятия.

Дадим определение исполнимого бизнес-процесса, основу которого составляют идеи С. Яблонского и С. Бусслера¹.

Исполнимый бизнес-процесс определяется при помощи задания следующих перспектив (точек зрения или слоев/уровней рассмотрения):

- перспектива управления потоком (control-flow perspective)
- перспектива данных (data perspective)
- перспектива ресурсов (resource perspective)
- перспектива операций (operational perspective)

Исполнимый бизнес-процесс можно запускать. Таким образом создаются выполняющиеся экземпляры бизнес-процесса. Отличия экземпляра от определения бизнес-процесса соответствуют отличию типа переменной от экземпляра переменной традиционного языка программирования. То есть определение бизнес-процесса содержит схему бизнес-процесса, типы переменных, названия ролей. В выполняющемся экземпляре бизнес-процесса на схеме находятся перемещающиеся точки управления, экземпляр бизнес-процесса содержит конкретные значения переменных, типы которых соответствуют типам определения бизнес-процесса. Также в экземплярах бизнес-процесса на роли назначаются конкретные исполнители заданий.

Рассмотрим подробно все уровни определения исполнимого бизнес-процесса. При этом в качестве примера будем использовать бизнес-процесс «Оплата счета поставщика». С его помощью постараемся пояснить все уровни определения бизнес-процесса.

2.3.1 Перспектива управления потоком

Перспектива управления потоком соответствует схеме бизнес-процесса. Изначально схема определялась как математическое понятие — направленный граф: множество узлов, соединенных между собой дугами (возможными переходами). Узлы бизнес-процесса могли быть двух типов — узлы, соответствующие шагам процесса, и маршрутные узлы. По переходам перемещается точка управления (указатель на активный узел процесса), руководствуясь правилами в маршрутных узлах.

В узле, соответствующем шагу процесса, находится узел-действие (Activity). Если точка управления пришла в узел-действие, то СУБПиАР дает задание исполнителю (сотруднику или информационной системе) и ждет ответа (сообщения, что работа выполнена). После ответа исполнителя точка управления движется по переходу к следующему узлу процесса. К узлу, соответствующему узлу-действию, может примыкать только один входящий и один исходящий переход.

¹S. Jablonski and C. Bussler. Workflow Management: Modeling Concepts, Architecture, and Implementation. International Thomson Computer Press, London, UK, 1996.

Маршрутный узел соответствует появлению, удалению, разветвлению-слиянию точек управления или выбору перехода, по которому точка управления будет перемещена дальше. В таких узлах СУБПиАР выбирает на основании содержащихся в маршрутных узлах правил следующий узел (узлы), в который будет передано управление. Часто с этими узлами связано более одного входящего или исходящего перехода.

В выполняющемся бизнес-процессе одновременно может быть несколько точек управления. В соответствии с бизнес-логикой процесса точка управления в маршрутном узле может разделиться на несколько точек управления. Также точки управления могут ждать друг друга в определенном маршрутном узле и далее слиться в одну точку управления.

Позже в различных спецификациях данное определение было расширено:

1. Были добавлены комбинированные узлы, представляющие собой слияние шага процесса с одним или несколькими маршрутными узлами. Например, при слиянии узла-действия с находящимся за ним маршрутным узлом, осуществляющим выбор одного из нескольких возможных направлений, в схему помещается только узел-действие и прямо к нему присоединяются переходы, которые должны выходить из маршрутного узла.
2. Были добавлены дополнительные конструкции, элементы которых не являются элементами графа (далее — дополнительные конструкции). Однако к этим элементам могут быть присоединены переходы и маршрутные узлы или же переходы могут пересекать эти элементы. Например, были введены события и области с прерыванием, объемлющие шаги бизнес-процесса. При нахождении точки управления внутри области с прерыванием может произойти событие (клиент может передумать делать заказ, в процессе выполнения договора могут возникнуть форс-мажорные обстоятельства и т.п.). В этом случае точка управления может из любого находящегося внутри области узла сразу переместиться в присоединенный к области маршрутный узел и уже из него продолжить движение по присоединенному к нему переходу.
3. Были добавлены узлы, соответствующие шагу процесса, но не являющиеся узлами-действиями. Например, узлы-ожидания, в которых не дается заданий исполнителям процесса, СУБПиАР просто ожидает в этих узлах наступления определенного события, после которого точка управления идет дальше. Или узлы-подпроцессы. Для этих узлов не определен конкретный исполнитель, в этих узлах СУБПиАР запускает другой бизнес-процесс в качестве подпроцесса текущего процесса и передает ему соответствующие данные.

После расширения понятия шага процесса не совсем понятно, в чем состоит принципиальное отличие шага процесса от маршрутного узла. В маршрутном узле надо только принять решение о дальнейшем пути (путях) движения точки управления на основании уже существующих данных, поэтому точка управления не должна находиться в маршрутном узле долго. На шаге процесса точка управления может находиться длительное время.

С учетом дополнений перспектива управления потоком можно определить следующим образом:

Перспектива управления потоком представляет собой схему бизнес-процесса. Схема бизнес-процесса состоит из направленного графа и, возможно, дополнительных конструкций. Узлы бизнес-процесса могут быть трех типов — узлы, соответствующие шагам процесса, маршрутные узлы и комбинированные узлы, представляющие собой слияние шага процесса с одним или несколькими маршрутными узлами.

Шаги процессов являются узлами-действиями или дополнительными узлами. По переходам перемещаются точки управления. В момент прихода точки управления в узел-действие СУБПиАР дает задание исполнителю. После выполнения задания исполнителем точка управления движется по переходу к следующему узлу процесса. К узлу, соответствующему узлу-действию, может примыкать только один входящий и один исходящий переход.

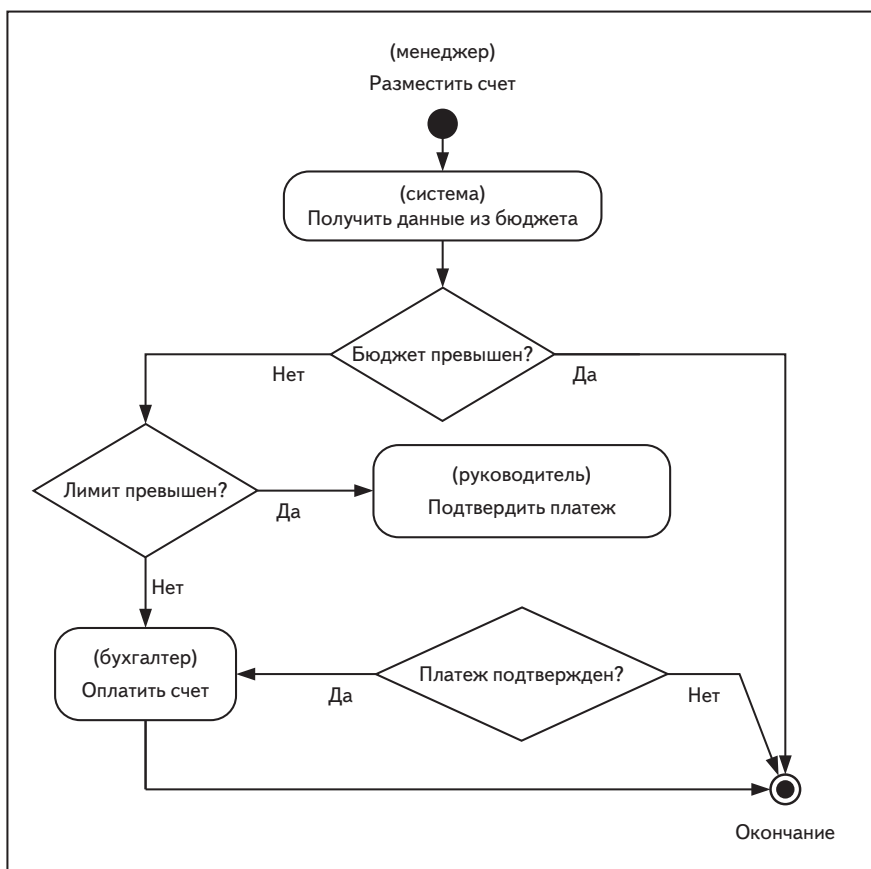


Рисунок 2.3.1. Пример схемы бизнес-процесса «Оплата счета поставщика» (UML-нотация).

Маршрутный узел (вентиль) соответствует появлению, удалению, разделению, слиянию точек управления или выбору перехода. В этих узлах СУБПиАР выбирает на основании содержащихся в маршрутных узлах правил следующий узел (узлы), в который будет передано управление.

На рисунке 2.3.1 приведен пример графа бизнес-процесса «Оплата счета поставщика». Шаги процесса изображены в виде прямоугольников со скругленными краями, началу процесса соответствует черный кружок, завершению — окружность с черным кружком внутри. Элемент «Оплатить счет» является комбинированным узлом, представляющим собой композицию маршрутного узла соединения переходов и узел-действие. Остальные прямоугольники со скругленными краями являются узлами-действиями. Элементы в виде ромбов соответствуют маршрутным узлам — местам разветвления маршрутов точек управления.

В начале бизнес-процесса бизнес-менеджер поставок вводит параметры предполагаемого платежа (номер счета, дата счета, сумма счета, фирма-контрагент, фирма-агент, комментарий). Далее автоматически производится контроль исполнения бюджета подразделения. Если текущая сделка превышает бюджет, то она автоматически отклоняется, и бизнес-процесс завершается. Если бюджет подразделения не превышен, сумма сделки сравнивается с лимитом платежа. Далее, если лимит не превышен, автоматически происходит оплата счета, после чего бизнес-процесс завершается. При превышении лимита необходимо, чтобы платеж был подтвержден финансовым директором.

Бизнес-процессу «Оплата счета поставщика» соответствуют следующие логические правила:

1. Если внешнее приложение, вызванное в узле «получить данные из бюджета», вернуло значение «нет» переменной «Превышен ли бюджет подразделения», то следует перейти к проверке лимита, в противном случае — перейти в узел завершения бизнес-процесса.
2. Если значение переменной «сумма счета» меньше значения константы «лимит разового платежа», нужно перейти к узлу «оплата счета», в противном случае — к узлу «подтвердить платеж».
3. Если исполнитель, принадлежащий к роли «Финансовый директор», заполняя поля в соответствующей форме, вернул значение «да» в переменную «утвердил ли руководитель», то перейти к узлу «оплата счета», в противном случае — к узлу завершения бизнес-процесса.

Данная диаграмма очень напоминает блок-схему алгоритма, так как здесь не происходит «размножения» точек управления.

Рассмотрим еще один пример, показывающий, что бизнес-процессы обладают существенным параллелизмом и в этом случае языком обычных алгоритмических блок-схем уже не описываются.

На рисунке 2.3.2 изображен пример схемы, соответствующей этапу оформления очередного отпуска сотрудником предприятия.
Данный пример иллюстрирует следующее:

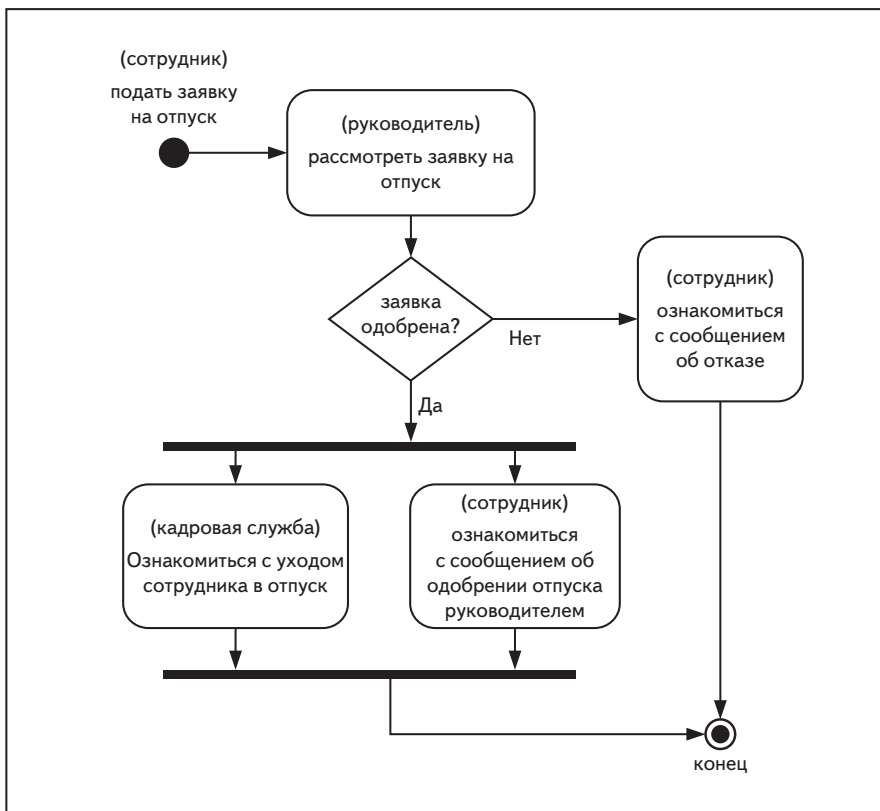


Рисунок 2.3.2. Схема бизнес-процесса рассмотрения заявления об уходе в отпуск (UML-нотация).

1. Правила, в соответствии с которыми выбирается исполнитель текущего задания, могут быть достаточно сложными. На втором шаге бизнес-процесса это правило следующее: задание направляется начальнику сотрудника, выполнившего предыдущий шаг (подавшего заявление на отпуск).
2. Управление бизнес-процессом также может быть сложным и отличаться от поведения точки управления в традиционной блок-схеме алгоритма. В данном примере в случае одобрения заявления начальником, поток управления разделяется на два параллельных потока, выполняющихся одновременно, которые потом «сливаются» в одной точке.

2.3.2 Перспектива данных

Перспектива данных соответствует набору внутренних переменных бизнес-процесса. Переменные бизнес-процесса могут являться входящими и исходящими параметрами при взаимодействии СУБПиАР с информационными системами предприятия. При помощи переменных происходит обмен информацией между шагами процесса и, как следствие, между внешними информационными системами, т. е. бизнес-процесс может переносить информацию в корпоративной информационной среде между разнородными информационными системами. Переменные бизнес-процесса также используются при выборе конкретного внутреннего перемещения точки управления между узлами по какому-либо из возможных переходов.

Таблица 2.3.1. Список глобальных переменных, соответствующих бизнес-процессу «Оплата счета», схема которого изображена на рисунке 2.3.1

Название переменной	Тип переменной
Номер счета	Строка
Дата счета	Дата
Id (идентификационный номер) фирмы контрагента (юридического лица, на которое выписан счет)	Число — уникальный идентификатор
Id фирмы-агента (юридического лица, которое будет осуществлять платеж)	Число — уникальный идентификатор
Комментарий	Многострочный текст
Превышен ли бюджет подразделения	Логический (да/нет)
Лимит разового платежа	Число
Утвердил ли руководитель	Логический (да/нет)

2.3.3 Перспектива ресурсов

Перспективе ресурсов бизнес-процесса соответствует набор исполнителей, которые могут выполнять его узлы-действия. Исполнителями могут быть как сотрудники предприятия, так и информационные системы или специализированные устройства.

В бизнес-процессе производится связывание узлов-действий с исполнителями заданий при помощи ролей. При разработке бизнес-процесса создается роль и ставится в соответствие определенным узлам-действиям. Во время выполнения бизнес-процесса ролям назначаются конкретные исполнители. Здесь можно провести аналогию с театральным спектаклем. В процессе написании сценария определяются используемые в спектакле роли.

Потом, при постановке в конкретном театре, на роли назначаются актеры — исполнители ролей. Например, роль может называться «Эдмон Дантес», а исполнителем быть — заслуженный артист Петров. Может даже так быть, что у роли «Эдмон Дантес» в спектакле в разные моменты времени будут разные исполнители, например, исполнителем роли Эдмона Дантеса в юности будет Иванов, а исполнителем роли Эдмона Дантеса в зрелые годы — артист Петров. В отличие от театра, в узле-действии бизнес-процесса может быть сразу несколько исполнителей роли.

В бизнес-процессе также могут быть различные правила выполнения заданий. Например, бизнес-процесс может послать задание на выполнение всем членам некоторой группы пользователей, а выполнять это задание будет первый пользователь, взявший задание на выполнение. У остальных членов группы это задание будет отозвано. Существуют бизнес-процессы, в которых, наоборот, требуется, чтобы все члены группы выполнили задание.

Некоторые СУБПиАР позволяют задать «взвешенные» правила распределения заданий по членам группы. В этом случае работа между ними распределяется в зависимости от их весовых коэффициентов, задаваемых в организационном компоненте СУБПиАР, и от количества заданий, уже принятых пользователями. Например, если группа содержит трех сотрудников с весами 20%, 30% и 50%, то при прохождении заданий первому сотруднику будет направлено на выполнение 20% от их общего числа, второму — 30%, а третьему — 50%.

Данная перспектива плотно связана с организационной моделью и моделью информационных систем предприятия.

Бизнес-процесс «Оплата счета поставщика» предполагает следующую структуру исполнителей, объединенных в соответствующие группы:

Сотрудники:

- менеджер поставок
- финансовый директор
- бухгалтер

Информационные системы предприятия:

- система контроля бюджета

Таблица 2.3.2. Описание перспективы ресурсов бизнес-процесса «Оплата счета поставщика»

Шаг	Роль	Исполнитель шага
Разместить счет	Менеджер	Конкретный менеджер поставок
Получить данные из бюджета	Система	Компьютерная система контроля бюджета
Подтвердить платеж	Руководитель	Финансовый директор предприятия
Оплатить счет	Бухгалтер	Бухгалтер, ответственный за платежи

2.3.4 Перспектива операций

Перспективе операций бизнес-процесса соответствует список элементарных действий, совершаемых исполнителями в рамках узла-действия.

Для сотрудника предприятия это будет набор операций, фиксируемый в визуальной форме, доступной ему на этапе исполнения шага. Для информационных систем предприятия — набор запросов или транзакций, позволяющих манипулировать данными через специальные интерфейсы.

Таблица 2.3.3. Перспектива операций бизнес-процесса «Оплата счета поставщика».

Шаг	Операция	Исполнитель операции
Разместить счет	Заполнить форму размещения счета и запустить экземпляр бизнес-процесса	Менеджер поставок
Получить данные из бюджета	Провести авторизацию	Компьютерная система контроля бюджета
	Получить остаток средств, доступных для закупок по департаменту	
Подтвердить платеж	Заполнить форму подтверждения / не подтверждения платежа	Финансовый директор
Оплатить счет	Провести платеж на указанную сумму и отметить это в форме	Бухгалтер, ответственный за платежи

2.4 «Война» стандартов, связанных с системами управления бизнес-процессами и административными регламентами

Область, к которой относятся СУБПиАР, сегодня активно развивается как в теории (предлагаются новые концепции, развиваются математические теории), так и в бизнес-сфере (появляется большое количество различных программных продуктов). Однако большинство СУБПиАР плохо совместимо между собой, так как системы реализуют разные интерфейсы взаимодействия. Их описания нередко даны в разной терминологии, и их трудно сравнивать. Если аналитик разобрался в одной системе, то при изучении следующей ему часто приходится начинать все сначала, так как она описана в других понятиях, имеет другой механизм взаимодействия компонентов. В этих условиях жизнь сильно облегчили бы единые стандарты для СУБПиАР.

Зачем вообще нужны стандарты для СУБПиАР? Стандарты полезны разработчику, поскольку позволяют:

- сосредоточиться на эффективном решении технических вопросов реализации. Не надо решать общие концептуальные проблемы;
- реализовывать не всю систему, а только ее части. Следование единому стандарту гарантирует, что разработанный компонент будет работать с компонентами других производителей в рамках данного стандарта.

Стандарты полезны пользователю системы, поскольку:

- можно будет сменить СУБПиАР на другую систему в рамках того же стандарта (если фирма-разработчик по каким-либо причинам перестанет поддерживать систему). Все определения старых бизнес-процессов будут работать в новой системе;

- исследования возможности описания бизнес-процессов того или иного класса можно провести не для каждой системы, а для стандарта. Далее можно, не исследуя функциональность конкретной системы, заключить, что если эта система удовлетворяет стандарту, то она гарантированно поддерживает процессы соответствующего типа.

Конечно, было бы удобно, если бы существовал общепринятый промышленный стандарт для СУБПиАР. Однако в настоящее время ситуация со стандартами сложная. Проблема в том, что стандартов для СУБПиАР слишком много. Различными международными сообществами разработано множество конкурирующих спецификаций и количество их постоянно возрастает. Данная ситуация получила название «война стандартов».

Если в 1995 г. стандартами для СУБПиАР занималась только коалиция WfMC, то к 2003-у году было уже 10 групп, разрабатывающих стандарты, относящиеся к СУБПиАР, и только для описания бизнес-процесса они предложили 7 различных стандартов. В настоящее время количество реально используемых стандартов уменьшилось, однако ситуация со стандартами все равно остается сложной. Война стандартов продолжается.

В таблице 2.4.1 приведен список наиболее известных международных коалиций, разработавших спецификации, относящиеся к СУБПиАР.

Таблица 2.4.1.

Коалиции	Спецификации, описывающие		
	архитектуру СУБПиАР	языки определения бизнес-процессов	графические нотации, которые используются при описании бизнес-процессов
WfMC www.wfmc.org	Workflow reference model	WPDЛ, Wf-XML, XPDL	
OMG www.omg.org	Workflow Management Facility Specification		Activity диаграмма языка UML
BPMI (в настоящее время поглощена коалицией OMG) www.bpmi.org		BPML	BPMN
Изначально — коалиция IBM, Microsoft, BEA, SAP, Siebel, в настоящее время - консорциум OASIS www.oasis-open.org		BPЕL	

2.4.1 Стандарты коалиции WfMC

Коалиция WfMC (Workflow Management Coalition) образована в 1993 г. и была первой группой, предложившей спецификации для СУБПиАР, многие из которых впоследствии несколько раз переписывались на основе различных ИТ-концепций. WfMC разработан словарь Terminology & Glossary, где определяются основные термины, относящиеся к СУБПиАР. Многие из этих терминов на сегодняшний день признаны как стандарт де-факто для описания элементов бизнес-процесса.

В спецификации Workflow reference model предлагается следующая общая архитектура для СУБПиАР:

- распределенное ядро системы, которое содержит набор выполняемых экземпляров бизнес-процессов;
- редактор определений бизнес-процессов;
- клиентское приложение, при помощи которого ядро взаимодействует с пользователями;
- внешние приложения, вызываемые СУБПиАР;
- административное приложение.

Стандарт предполагает, что все компоненты взаимодействуют не напрямую друг с другом, а только с распределенным ядром системы. Стандарт не оговаривает детально, как должны быть устроены компоненты. В основном в нем описываются интерфейсы взаимодействия этих компонентов с ядром системы. В «Workflow Reference Model» интерфейсы описаны неформально — практически в терминах предметной области. В дополнительных документах интерфейсы определены более строго. Всего предлагается пять интерфейсов:

1. первый описывает взаимодействие ядра системы с редактором определений бизнес-процессов;
2. второй — взаимодействие ядра с клиентским приложением;
3. третий — взаимодействие ядра с внешними приложениями;
4. четвертый соответствует взаимодействию друг с другом компонентов ядра системы, находящихся на различных компьютерах в распределенной сети;
5. пятый — взаимодействию ядра с административным приложением.

Разработанный коалицией WfMC в 1999 г. язык определения бизнес-процессов WPDЛ был основан на формулах Бэкуса-Наура. В рамках Workflow reference model язык определения бизнес-процесса соответствует первому интерфейсу. В 2002 г. язык WPDЛ был переписан. Его новая версия XPDL уже была основана на XML.

В дополнительных документах WfMC достаточно подробно и строго описаны и другие интерфейсы, например, WAPI (Workflow Application Programming Interface). Интерфейсы взаимодействия в WAPI не объектноориентированы, а соответствуют процедурному подходу. В приложениях к WAPI сделана попытка перейти от процедурного подхода к объектному (в рамках уже несколько устаревших к настоящему времени концепций OLE и CORBA): процедурные спецификации преобразованы в OLE и IDL-спецификации, которые, однако, по мнению экспертов, сохранили в себе наследие процедурного подхода.

Сначала предполагалось, что таким образом будет описан только второй интерфейс, однако оказалось, что одни и те же функции (объекты) используются различными интерфейсами и писать спецификации по-интерфейсно нет смысла. Тогда документ стал фактически относиться к интерфейсам со второго по пятый.

Следующий шаг в рамках данной эволюции сделала коалиция OMG (Object Management Group). В 2000 г. она выпустила документ Workflow Management Facility Specification. В нем построены основы архитектуры ядра WF-системы. На языке IDL определены основные интерфейсы многих компонентов. Несмотря на то, что в предисловии к документу говорится, что спецификация основана на WAPI WfMC (OMG IDL binding), — это другая спецификация, которая унаследовала только основные принципы построения общей архитектуры системы коалиции WfMC.

Большинство известных нам открытых СУБПиАР, разработчики которых утверждают, что они полностью соответствуют спецификации WfMC, реализуют именно эту архитектуру, предложенную OMG (а не WAPI коалиции WfMC).

Программистские технологии продолжали развиваться. Появились Web-сервисы, вслед за ними языки и спецификации (не совместимые со стандартами коалиции WfMC), ориентированные на эти технологии. В конце 2001 г. WfMC выпустила документ Workflow Standard — Interoperability Wf-XML Binding, в котором для реализации четвертого интерфейса предлагался язык Wf-XML. Этот язык можно использовать в рамках технологии Web-сервисов.

Складывается впечатление, что WfMC постепенно теряет инициативу по сравнению с другими разработчиками стандартов. Если в середине 90-х годов внутри коалиции появлялось много новых интересных идей, то сейчас она часто модернизирует свои спецификации на основе чужих идей, уже примененных в других спецификациях и программных продуктах.

2.4.2 Стандарты коалиции BPMI и консорциума OMG

В 2000 г. появилась коалиция BPMI (Business Process Management Initiative), которая достаточно быстро разработала основанный на технологии Web-сервисов язык определения бизнес-процессов BPML и начала создание других полезных стандартов (не совместимых со стандартами WfMC). Через некоторое время коалиция BPMI подготовила стандарт графических диаграмм, описывающих бизнес-процесс — BPMN. Язык также содержал правила автоматического перевода графических диаграмм BPMN в язык BPML.

Однако вслед за этим компании IBM, Microsoft и BEA (а также присоединившиеся к ним впоследствии SAP и Siebel) создали другой язык определения бизнес-процессов, также основанный на технологии Web-сервисов — BPEL, который в конечном счете, используя маркетинговую мощь создавших его компаний, полностью вытеснил язык BPML.

Ситуация с BPMN оказалась тоже не простой. Консорциум OMG (Object Management Group), который существует с 1989 года и в настоящее время с ним сотрудничает около 800 крупнейших производителей программного обеспечения, разработал к моменту появления BPMN диаграмму Activity в языке UML 2.0 (UML AD).

Эта диаграмма была в некотором смысле альтернативой языку BPMN (по графической выразительной силе эти нотации были примерно одинаковы). В 2005 году консорциум OMG поглотил коалицию BPMI. После поглощения консорциум OMG опубликовал специальное разъяснение относительно спецификаций BPMN и UML AD. В соответствии с этим разъяснением спецификация BPMN должна использоваться для бизнес-моделирования, а спецификация UML AD — для программного моделирования (описания взаимодействия компонентов программного обеспечения).

В настоящее время BPMN — наиболее распространенная графическая нотация для описания бизнес-процессов, однако во многих организациях для разработки бизнес-процессов по-прежнему используется нотация UML AD как вследствие того, что так сложилось исторически, так и из-за того, что в некоторых случаях UML AD обладает преимуществами перед BPMN. Следует признать, что ситуация, в которой для подобных видов деятельности один и тот же консорциум предлагает два разных графических языка (один для аналитиков, другой для программистов), является странной. Если бы и программисты, и аналитики пользовались одним языком, им легче было бы понимать друг друга.

2.4.3 Стандарт BPEL

К моменту образования коалиции BPMI корпорация IBM начала работу над своим стандартом WF-языка (WSFL). Компания Microsoft также приступила к составлению собственного стандарта (XLANG) (оба несовместимы с XPDЛ и BPML). В августе 2002-го IBM, Microsoft и BEA Systems объявили о подготовке совместного стандарта — языка BPEL4WS (или просто BPEL). Позже к этим компаниям примкнули SAP и Siebel Systems.

В апреле 2003, BEA Systems, IBM, Microsoft, SAP и Siebel Systems перепедали BPEL4WS в консорциум OASIS для стандартизации. В OASIS спецификация официально была переименована в WS-BPEL, однако на практике на нее обычно ссылаются как на BPEL.

BPEL очень сложен для понимания. Понятия, относящиеся к предметной области, находятся в нем на одном уровне с техническими понятиями, специфическими для технологии Web-сервисов. Это сильно ухудшает читаемость языка (например, по сравнению с XPDЛ).

Несмотря на то, что в настоящее время BPEL является наиболее распространенным языком определения бизнес-процессов, в последние годы он подвергается серьезной критике. Многие эксперты считают, что, несмотря на то, что СУБПиАР хорошо совместимы с технологией Web-сервисов, идея ограничивать все взаимодействие с СУБПиАР только технологией Web-сервисов, является сомнительной. В этом случае все равно приходится функциональность, реализуемую автоматическим исполнителем задания, программировать на каком-либо языке программирования, потом «окружать» веб-сервисом, результаты работы программного кода переводить в текстовый вид (XML), пересылать по внутренней сети предприятия, потом переводить «обратно» в объекты программного кода для обработки в ядре СУБПиАР. На это тратятся совершенно неоправданные ресурсы: время программистов-разработчиков

и компьютерные ресурсы, потребляемые при переводе всех данных в текстовый вид и обратно и пересылке текстов по внутренней сети предприятия.

Кроме того, сам язык BPEL неоправданно усложнен. В нем одни и те же конфигурации можно описать совершенно разными способами (например, в стиле XLANG или в стиле WSFL). Для того, чтобы СУБПиАР поддерживала все эти особенности BPEL, разработчикам СУБПиАР, поддерживающих BPEL, приходится долго разбираться в конструкциях языка и писать очень много кода, что приводит к росту затрат на разработку СУБПиАР, сложностям в сопровождении и поддержке, а также к повышению требований к вычислительным ресурсам при эксплуатации СУБПиАР.

2.4.4 Математические основы языков определения бизнес-процессов

Для переноса описания бизнес-процесса в СУБПиАР требуется описать бизнес-процесс формально: задать схему процесса, наборы переменных, графических элементов форм, связать узлы графа процесса с соответствующими внешними приложениями или ролями пользователей и т. д.

На этом этапе встает вопрос о формальных языках определения бизнес-процессов, которые позволяют перенести их в СУБПиАР. В основе большинства языков (как стартовая точка разработки концепции языка) лежит одна из двух хорошо известных математических теорий:

- теория сетей Петри;
- концепция Пи-исчисления.

2.4.4.2 Теория сетей Петри

Эта математическая теория, основанная на классической теории графов, является расширением теории конечных автоматов. Она возникла в 60-х годах XX века и с тех пор постоянно развивается. Теория сетей Петри — сложная, очень хорошо разработанная теория. В ней строго определены такие понятия, как состояния, условия, переходы и т. п. Также теория включает графическую нотацию (систему графических обозначений, на основе которых можно рисовать соответствующие графы). Сети Петри хорошо исследованы математиками. Установлены многие их свойства, доказано большое количество теорем.

Практическое использование теории сетей Петри в основном было связано с описанием поведения очень сложных систем, например элементов интегральных схем. Построив для системы соответствующую сеть Петри, далее можно было использовать результаты соответствующих теорем и таким образом исследовать свойства системы.

Для описания СУБПиАР использовать концепцию сетей Петри в явном виде неудобно, так как графическая нотация сетей Петри не является интуитивно понятной. С ней сложно работать бизнес-аналитикам, а тем более менеджерам. Кроме того, появились некоторые классы бизнес-процессов, которые нельзя описать с ее помощью.

Наследниками теории сетей Петри стали первые языки определения бизнес-процессов (например, WPDЛ и XPDЛ коалиции WfMC). Они основаны на теории графов и концептуально включают в себя многие понятия и концепции

сетей Петри: узлы, переходы, условия и т.д. Однако, в отличие от сетей Петри, эти языки не являются строгими. В ряде случаев можно составить такие предложения языка, которые будут синтаксически допустимыми, однако поведение порожденного бизнес-процесса не будет определено однозначно.

Ученые W. M. P. van der Aalst и A. H. M. ter Hofstede предложили расширение концепции сетей Петри для СУБПиАР, введя в нее дополнительные базовые элементы. Этот подход привел к появлению нового строгого языка для СУБПиАР — YAWL (Yet Another Workflow Language, в переводе — «еще один язык потоков работ»). К сожалению, он оказался очень сложным, его графические диаграммы не являются интуитивно понятными и, скорее всего, использовать он будет исключительно в теоретических целях.

Ограниченность языков, основанных на теории сетей Петри

Эта ограниченность следствие того, что концепция сетей Петри основана на теории графов. В последнее время в программировании предложены понятия, не укладывающиеся в рамки теории графов, например такое понятие, как исключения (exceptions). Эти новые «программистские» понятия были применены при разработке некоторых языков для СУБПиАР (в частности, BPMML, BPMN) и оказались очень полезными. Таким образом, роль теории сетей Петри в мире языков для СУБПиАР неоднозначна: с одной стороны, эту теорию можно использовать для исследования бизнес-процессов некоторых видов, с другой — с ее помощью нельзя описать все бизнес-процессы. Кроме того, диаграммы сетей Петри очень громоздки. В случае сложных процессов соответствующие им сети Петри содержат огромное количество элементов и разобраться в них очень трудно.

2.4.4.3 Концепция Пи-исчисление

Концепция Пи-исчисление (Pi calculus) была разработана в конце 80-х годов XX века Робинот Милнером и основана на алгебре параллельных процессов. В отличие от сетей Петри, математическими объектами Пи-исчисления являются не графы, а выражения над элементами специальных множеств и преобразования над этими выражениями. В настоящее время Пи-исчисление является перспективной, но еще очень молодой и развивающейся теорией, в ней еще очень много открытых вопросов и нерешенных проблем.

Разработчики языков BPEL и BPML утверждают, что эти языки обладают очень высокой выразительной мощностью, так как в основе этих языков лежит серьезная математическая теория — Пи-исчисление. Однако немало и скептиков, считающих, что связь этих языков с концепцией Pi calculus неочевидна, и предполагающих, что это скорее маркетинговый ход, чем реальное использование сложной математической теории для построения языка определения бизнес-процессов для промышленных систем масштаба предприятия.

2.4.5 Выводы

У всех существующих в настоящее время стандартов, относящихся к СУБПиАР, есть серьезные проблемы. Лидирующие в этой области стандарты выглядят неоправданно сложными. Вполне возможно, что в будущем реально

используемым стандартом станет какая-то новая спецификация. Например, в разрабатываемую в настоящее время версию 2.0 графической нотации BPMN была добавлена семантика исполнения бизнес-процесса. Таким образом, после выхода версии 2.0 этого стандарта, он не только будет описывать графические элементы, использующиеся в схеме бизнес-процесса, но также сможет быть языком, описывающим исполнение бизнес-процесса, то есть станет еще одним стандартом, конкурирующим с XPDЛ и BPEL.

Стандарт XPDЛ (версия 2.1) в настоящее время позиционируется как форма хранения бизнес-процесса в виде XML-текста, в которую можно преобразовать процесс, нарисованный в графической нотации BPMN (имеется в виду действующая в настоящее время версия стандарта BPMN — 1.2). То есть появление семантики исполнения в BPMN 2.0 должно сделать таким образом позиционируемый стандарт ненужным. На стандарте XPDЛ основано более 80 популярных программных продуктов и в ближайшее время он, несомненно, будет востребован, однако будущее этого стандарта является неопределенным.

Стандарт BPMN позиционировался, в частности, как простая и доступная нотация для бизнес-аналитиков, не имеющих специального программистского образования. Однако разрабатываемую в настоящее время версию 2.0 стандарта BPMN никак нельзя назвать простой. Документ содержит более 500 страниц. В нем используются традиционно «программистские» понятия, такие как «транзакции», «исключения», «события», «сигналы», «циклы» и т.п. Вполне вероятно, что реально использовать стандарт BPMN 2.0 будут не бизнес-аналитики (он для них слишком сложен и написан в непонятных для них терминах), а ИТ-специалисты — разработчики программных систем, поддерживающих этот стандарт. Для бизнес-аналитиков тогда потребуется другой, более простой в изучении и использовании стандарт. Возможно, это будет какое-то подмножество BPMN 2.0.

То есть, за последние годы, ситуация со стандартами сильно изменилась, но еще не стала определенной. Война стандартов продолжается.

2.5 Workflow паттерны

2.5.1 Зачем нужны Workflow-паттерны

Пока еще нет универсальной общепризнанной теории, описывающей бизнес-процессы. Различными компаниями и сообществами разработано немало различных несовместимых друг с другом стандартов, таких как: BPEL, XPDЛ, WSCI, YAWL.

Ситуация с программными продуктами тоже достаточно запутанная. Современные СУБПиАР в большинстве случаев соответствуют какому-либо из известных стандартов, однако также существует большое количество СУБПиАР, которые никаким стандартам не следуют.

В этих условиях группа ученых — W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski и A.P. Barros, попыталась разработать единую методологию систематизации и классификации СУБПиАР и связанных с ними стандартов. Они проанализировали распространенные СУБПиАР и стандарты, выделили в них типичные элементы, выявили наиболее часто повторяющиеся структуры и назвали их Workflow-паттернами (Workflow patterns, или сокращенно WF-паттерны).

WF-паттерны — это характерные структуры внутри бизнес-процесса, рассматриваемые в его контексте. Видно, что определение содержит в себе субъективный элемент. Это не удивительно, т.к. для выявления WF-паттернов был применен феноменологический подход, в рамках которого набор WF-паттернов явно зависит как от исследователя, так и от набора исследуемых СУБПиАР. Тем не менее, в результате удалось выявить набор из двадцати WF-паттернов, на основе которых можно построить большую часть всех реальных бизнес-процессов.

Паттерны помогают решить две задачи:

- Сравнить предлагаемые стандарты и/или СУБПиАР. Составив таблицу, показывающую, какие паттерны в каких системах и стандартах поддерживаются, можно сравнить их друг с другом;
- Выбрать СУБПиАР для предприятия — определить множество паттернов, характерных для бизнес-процессов данного предприятия, далее сузить круг рассматриваемых систем, исключив те, в которых эти паттерны не поддерживаются.

2.5.2 Эволюция Workflow-паттернов

Изначально было предложено 20 WF-паттернов, относящихся в основном к управлению потоком. Далее количество WF-паттернов потока управления было увеличено до 43, были разработаны паттерны перспективы данных, перспективы ресурсов и паттерны, связанные с бизнес-исключениями. Также понятие WF-паттерны стали развивать другие группы ученых. Например, Ру, Маджинис и Браун² добавили шесть дополнительных паттернов коммуникации, относящихся к взаимодействию потоков управления. Появились WF-паттерны, характеризующие взаимоотношения продавцов-покупателей, WF-паттерны, связанные с транзакциями, и другие типы WF-паттернов.

Постепенно новые WF-паттерны становятся похожи на дизайн-паттерны объектно-ориентированных языков программирования. Мы считаем, что это не случайно. Языки определения бизнес-процессов можно рассматривать как высокоуровневые языки программирования, следовательно, к ним должны быть применимы многие общие концепции программирования.

2.5.3 Наиболее распространенные Workflow-паттерны

Ниже мы приведем описание наиболее распространенных WF-паттернов вместе с их графическими образами в нотации UML.

«Последовательность». (Англоязычный термин — Sequence).

Простейший «участок» бизнес-процесса: Два узла-действия соединены переходом. После того, как исполнитель выполнил действие первого узла, управление переходит ко второму.

² W. A. Ruh, F. X. Maginnis, W.J. Brown. Enterprise Application Integration: A Wiley Tekh Brief. John Wiley and Sons, Inc, 2001.



Рисунок 2.5.1. Паттерн «Последовательность»

«Параллельное расщепление». (Англоязычные термины — AND-split, Fork).

Представляет собой маршрутный узел, в который приходит только один переход и из которого исходит два или более переходов. Причем после того, как управление передано узлу, поток управления бизнес-процессом распадается на несколько потоков, которые выполняются параллельно. Для каждого исходящего перехода должен существовать «свой» поток управления.

Данный паттерн часто используется в связке с паттерном «синхронизация».

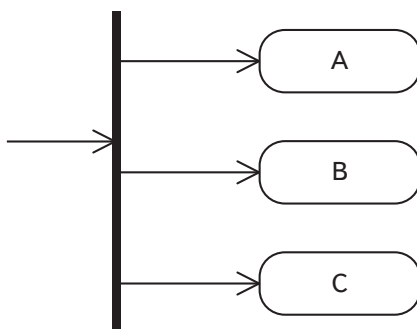


Рисунок 2.5.2. Пример паттерна «Параллельное расщепление»

«Синхронизация». (Англоязычные термины — AND-join, Join).

Маршрутный узел, в котором соединяются два или более перехода, а выходит только один. Управление не будет передано дальше, пока все потоки управления бизнес-процессом (по количеству входящих переходов) не достигнут данного узла. После того, как в узел придут все потоки управления, будет инициирован только один поток, соответствующий исходящему переходу.

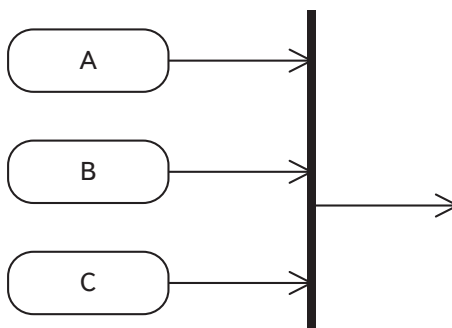


Рисунок 2.5.3. Пример паттерна «Синхронизация».

«Исключающий выбор». (Англоязычные термины — XOR-split, Decision).

Маршрутный узел в графе бизнес-процесса, в который приходит только один и из которого исходит два или более переходов. Причем, после того, как управление перешло к данному узлу, в нем делается выбор, по какому из исходящих переходов оно будет передано далее.

Паттерн часто используется вместе с паттерном «простое соединение».

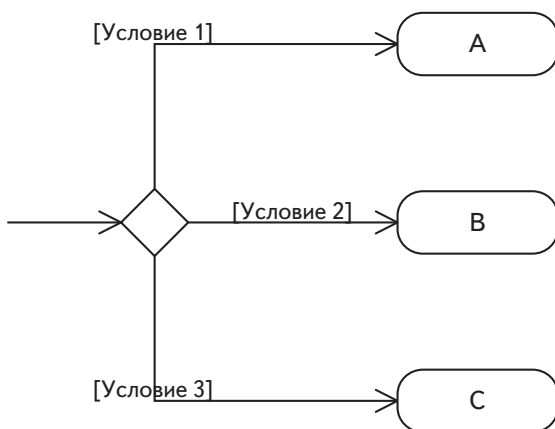


Рисунок 2.5.4. Пример паттерна «Исключающий выбор»

«Простое соединение». (Англоязычные термины — XOR-join, Merge).

Маршрутный узел в графе бизнес-процесса, в котором соединяются два или более перехода, а выходит только один. После того, как в узел пришло управление от любого из входящих потоков, поток управления передается на единственный исходящий переход.

Предполагается, что управление может прийти в узел только по одному из входящих переходов.

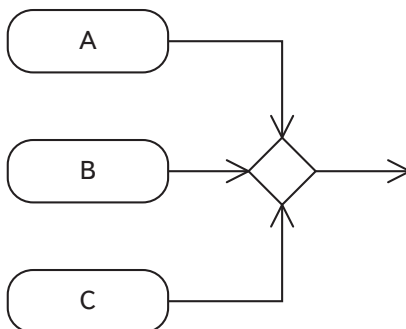


Рисунок 2.5.5. Пример паттерна «Простое соединение»

Пять вышеописанных паттернов поддерживаются всеми известными стандартами для СУБПиАР и являются основой описания абсолютного большинства существующих бизнес-процессов. Однако в некоторых достаточно редких бизнес-процессах используются и паттерны, включенные в группу «дополнительных». Эти паттерны либо нельзя свести к комбинации и суперпозиции основных, либо при таком сведении получаются крайне громоздкие, неудобные для восприятия схемы. Для реализации подобных паттернов нужны дополнительные элементы языков определения бизнес-процессов и, соответственно, дополнительная функциональность СУБПиАР.

Приведем несколько дополнительных паттернов:

«Дискриминатор». (Англоязычный термин — Discriminator).

Узел в графе бизнес-процесса, в котором соединяются два или более перехода, а выходит только один переход. Как только узлу передается управление (по любому из входящих переходов), сразу активизируется исходящий переход, то есть начинает выполняться следующее действие бизнес-процесса. Выполнение других активных потоков не прерывается, однако передача управления каждому из этих потоков в узел дискриминатора игнорируется и пришедший поток завершает свое существование.

Данный паттерн обычно используется вместе с паттерном «Параллельное расщепление».

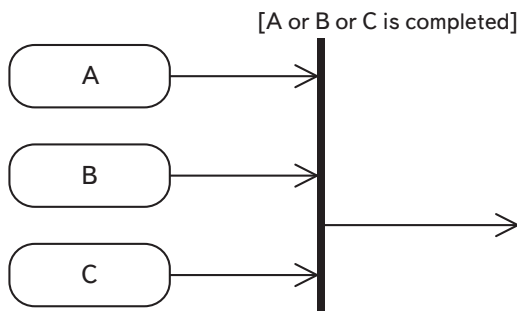


Рисунок 2.5.6. Пример паттерна «Дискриминатор».

Пример. Бизнес-процесс «Подбор ключа криптозащиты». Для проверки стойкости криптозащиты часто применяются параллельные вычисления, построенные по следующему принципу: каждому участнику выдается пакет исходных данных, причем (при большом количестве участников) эти пакеты не велики и обсчитываются за несколько дней. Как только один из участников находит ключ, он должен его сразу прислать в исследовательский центр. Результат работы остальных участников может быть проигнорирован.

«Произвольный цикл». (Англоязычный термин — Arbitrary cycle).

Набор узлов в графе бизнес-процесса, в котором один или несколько узлов могут проходиться многократно, то есть в эти узлы много раз может приходиться (и, соответственно уходить) управление. Причем, в отличие от регулярных циклов, точки «входов» в набор повторяющихся узлов (или «выходов» из них) могут быть различными.

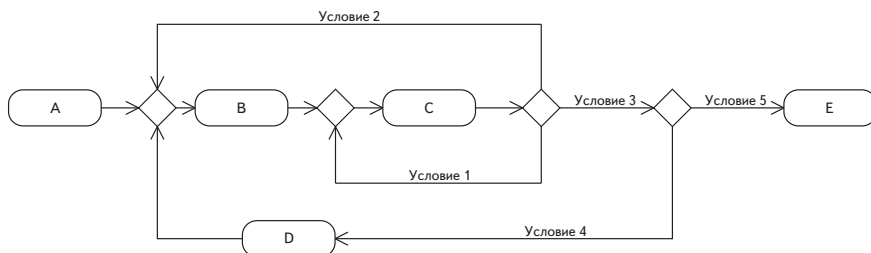


Рисунок 2.5.7. Пример произвольного цикла

«Отложенный выбор». (Англоязычный термин — Deferred choice).

Узел в графе бизнес-процесса, в который приходит только один и из которого исходит два или более переходов. После прохождения узла поток управления бизнес-процессом распадается на несколько потоков, число которых равно количеству исходящих переходов. Далее все потоки, соответствующие исходящим переходам становятся активными. Однако после того, как первый узел для исполнения выбран, активным остается только поток этого узла, а все остальные принудительно завершаются.

Паттерн часто используется вместе с паттерном «Простое соединение».

Пример: По электронной почте приходит письмо от клиента с запросом на поставку товара. Это письмо отправляется нескольким менеджерам отдела продаж. Тот менеджер, который первым подтвердит желание работать с данным заказом, и будет его исполнять, у остальных менеджеров запрос будет отозван.

2.5.4 Применение паттернов для сравнения языков определения бизнес-процессов и графических нотаций

Авторы данной концепции произвели классификацию наиболее известных языков относительно возможности «прямой» (в оригинале «direct») поддержки WF-паттернов. Однако произведенная ими классификация весьма субъективна и сильно зависит от определения понятия «прямая поддержка». Во многих случаях, когда по их мнению тот или иной стандарт не обеспечивает «прямой» поддержки данного паттерна, на самом деле существует синтаксически допустимая конструкция языка, реализующая паттерн, но кажущаяся им слишком громоздкой (что в ряде случаев является спорным). В таблице приведены опубликованные авторами концепции результаты сравнения трех стандартов (в таблицу включены только представленные в настоящей книге WF-паттерны).

Таблица 2.5.1

Паттерн	WF-языки		
	XPDL	BPEL	BPEL
Последовательность	+	+	+
Параллельное расщепление	+	+	+
Синхронизация	+	+	+
Исключающий выбор	+	+	+
Простое соединение	+	+	+
Дискриминатор	-	-	-
Произвольный цикл	+	-	-
Отложенный выбор	-	+	+

Из таблицы 2.5.1 следует, что ни один из рассматриваемых языков не поддерживает всех паттернов, и из них нельзя выбрать «наилучший» в этом смысле язык. Для каждого языка существует паттерн, который данным языком не поддерживается, но поддерживается, по крайней мере, одним из оставшихся языков.

Заключение

В условиях «войны стандартов» WF-паттерны, несмотря на некоторые недостатки, являются хорошим средством сравнения и анализа как различных связанных с СУБПиАР стандартов, так и конкретных СУБПиАР.

В силу несовершенства существующих стандартов как разработчикам СУБПиАР, так и организациям, выбирающим СУБПиАР для внедрения, опасна жесткая привязка к какому-то одному стандарту. Возможно, в будущем этот стандарт будет кардинально переработан, может быть, даже полностью вытеснен новым, более удобным и принципиально другим стандартом. Таким образом, имеет смысл выбирать гибкую, легко изменяемую СУБПиАР, допускающую реализацию импорта-экспорта в различные языки, а также адаптацию к стандартам, которые могут появиться в будущем.

В этих условиях СУБПиАР с открытым кодом обладают преимуществами по отношению к закрытым проприетарным системам.

2.6 Языки описания бизнес-процессов

СУБПиАР реализуют процессный подход к управлению предприятием. Для описания автоматизируемых бизнес-процессов (административных регламентов) в этих системах применяются специальные языки описания бизнес-процессов.

Задача языка — описать бизнес-процесс (регламент) формально: задать его возможные состояния, в которых определены соответствующие действия, определить набор внутренних переменных, бизнес правила, графические элементы форм, связать узлы-действия бизнес-процесса с соответствующими внешними приложениями или ролями пользователей и т.д. Процесс, описанный на каком-либо стандартном языке можно переносить в другие СУБПиАР, поддерживающие этот стандарт.

В настоящее время не существует единого стандартного языка. Существует более десяти несовместимых друг с другом стандартов, относящихся к управлению бизнес-процессами.

Наиболее известными языками описания бизнес-процессов являются следующие языки:

- XPDЛ
- BPEL
- BPML

В последнее время коммерческие проприетарные системы скорее тяготеют к языку BPEL, а системы с открытым кодом часто реализуют язык XPDЛ (а также другие стандарты коалиции WfMC). Некоторые системы реализуют экспорт определений бизнес-процессов при помощи нескольких спецификаций.

В настоящем разделе дается краткое описание наиболее известных языков определения бизнес-процессов. Также производится сравнение языков. При помощи всех языков реализуются некоторые характерные WF-паттерны, анализируются достоинства и недостатки реализаций.

В разделе также рассматривается язык jPDL проекта JBOSS jBPM в качестве примера нестандартного языка, ориентированный на поддержку наиболее используемых WF-паттернов.

2.6.1 Краткое описание основных конструкций языка XPDЛ

В основе языка XPDЛ лежит математическое понятие: ориентированный граф. Граф представляет собой набор узлов, некоторые из которых соединены переходами. Изменение состояния бизнес-процесса соответствует переходу точки управления из одного узла графа в другой.

Основные элементы языка:

- Activity (Узел-действие)
- Transition (Переход)
- Participant (Участник бизнес-процесса)
- Application (Внешнее приложение)
- DataType (Тип переменной)
- DataField (Переменная)

В версии 2.1 языка XPDЛ дополнительно введены конструкции, позволяющие записывать элементы графической нотации BPMN в виде текстовых конструкций языка XPDЛ, а также введены конструкции, соответствующие некоторым элементам нотации BPMN.

Описание использования элементов языка

Элементы, определяющие выполняемые действия и порядок их выполнения

Граф бизнес-процесса определяется наборами элементов Activity и Transition. Activity — основной элемент бизнес процесса. Элементы Activity соединяются при помощи элементов Transition. Существует три типа элементов Activity:

- Route
- Implementation
- BlockActivity

Route — узлы, не выполняющие действий, используются в целях маршрутизации точек управления.

Implementation — узлы, с которыми связаны действия в бизнес-процессе. Существует три варианта Implementation:

- Узел взаимодействия с пользователем
- Узел взаимодействия с Внешним приложением
- Узел, запускающий подпроцесс.

BlockActivity — узел-контейнер. Содержит в себе не имеющую разветвленности последовательность узлов.

Элемент Transition используется для описания переходов между элементами Activity. Каждый элемент Transition содержит информацию, из какой Activity в какую Activity и при каких условиях происходит переход (бывают условные и безусловные переходы).

Элементы, описывающие данные бизнес-процесса

В начале описания бизнес-процесса находятся описания типов (тег TypeDeclaration). Для описания данных, относящихся к процессу и параметров, передаваемых и возвращаемых приложениями, используются элементы DataField и DataType. Также в XPDЛ существует понятие ExtendedAttribute. Оно дает возможность расширять язык путем ввода дополнительных типов переменных.

Элементы, описывающие исполнителей заданий

Для описания участников бизнес процесса, т.е. сущностей, которые могут выполнять работу, используется элемент Participant. Существует 6 подтипов элемента Participant:

- Role — соответствует роли участника бизнес-процесса. Для каждой роли должен существовать список людей, которые могут быть «назначены» на эту роль;
- OrganisationUnit — соответствует административному подразделению организации;
- Human — конкретный человек, который будет взаимодействовать с бизнес-процессом при помощи графического интерфейса;
- System — конкретное автоматическое приложение;
- Resource — некоторый ресурс, который может «выдать» исполнителя работы;
- ResourceSet — набор ресурсов.

Взаимодействие с внешними приложениями

В языке XPDЛ описываются спецификации внешних приложений. Фактически — это описание функций. Описываются названия функций и их параметры (при помощи тега Application). Внутри Activity конкретное приложение указывается в виде параметра тега Tool, внутри этого тега также производится отображение формальных параметров на фактические параметры.

Структура бизнес-процесса

Упрощено описание бизнес-процесса в XPDЛ выглядит следующим образом:

1. Описывается множество переменных бизнес-процесса и их типы;
2. Описывается множество участников бизнес-процесса;
3. Описывается множество внешних приложений, вызываемых бизнес-процессом — имена функций и типы их параметров;
4. Описывается множество всех узлов графа бизнес-процесса. Для узла задается исполнитель, фактические параметры, набор исходящих переходов и другие данные;
5. Описывается множество всех переходов. Для каждого перехода определяется, какие узлы он связывает и, если требуется, условие, при котором по данной связи осуществляется переход управления;

Переход может соединять любые два узла. То есть бизнес-процессу может соответствовать граф любой сложности и топологии. В частности, в графе бизнес-процесса допустимы циклы (WF-паттерн «произвольный цикл»). Наличие поддержки «произвольных циклов» в языке описания бизнес-процессов можно поставить в соответствие аналогии с обычными языками программирования с использованием оператора «goto».

Однако, в силу того, что переход управления осуществляется только по ребрам графа, в XPDЛ нельзя реализовать WF-паттерн «отложенный выбор».

Работа с бизнес-процессом

Технология работы с определениями и экземплярами бизнес-процессов, записанных на языке XPDЛ, определяется другими спецификациями коалиций WFMC и OMG:

- OMG. Workflow Management Facility Specification
- WfMC. WAPI (Workflow Application Programming Interface)

В этих спецификациях описывается общая архитектура СУБПиАР, интерфейсы взаимодействия различных компонентов друг с другом. В частности спецификации определяют интерфейсы взаимодействия клиентского приложения и внешней системы с ядром СУБПиАР, в которое загружен бизнес-процесс. Эти интерфейсы содержат такие команды, как: запустить процесс, посмотреть состояние процесса и т.д. Основные команды, относящиеся к работе с экземпляром бизнес-процесса, следующие:

- Сгенерировать список текущих заданий;
- Сообщить ядру, что данное задание выполнено.

То есть предполагается, что ядро системы, реализующей XPDЛ, полностью пассивно. Оно только отвечает на действия взаимодействующих с ним субъектов.

2.6.2 Краткое описание языка BPML

В настоящее время стандарт BPML практически вытеснен стандартом BPEL, однако BPML представляет интерес как предшественник BPEL, из которого были взяты многие принципы и конструкции.

BPML — язык, основанный на XML и ориентированный на web-сервисы. В отличие от XPDL, BPML принадлежит к так называемым структурно-ориентированным языкам. Бизнес-процесс в BPML соответствует не математическому графу, а иерархическому набору вложенных и последовательных тегов.

Основные элементы, при помощи которых определяется бизнес-процесс в BPML:

- Activity (Узел — действие)
- Context (Контекст)
- Property (Свойство)
- Signal (Сигнал)
- Exception (Исключение)

Описание использования элементов языка

Элементы, определяющие выполняемые действия и порядок их выполнения

Activity — основной элемент бизнес-процесса. Элементы Activity могут соединяться последовательно или быть вложены один в другой. Соответственно Activity могут быть простыми (без вложенных Activity) или сложными. В описании языка написано, что бизнес-процесс является специальным типом сложной Activity.

Всего существует 17 типов простых Activity. Основной тип простой Activity называется Action. Когда управление бизнес-процессом попадает в Activity этого типа, происходит вызов описанных там web-сервисов. Есть типы Activity, соответствующие ветвлению процесса (ветвление относится только к содержащимся внутри них Activities) — это Switch и All Activities. Также существуют типы Activities, которые запускают дочерние процессы (как с ожиданием их окончания, так и без), организуют задержки выполнения процесса, и т.д. Существуют несколько типов Activities, которые реализуют разного вида циклы. Для синхронизации точек управления, находящихся в Activities одного уровня вложенности, в языке используются сигналы (Signals).

В BPML введены понятия «Исключение» (Exception) и «Компенсация» (Compensation). «Исключение» соответствует возникновению внештатной ситуации, при которой оказывается, что выполнять некоторый участок бизнес-процесса уже не требуется. Например, во время выполнения бизнес-процесса оформления туристической поездки клиент позвонил в туристическую компанию и сообщил, что он от туристической поездки отказывается. «Компенсация» соответствует необходимым действиям по корректному завершению ситуации, возникшей в связи с «Исключением». Если в вышеприведенном примере для клиента были забронированы билеты на самолет и номер в гостинице, то задачей «Компенсации» будет отменить бронирование.

Элементы, описывающие данные бизнес-процесса

На языке BPML данные описываются в рамках контекста (Context). Контекст содержит относящиеся к процессу переменные, локальные определения процессов, сигналов и т.д., служит для передачи информации между узлами и синхронизации. Переменные описываются тегом Property. Переменные могут быть локальными или глобальными по отношению к данному контексту.

Описание исполнителей заданий

В спецификации исполнители практически не описываются. Все это «перенесено» на технологию web-сервисов.

Взаимодействие с внешними приложениями

В языке BPML синтаксис конструкций для взаимодействия бизнес-процесса с внешними приложениями не описан. Используется технология web-сервисов.

Сравнение BPML с XPDL

Язык BPML существенно «легче» языка XPDL. Это происходит за счет следующего:

- В языке не надо описывать внешние приложения (Applications в XPDL). Эти функции «перекладываются» на технологию web-сервисов.
- Не надо описывать и «присоединять к Activities» переходы. Архитектура связей определяется вложенностью тегов, соответствующих элементам Activity. В BPML нет понятия Transition.
- Не надо в рамках языка специально определять описание участника бизнес-процесса. Все участники — это web-сервисы, следовательно, их описания соответствуют спецификации web-сервисов.

Кроме того, для работы с бизнес-процессами, написанными на XPDL, требуются дополнительные спецификации. В частности они описывают, каким образом можно «сообщить» определенной Activity, что она «выполнена» и управление может «двигаться дальше». В соответствии с идеологией языка XPDL среда, в которой выполняется бизнес-процесс, не является активной, активными должны быть внешние участники, а среда выполнения процесса только реагирует на их действия.

«Идеология» языка BPML скорее противоположна. В этой парадигме бизнес-процесс может быть активным и давать задания своим участникам-исполнителям. Исполнители, являясь web-сервисами, могут «ничего не знать» о бизнес-процессе, в котором они участвуют.

Однако, отказ от понятия Переход (Transition) и замена его вложенностью тегов приводят к тому, что граф, соответствующий бизнес-процессу, в BPML не может быть графом произвольной структуры, например, он не может содержать сложные циклы. Вследствие этого в BPML невозможно реализовать WF-паттерн «Произвольный цикл». Для того, чтобы выполнять повторяющиеся последовательности шагов, в BPML введено несколько типов Activity-циклов, однако в этом случае циклически повторяться может только содержащаяся внутри данной Activity последовательность узлов. Отсутствию произвольных циклов в бизнес-процессе можно поставить в соответствие аналогию программирования «без goto» в обычных языках программирования.

Замечание: В BPML параллельно выполняющиеся ветви процесса могут еще дополнительно обмениваться сигналами, производя, таким образом, синхронизацию. Трудно сказать — облегчает это понимание бизнес-процесса или наоборот затрудняет.

В силу того, что язык BPML основан на тегах, в нем легко организовать генерацию и обработку исключений. Также, т.к. BPML не основан явно на теории графов, он позволяет реализовать WF-паттерн «отложенный выбор». В BPML реализованы такие концепции, как транзакции и расписания. В BPML можно устанавливать задержки и сроки. В XPDЛ тоже появились такие понятия, однако появились они заметно позже, чем в BPML, а функциональность их была ниже функциональности соответствующих конструкций BPML.

2.6.3 Краткое описание языка BPEL

Язык BPEL, также как и BPML, ориентирован на web-сервисы. Во многом этот язык похож на BPML, однако, существенно сложнее. BPEL появился путем слияния языков WSFL и XLANG. Эти языки основаны на разных моделях: WSFL основан на теории графов, XLANG — на иерархии тегов XML.

BPEL унаследовал конструкции обоих языков. Например, он допускает реализацию некоторых WF-паттернов в двух вариантах: в стиле WSFL и в стиле XLANG. В некоторых случаях допустим и смешанный стиль. Это делает BPEL трудным для изучения. Несмотря на то, что BPEL является наследником языков различной природы, его можно отнести к классу структурно-ориентированных языков. Унаследованные «граф-ориентированные» конструкции реально соответствуют некоторым ограничениям на порядок выполнения узлов-действий внутри параллельного блока.

Описание использования элементов языка

BPEL определяет два вида процессов: абстрактный процесс и исполняемый процесс. Абстрактный процесс определяет протокол обмена сообщениями между различными участниками, не «открывая» алгоритмы «внутреннего» поведения участников процесса. В отличие от абстрактного процесса, исполняемый процесс содержит в себе алгоритмы, определяющие порядок выполнения Activities (узлов процесса), назначение исполнителей, обмен сообщениями, правила обработки исключений и т.д.

Activities в BPEL делятся на примитивные и структурные.

Список примитивных Activities:

- Receive — ожидает сообщения внешнего источника;
- Reply — отвечает внешнему источнику;
- Invoke — запускает операцию какого-либо web-сервиса;
- Wait — ждет определенный период времени;
- Assign — копирует значение одной переменной в другую;
- Throw — бросает исключение в случае ошибки;
- Terminate — принудительно завершает выполнение службы;
- Empty — не выполняет никаких действий.

Список структурных Activities:

- Sequence — соответствует последовательному выполнению содержащихся внутри узлов;
- Switch — условная передача управления (соответствует оператору Switch языков программирования C++, Java и т. д.);
- While — организует цикл типа «While»;
- Pick — запускает обработку событий и исключительных ситуаций;
- Flow — соответствует параллельному выполнению узлов, содержащихся внутри его;
- Scope — группирует узлы для обработки одним fault-handler'ом.

Также в языке присутствует понятие «связь» (link). Эта конструкция унаследована из граф-ориентированного предка BPEL. Как правило, применяется она к Activities, находящимся внутри параллельного блока, и накладывает ограничения на порядок их выполнения. На конструкции языка, использующие link, накладываются ограничения, например, Activities, соединенные при помощи этого элемента, не могут образовывать циклов.

Описание данных

Переменные описываются при помощи тега variables.

Описание исполнителей заданий

Для описания исполнителя используется тег partnerLink, в языке активно используется технология web-сервисов.

Взаимодействие с внешними приложениями

Для описания протоколов общения с «внешним миром» в BPEL используется технология web-сервисов. В некоторые теги добавлен параметр variable.

Расширение языка для взаимодействия с людьми

В июне 2007 Active Endpoints, Adobe, BEA, IBM, Oracle и SAP опубликовали спецификации BPEL4People и WS-HumanTask, в которых описывается, как может быть реализовано в BPEL взаимодействие с людьми.

Сравнение языков BPEL и BPML

Идеологически языки BPEL и BPML очень похожи. Нам кажется, что BPML проще и удобнее BPEL, однако за BPEL стоят корпорации-гиганты: IBM, Microsoft, BEA, SAP и Siebel. Именно благодаря маркетинговой мощи этих компаний в настоящее время BPML практически вытеснен языком BPEL.

2.6.4 jPDL — «нестандартный» язык, ориентированный на поддержку основных WF-паттернов

Некоторое время назад появился еще один подход к построению языков описания бизнес-процессов. Он заключается в следующем:

Так как

1. В настоящее время не существует единого стандартного языка для СУБ-ПиАР — между международными коалициями идет «война» стандартов и еще непонятно, какой стандарт окажется «победителем».

2. Многие стандарты, относящиеся к области управления бизнес-процессами, выглядят неоправданно сложными.

При разработке СУБПиАР имеет смысл не реализовывать полностью какую-либо из вышеописанных спецификаций, а использовать упрощенные языки описания бизнес-процессов, поддерживающие основные WF-паттерны. В этом случае сами СУБПиАР будут значительно проще и надежнее, а поддержка основных WF-паттернов гарантирует некоторый базовый уровень функциональности системы.

Примером такого языка является язык jPDL проекта JBOSS jBPM. jPDL относится к классу граф-ориентированных языков и разработан «в духе» идеологии WfMC, фактически он соответствует упрощенному варианту спецификаций этой коалиции.

Основным упрощением jPDL является то, что во многих случаях вместо сложных конструкций XPDL этот язык напрямую использует классы и конструкции языка программирования Java.

Описание использования элементов языка

Основные элементы языка jPDL:

- task-node (Узел-действие);
- process-state (Узел-подпроцесс);
- decision (Исключающий выбор);
- fork (Параллельное расщепление);
- join (Синхронизация);
- swimlane (Роль-дорожка);
- variable (Переменная);
- transition (Переход).

В соответствии с определением языка jPDL бизнес-процесс определяется файлом-архивом, который содержит несколько XML-описаний, в которых задаются узлы графа бизнес-процесса, переходы между узлами, роли-дорожки участников бизнес-процесса, переменные бизнес-процесса. Также архив содержит описания HTML-форм, с которыми будет работать пользователь в соответствующих узлах бизнес-процесса, java-классы, специфические именно для данного бизнес-процесса, которые должны быть подгружены в ядро системы при загрузке бизнес-процесса а также дополнительную информацию, например, — визуальное представление графа бизнес-процесса, на котором пользователю будет показано текущее положение точек управления.

Описание данных

Переменные описываются при помощи тега variable. Фактический тип переменной должен соответствовать некоторому Java-классу.

Описание исполнителей

Исполнители описываются при помощи тегов swimlane и assignment. Для указания инициализатора можно вставлять в тег swimlane соответствующую ссылку на Java-класс.

Выводы по jPDL

Оказалось, что отказ от следования стандартам международных коалиций и ориентация на язык программирования Java принесли языку много преимуществ:

- Язык оказался простым, но достаточно мощным;
- Ядро JBOSS jBPM, интерпретирующее jPDL, оказалось простым и понятным для большого количества Java-программистов, что привело к росту популярности этого ядра и включению его в линейку продуктов JBOSS;
- Ориентация на Java привела к возможности организации разделения труда между программистом и менеджером при разработке бизнес-процессов:
 - Программист реализует некоторые часто используемые компоненты;
 - Менеджер разрабатывает бизнес-процесс, используя разработанные программистом компоненты, не вдаваясь в механизм их работы.

То есть, опыт проекта JBOSS jBPM позволяет предположить, что в настоящих условиях решения, ориентированные не на поддержку стандартов международных коалиций, а на удовлетворение некоторого класса требований к функциональности (например, поддержки основных WF-паттернов) могут быть успешными.

2.6.5 Сравнение языков при помощи Workflow-паттернов

Применение WF-паттернов «исключающий выбор» и «простое соединение» для сравнения WF-языков

В данном разделе основные конструкции языков будут продемонстрированы на примере реализации (в несколько упрощенной форме) WF-паттернов «исключающий выбор» и «простое соединение». Примеры показывают, что при реализации WF-паттернов на различных языках конкретный синтаксис выражений отличается, но основные конструкции языка можно отнести к одному из двух классов (либо к граф-ориентированному классу, либо к структурно-ориентированному).

Во всех примерах данного раздела рассматривается один и тот же тестовый бизнес-процесс (граф которого изображен на рисунке 2.6.1.), соответствующий связке паттернов «исключающий выбор» и «простое соединение»: Управление находится в узле «А», далее управление переходит либо в узел «В», либо в узел «С» в зависимости от выполнения некоторого условия, далее из любого из этих узлов управление переходит в узел «D».

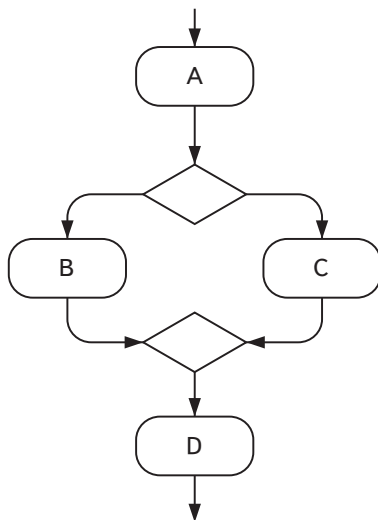


Рисунок 2.6.1 Граф бизнес-процесса с паттернами «исключающий выбор» и «простое соединение»

Реализация паттернов «исключающий выбор» и «простое соединение» на языке XPDЛ

```
<WorkflowProcessId="1" Name="PatternsExclusiveChoiceSimpleMerge">
  ...
  <Activities>
    <Activity Id="1" Name="A">
      ...
      <Transition Restrictions>
        <Transition Restriction>
          <Split Type="XOR">
            <TransitionRefs>
              <TransitionRef Id="1" />
              <TransitionRef Id="2" />
            </TransitionRefs>
          </Split>
        </Transition Restriction>
      </Transition Restrictions>
    </Activity>
    <Activity Id="2" Name="B">
      ...
    </Activity>
    <Activity Id="3" Name="C">
      ...
    </Activity>
    <Activity Id="4" Name="D">
      ...
      <Transition Restrictions>
        <Transition Restriction>
          <Join Type="XOR" />
        </Transition Restriction>
      </Transition Restrictions>
    </Activity>
  </Activities>
  <Transitions>
    <Transition Id="1" From="1" To="2" Name="FromAtoB">
      <Condition> ... </Condition>
    </Transition>
    <Transition Id="2" From="1" To="3" Name="FromAtoC">
      <Condition> ... </Condition>
    </Transition>
    <Transition Id="3" From="2" To="4" Name="FromBtoD" />
    <Transition Id="4" From="3" To="4" Name="FromCtoD" />
  </Transitions>
</WorkflowProcess>
```

В данном примере мы видим, что в первой части описания процесса последовательно перечисляются узлы, а во второй части — переходы между узлами. Каждый переход соответствует упорядоченной паре: ссылка на исходящий узел, ссылка на входящий узел. Понятно, что таким образом можно описать математический граф любой сложности. С другой стороны, все описания узлов в XPDL «равноправны», описания переходов определяются независимо друг от друга. В языке можно задать переход между любыми двумя узлами. В этом языке сложно будет определить системы вложенных областей, в которые можно входить только через определенные узлы (нельзя входить через «боковые стороны»). То есть такие конструкции как бизнес-исключения (exceptions) и компенсации (compensations), удобные для многих реальных бизнес-процессов, в этом языке ввести сложно.

В данном примере мы также видим, что разветвления (Split) и слияния (Join) определяются внутри описаний узлов как ограничения на входящие и исходящие переходы, а условия выбора конкретного перехода определяется в самом переходе.

Реализация паттернов «исключающий выбор» и «простое соединение» на языке BPML:

```
<process name="PatternsExclusiveChoiceSimpleMerge">
  <sequence>
    <action name="A" ...>
      ...
    </action>
    <switch>
      <case>
        <condition> ... </condition>
        <action name="B" ...>
          ...
        </action>
      </case>
      <case>
        <condition> ... </condition>
        <action name="C" ...>
          ...
        </action>
      </case>
    </switch>
    <action name="D" ...>
      ...
    </action>
  </sequence>
</process>
```

Данный вариант записи бизнес-процесса существенно короче и легче для понимания, чем вариант на языке XPDЛ. В значительной степени это следствие того, что в BPMЛ вообще не употребляется конструкция «переход между узлами». Если нет специальных управляющих тегов, то узлы выполняются последовательно в порядке их расположения в описании процесса.

Тег switch определяет выбор для выполнения единственного из перечисленных в нем узлов (по аналогии с языками программирования, в зависимости от выполнения условия тега condition). В языке BPMЛ также существует тег, определяющий одновременный порядок выполнения всех находящихся в нем узлов (тег All). Комбинирование этих тегов позволяет создавать достаточно мощные конструкции из вложенных параллельных и последовательных элементов разных типов.

Однако, как известно из математики, описать таким образом граф произвольной структуры нельзя. В частности, нельзя таким образом задать нерегулярные циклы — такие циклы, входить и выходить в которые можно через «боковые стороны». Это достаточно серьезное ограничение всех структурно-ориентированных WF-языков.

С другой стороны, конструкция из вложенных и последовательных тегов хорошо подходит для определения в них таких элементов, как exceptions (исключения) и compensations (компенсации).

Реализация паттернов «исключающий выбор» и «простое соединение» на языке BPEL:

```
<process name="PatternsExclusiveChoiceSimpleMerge"...>
  ...
  <sequence>
    <receive ... operation="operation A" ...>
      ...
    < /receive >
    <if...>
      <condition>
        ...
      </condition>
      <receive ... operation="operation B" ...>
        ...
      < /receive >
    <else>
      <receive ... operation="operation C" ...>
        ...
      < /receive >
    </else>
  </if>
  < receive ... operation="operation D" ...>
    ...
  < /receive >
</sequence>
</process>
```

Мы видим, что данная реализация примера паттернов «исключающий выбор» — «простое соединение» очень похожа на реализацию на языке BPML. BPEL, однако, допускает второй вариант реализации примера — с использованием конструкции link:

```
<process name="PatternsExclusiveChoiceSimpleMergeWithLink"...>
  ...
  <flow>
    <links>
      <link name="FromAtoB">
      <link name="FromAtoC">
      <link name="FromBtoD">
      <link name="FromCtoD">
    </links>

    <receive ... operation="operation A" ...>
      ...
      <source linkName="FromAtoB transitionCondition="..." />
      <source linkName="FromAtoC transitionCondition="..." />
      ...
    < /receive >

    <receive ... operation="operation B" ...>
      ...
      <target linkName="FromAtoB/>
      <source linkName="FromBtoD/>
      ...
    < /receive >

    <receive ... operation="operation C" ...>
      ...
      <target linkName="FromAtoC/>
      <source linkName="FromCtoD/>
      ...
    < /receive >

    < receive ... operation="operation D" ...>
      ...
      <target linkName="FromBtoD/>
      <target linkName="FromCtoD/>
      ...
    < /receive >
  </flow>
</process>
```

В этом примере все узлы бизнес-процесса помещены внутрь тега параллельного выполнения «flow», однако часть узлов связана отношениями:

- Выполнить перед — соответствует тегу «source»;
- Выполнить после — соответствует тегу «target».

Видно, что второй вариант реализации длиннее и менее понятен, но иногда использование конструкции link имеет преимущества.

Замечание: В примере реализации на языке BPML использован тег action, который подразумевает «активность» самого бизнес-процесса, т.е. бизнес-процесс сам «дает задания» web-сервисам исполнителям и ждет результатов их работы. В примере реализации на языке BPEL использован «пассивный» тег receive. После того, как управление попадает в этот тег, бизнес-процесс останавливается и, не выполняя никаких активных действий, ждет, пока определенный web-сервис пришлет ему сообщение. Данный тег был использован в примере для того, чтобы еще раз подчеркнуть, что языки BPML и BPEL допускают как «активное», так и «пассивное» поведение узлов, в отличие от XPDL, в котором все узлы только «пассивные». Для «активного» поведения узла в BPEL надо использовать тег Invoke.

Реализация паттернов «исключающий выбор» и «простое соединение» на языке jPDL:

```
<process-definition name="PatternsExclusiveChoiceSimpleMerge"...>
  ...
  <task-node name="A">
    <task name="A"/>
    <transition ... to="Decision"/>
  </task-node>

  <decision name="Decision">
    <handler...>
      if ( ... ) {
        return "approve";
      }
      return "disapprove";
    </handler>
    <transition name="approve" to="B"/>
    <transition name="disapprove" to="C"/>
  </decision>

  <task-node name="B">
    <task name="B" ... />
    <transition ... to="D"/>
  </task-node>

  <task-node name="C">
    <task name="C" ... />
    <transition ... to="D"/>
  </task-node>

  <task-node name="D">
    <task name="D"/>
    <transition ... to="..." />
  </task-node>
  ...
</process-definition>
```

Конструкции языка jPDL, связанные с переходами, в отличие от XPDL, описываются не как отдельные сущности, «равноправные» с узлами графа бизнес-процесса, а только как вложенные элементы относительно узлов графа. Вследствие этого эти конструкции существенно «легче» аналогичных конструкций языка XPDL. В целом язык jPDL значительно проще XPDL.

Заключение.

В настоящем разделе на примерах реализаций некоторых WF-паттернов показаны базовые конструкции нескольких языков, относящиеся к перспективе потока управления. Все рассмотренные языки определения бизнес-процессов «распадаются» на два класса принципиально различных языков (граф-ориентированные языки и структурно-ориентированные языки). У обоих этих классов языков есть серьезные проблемы, например, при помощи граф-ориентированных языков сложно определить такие элементы, как бизнес-исключения, а структурно-ориентированные языки плохо применимы при описании бизнес-процессов, соответствующих графам сложной структуры.

В силу несовершенства всех существующих стандартов, как разработчикам программного обеспечения, так и организациям, выбирающим СУБПиАР, опасно «жестко» привязываться к какому-то одному из существующих стандартов. Велика вероятность, что в будущем этот стандарт будет кардинально переработан, может быть, даже, все современные языки будут вытеснены новым, более удобным, принципиально другим языком.

Таким образом, имеет смысл выбирать СУБПиАР не «жестко» привязанную, к какой-либо спецификации, а гибкую, допускающую импорт-экспорт в различные языки, у которой будет возможность настройки на новые, еще не существующие языки.

2.6.6 Графические нотации.

Графические нотации для описания бизнес-процессов.

На схеме бизнес-процесса узлы процесса можно изображать по-разному. Способ изображения узлов и переходов важен, потому что от этого зависит легкость (или сложность) понимания бизнес-процесса людьми.

Согласованные наборы графических элементов, из которых строятся схемы бизнес-процессов, называются графическими нотациями изображения бизнес-процессов.

Наиболее известными графическими нотациями изображения бизнес-процессов являются:

- UML Activity Diagram (далее UML AD)
- BPMN

В работе Stephen A. White Process Modeling Notations and Workflow Patterns³, посвященной сравнению выразительной мощи UML AD и BPMN нотаций, основанной на реализациях с помощью этих нотаций типичных WF-паттернов содержится вывод, что выразительная мощь основных конструкций обеих нотаций примерно одинакова. Позже этот результат был подтвержден в более полном исследовании Lauri Eloranta, Eero Kallio, Ilkka Terho A Notation Evaluation of BPMN and UML Activity Diagrams⁴.

³<http://www.bptrends.com/publicationfiles/03-04%20WP%20Notations%20and%20Workflow%20Patterns%20-%20White.pdf>

⁴ http://www.soberit.hut.fi/T-86/T-86.5161/2006/BPMN_vs_UML_final.pdf

Рассмотрим некоторые базовые элементы обеих нотаций, относящиеся к перспективе управления потоком.

Базовые элементы нотации UML AD, относящиеся к перспективе управления потоком:

Узел-Действие:

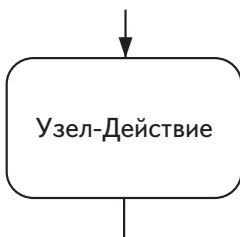


Рисунок 2.6.2

Маршрутные узлы.

Ветвление — Узел выбора направления дальнейшего движения точки управления, соответствует WF-паттерну «Исключающий выбор»:

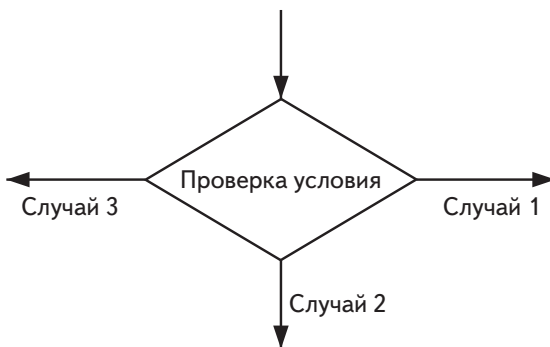


Рисунок 2.6.3

Разделение — Разделение точки управления на несколько точек управления, соответствует WF-паттерну «Параллельное расщепление»:

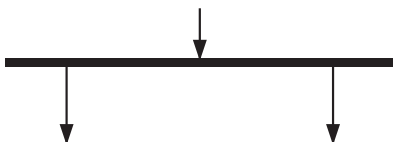


Рисунок 2.6.4

Слияние — Слияние точек управления в одну точку управления, соответствует WF-паттерну «Синхронизация»:

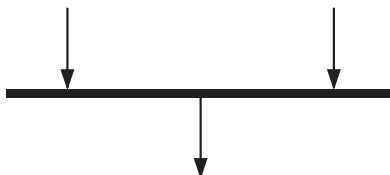


Рисунок 2.6.5

Базовые элементы нотации BPMN, относящиеся к перспективе управления потоком:

Узел-Действие:

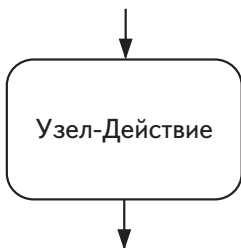


Рисунок 2.6.6

Маршрутные узлы.

В BPMN существует единая форма для маршрутного узла, представляющая собой ромбик:

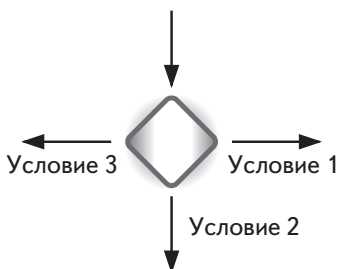


Рисунок 2.6.7

Конкретные маршрутные узлы отличаются изображенными внутри этой формы иконками.

Ветвление — Узел выбора направления дальнейшего движения точки управления:

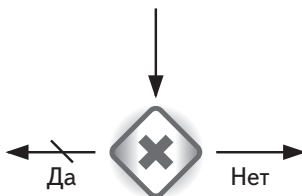


Рисунок 2.6.8

Внутри ромбика содержится иконка — «крестик».

Разделение — Разделение точки управления на несколько точек управления:

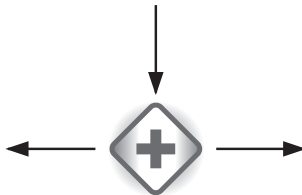


Рисунок 2.6.9

Внутри ромбика содержится иконка — «плюсик».

Слияние — Слияние точек управления в одну точку управления:

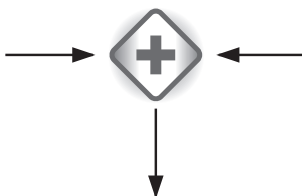


Рисунок 2.6.10

Элемент точно такой же, как и разделение, однако, у него должен быть только один исходящий переход и несколько входящих.

2.6.7 Сравнение графических нотаций

В BPMN нотации — более универсальные элементы. Элементы BPMN нотации определяются парой графических объектов: формой элемента и изображенной внутри нее иконкой. Например, форма для всех маршрутных узлов BPMN одинакова, а поведение определяется иконкой: «крестик» соответствует выбору одного из нескольких направлений, а «плюсик» — разделению точки управления на несколько одновременно перемещающихся точек. Это позволяет использовать различные комбинации форм и иконок вместо того, чтобы вводить новые графические элементы, и, таким образом, можно уменьшить общее число используемых в нотации объектов.

Однако UML AD нотация проще для изучения неподготовленным пользователем, она интуитивно понятна. UML AD нотация использует хотя и не универсальные, но широко известные графические элементы. Например, в ней для выбора одного из нескольких направлений используется «ромбик». А параллельно выполняющиеся узлы-действия в UML AD нотации как правило соединены с элементами разделения-слияния параллельными линиями, что интуитивно соответствует одновременно выполняющимся действиям.

В UML AD нотации изображение процессов очень похоже на блок-схемы, которые изучаются в российских технических ВУЗах и техникумах. В начальной школе при изучении математики в некоторых учебниках также активно используются те же блок-схемы. (Например, Петерсон Л. Г. Математика. Учебники для 1-4 класса.). То есть многим российским пользователям изображения в UML AD нотации сразу будут интуитивно понятны, а для понимания изображений в BPMN нотации придется потратить время и усилия на ее изучение.

У BPMN-нотации есть свои сильные стороны, например, очень велика маркетинговая мощь международных софтверных компаний, продвигающих эту нотацию. Есть элементы, пользоваться которыми в BPMN нотации удобнее, чем в UML нотации.

Пример процесса «заявка на платеж» в UML-нотации:

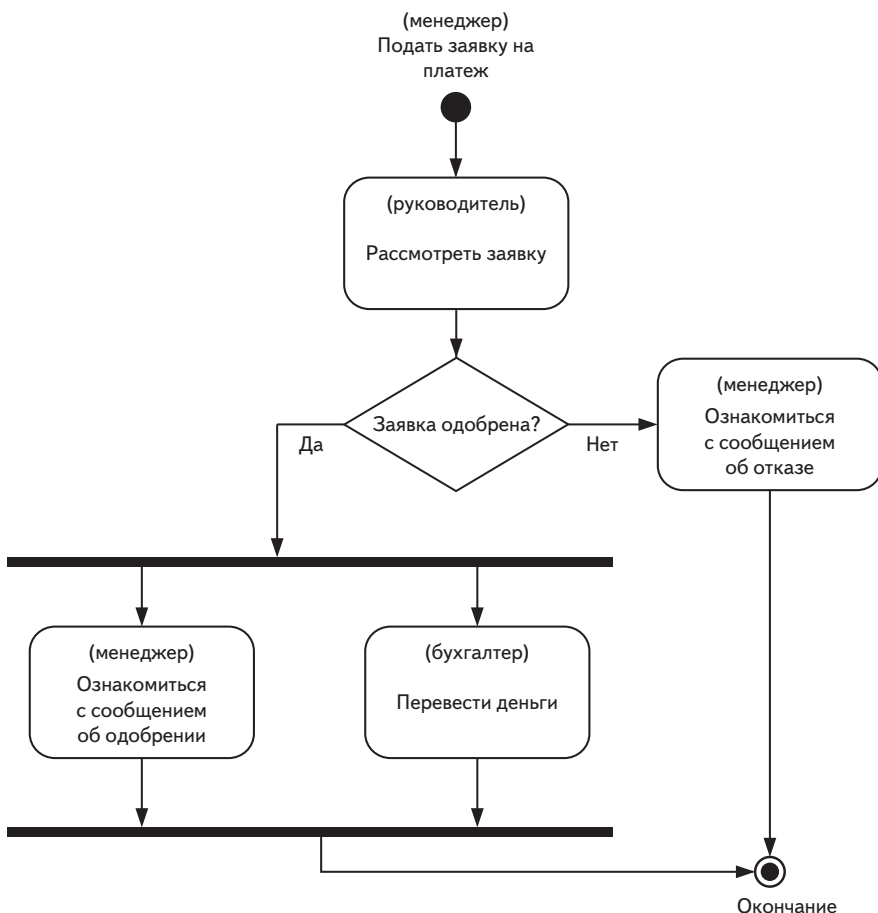


Рисунок 2.6.11

Пример процесса «заявка на платеж» в BPMN-нотации:

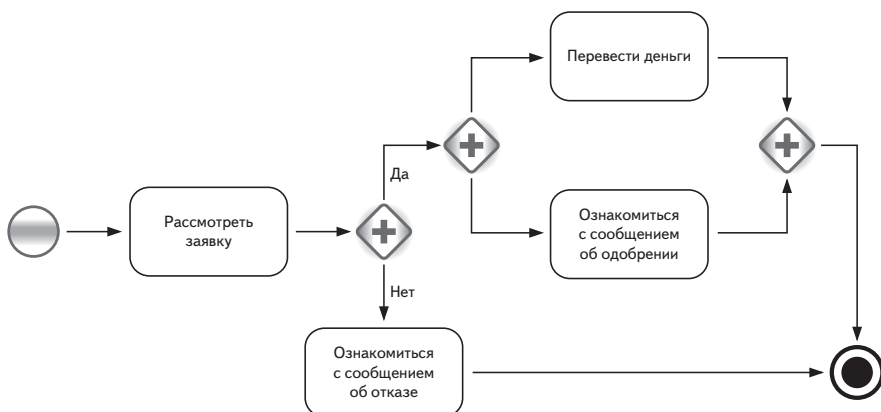


Рисунок 2.6.12

UML нотация имеет очень широкое распространение, например, с ней в той или иной степени знакомо большинство IT специалистов, она интуитивно понятна. Для того, чтобы объяснять схемы несложных бизнес-процессов, нарисованные в UML AD нотации, не требуется много усилий. В случае же BPMN нотации требуются учебные курсы и различные консультации.

Поэтому для описания бизнес-процессов мы преимущественно используем UML AD нотацию. Кратко преимущества нотаций можно сформулировать так:

Преимущества UML нотации:

- UML нотация проще. Ее легче изучать;
- Значительному числу пользователей графы процессов, нарисованные в UML нотации (с движением точек управления бизнес-процесса преимущественно сверху-вниз), более понятны, чем процессы, нарисованные в BPMN нотации.

Преимущества BPMN нотации:

- Более понятные изображения некоторых элементов (например, таймеров);
- Более удобно работать с бизнес-исключениями.

2.7 Возможное расширение стандартов на ситуации, которые существующими стандартами не описываются

В рамках существующих стандартов возможно создание таких комбинаций элементов, поведение которых стандартами не определено. Например, в конструкции, изображенной на рисунке 2.7.1, точка управления в элементе-разделении трансформируется в три «параллельно движущихся» точки управления. Далее, после прохождения элемента-соединения, они окажутся тремя точками управления, идущими друг за другом по одному переходу. Взаимодействие этих точек управления со следующим узлом-действием существующими стандартами не определено.

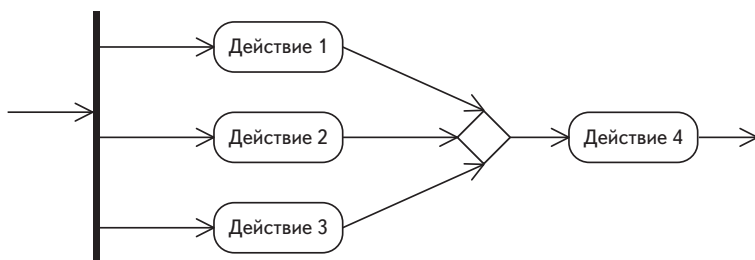


Рисунок 2.7.1

При исполнении бизнес-процессов иногда возникают типичные ситуации, требующие решения, однако решения для таких ситуаций не описываются существующими стандартами. Примером такой ситуации является необходимость замещения сотрудника предприятия другим сотрудником при исполнении задач СУБПиАР (когда пользователь, которому предназначено задание, заболел, находится в отпуске или командировке).

В последующих разделах приводятся возможные расширения стандартов на описанные выше ситуации.

2.7.1 Возможные варианты взаимодействия последовательно идущих по одному переходу точек управления с узлом-действием

Первый вариант возможного взаимодействия

Точки управления становятся в очередь перед узлом-действием и переходят в узел-действие только после того, как предыдущая точка управления его покинет.

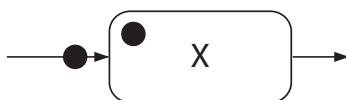


Рисунок 2.7.2

Возможные минусы. При таком поведении одна «медленно выполняющаяся» точка управления может остановить продвижение всех остальных точек управления.

Второй вариант возможного взаимодействия

Для данного узла-действия создается несколько независимых экземпляров узла-действия с соответствующими локальными переменными. Изменение глобальных переменных экземплярами узла-действия происходит в момент покидания точкой управления узла-действия. Значение локальной переменной копируется в глобальную переменную. В данном случае во время выполнения заданий потоки управления независимы. После выполнения действия количество потоков управления не изменяется, они опять «идут» один за другим. При таком поведении в узле-действии позже пришедший поток управления может «обогнать» ранее пришедший поток управления.

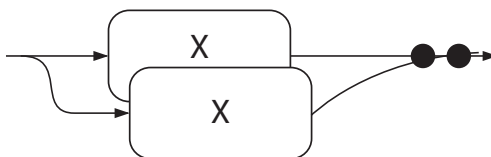


Рисунок 2.7.3

Возможные минусы. Возможность «работы» нескольких последовательных точек управления с одними данными иногда может привести к нарушению целостности данных. Например:

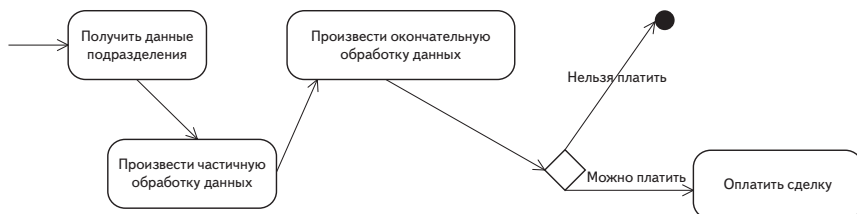


Рисунок 2.7.4

В случае, если по первой сделке платить нельзя, а по второй можно, ситуация, когда вторая точка управления «обгонит» первую, может привести к тому, что первая сделка будет оплачена. В первом варианте такой проблемы не возникнет.

То есть в обоих приведенных случаях есть минусы. Однозначно лучшего варианта из них выбрать нельзя.

2.7.2 Возможные варианты взаимодействия последовательно идущих по одному переходу точек управления с разделением и слиянием

Элемент «разделение» соответствует WF-паттерну «Параллельное расщепление», элемент «слияние» соответствует WF-паттерну «Синхронизация».

Взаимодействие точек управления с разделением можно организовать двумя способами, также как с узлом-действием. Гораздо интереснее случай взаимодействия точек управления со слиянием.

Первый вариант возможного взаимодействия

Предполагается, что в бизнес-процессе каждому элементу «разделение» должен быть поставлен в соответствие единственный элемент «слияние». Каждая пришедшая в «слияние» точка управления «проверяет», все ли одновременно вышедшие с ней из «разделения» точки управления пришли в «слияние». Если нет, то точка управления ждет в «слиянии» остальные точки управления. Если да, то все вышедшие из «разделения» точки управления уничтожаются. На выходе из «слияния» генерируется одна точка управления.

Замечание: В данном случае не обязательно даже, чтобы соответствие разделений и слияний было взаимно однозначным.

Например, допустима следующая конфигурация:

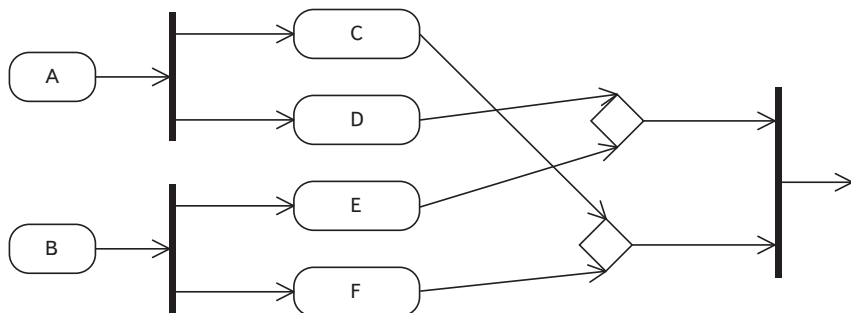


Рисунок 2.7.5

Возможные минусы. Существует практика использования разделений и слияний для синхронизации действий в процессе. Например, на рисунке 2.7.6 слияние «препятствует» рассылке материалов о конференции до заключения договора аренды.

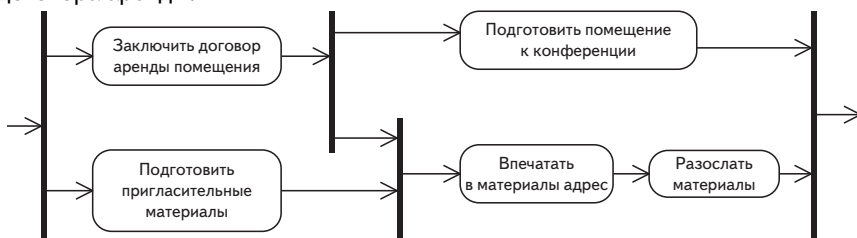


Рисунок 2.7.6

Второй вариант возможного взаимодействия

Для того, чтобы поддерживались такие конструкции, как изображенная на рисунке 2.7.6, можно определить поведение элемента «слияние» в случае нескольких последовательно перемещающихся по одному переходу точек управления следующим образом: по каждому переходу пришедшая в «слияние» точка управления ставится в очередь. Если для каждого входящего перехода «слияния» его очередь заполнена хотя бы одной точкой управления, то все точки управления, находящиеся на первой позиции в очереди соответствующего перехода, уничтожаются и на выходе из «слияния» генерируется одна точка управления. Все остальные точки, находящиеся в очередях в «слияние», перемещаются на одну позицию вперед.

Возможные минусы

1. Если внутри «разделения»-«слияния» одна из точек управления «обогнала» другую, то в узле «слияние» синхронизирован будет набор точек управления, одна из которых была порождена следующей точкой управления, пришедшей в разделение. Возможность такой ситуации может быть источником нестабильности в системе. Вследствие этой возможности в системе могут появиться трудно отлаживаемые бизнес-процессы, которые иногда будут работать «правильно», а иногда «неправильно».

2. Возможно существование бизнес-процессов, в которых количество точек управления заметно возрастает с течением времени. Такие процессы могут создать слишком большую нагрузку на СУБПиАР.

Пример участка схемы процесса с быстро возрастающим количеством точек управления

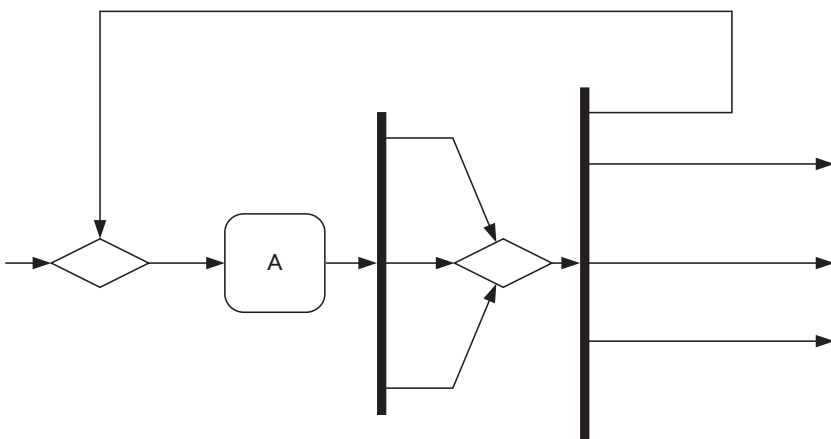


Рисунок 2.7.7

2.7.3 Назначение заместителя для исполнителя заданий

Во многих случаях, если исполнитель текущего задания по каким-либо причинам не может его выполнить в необходимые сроки, например, болеет или находится в командировке, необходимо передать это задание другому исполнителю. Стандартами эта ситуация не описывается, поэтому в СУБПиАР применяется большое количество различных конструкций для замещения одного исполнителя заданий другим.

Мы считаем, что наиболее перспективной является конструкция, основанная на правилах замещения исполнителей. В рамках этого подхода система замещения является независимой от бизнес-процессов. То есть в бизнес-процессе не содержится никакой информации о возможных замещениях исполнителей. Система замещения привязана к людям. Такая организация замещения удобна для управленцев, им так проще принимать решения. Управленцы говорят: «Если я узнаю, что Иванов заболел, то я хотел бы иметь возможность сообщить СУБПиАР, что такие-то задачи таких-то процессов должны быть перенаправлены Петрову, такие-то задачи никому не должны быть перенаправлены, а все остальные задачи должны быть перенаправлены Сидорову». При этом не потребуется искать все бизнес-процессы, в которых задействован Иванов и что-то в них настраивать.

Данная конструкция замещения содержит набор правил, которые последовательно просматриваются в некотором порядке до тех пор, пока либо не будет найдено подходящее правило замещения, либо будет выяснено, что ни одного подходящего правила нет.

Правила назначения заместителя.

Каждое правило содержит функцию над организационной структурой предприятия, которая возвращает пользователя-заместителя. Список параметров правила:

- Замещаемый Пользователь (Пользователь);
- Заместитель (Функция над организационной структурой, возвращающая пользователя);
- Применимо ли правило (формула).

Пример правила назначения заместителя:

- Иванов
- Руководитель сотрудника (Иванов)
- (Роль=«инспекторКадровойСлужбы») & (Бизнес-процесс=«больничный»)

Применение правил замещения Пользователя.

У пользователя может быть одно из двух состояний:

- Активен
- Не активен

Механизм замещения применяется только к пользователям, имеющим статус «не активен». В этом случае из списка правил будут выбраны все правила замещения, относящиеся к данному пользователю. Далее из этих правил будет выбрано первое по порядку правило, которое применимо (выполняется формула в «Применимо ли правило») и заместитель в котором имеет статус «Активен». В список заданий этого пользователя (заместителя) и будет перенаправлено данное задание.

Замечание: Возможны ситуации, в которых у Пользователя не будет заместителя.

2.8 Возможное применение СУБПиАР для автоматизации деятельности государственного ведомства.

Основная идея применения бизнес-процессного подхода в государственном управлении — повысить «прозрачность» административных процессов, а также построить в органах государственного управления аналог производственного конвейера и получить от этого такое же увеличение производительности труда, какое обеспечило применение конвейера на производстве.

В частности, важной государственной задачей является задача организации взаимодействия граждан и различных ведомств через интернет. Эта задача может быть эффективно решена при помощи СУБПиАР. Например, возможна следующая реализация:

Гражданин заходит под своим именем на определенную страницу интернет-портала ведомства, заполняет форму какой-либо заявки и отправляет ее в ведомство. После этого управление переходит СУБПиАР, система создает экземпляр регламентной схемы (бизнес-процесса) и начинает перемещать по нему точки управления. При этом гражданин всегда может зайти на определенную страницу интернет-портала ведомства (в так называемый «личный кабинет») и выполнить свои задачи или посмотреть состояния своих экземпляров регламентных схем.

Пользователями (исполнителями заданий СУБПиАР) в данном случае могут быть как сотрудники ведомств, так и граждане.

Пример экземпляра выполняющейся регламентной схемы:



Рисунок 2.8.1

Однако при внедрении систем управления бизнес-процессами и административными регламентами возникает следующая проблема: В ведомствах уже существуют готовые настроенные и хорошо работающие программные средства, которые автоматизируют некоторые элементы работ по запросам граждан. Такое состояние называется «лоскутной автоматизацией». Оно типично для российских учреждений. Переписывать эти программные средства нецелесообразно. Это потребует времени, переобучения персонала ведомства и дополнительного финансирования.

Поэтому предлагается использовать дополнительный процесс интеграции, позволяющий работать с регламентной схемой как гражданам-пользователям, так и программному обеспечению нескольких ведомств.

Пример такого процесса:

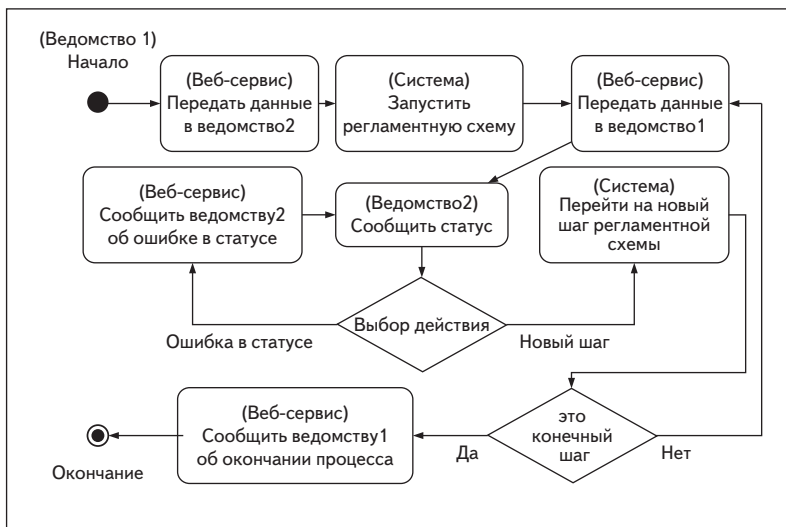


Рисунок 2.8.2

Для каждого экземпляра регламентной схемы запускается экземпляр процесса интеграции. Если элемент работы выполняется при помощи программных средств ведомства, то ведомство обращается к процессу интеграции и процесс интеграции от имени соответствующего исполнителя передвигает точку управления регламентной схемы в следующий узел. Если одно из ведомств закончило выполнять «свои» элементы работ, то процесс интеграции обращается к программным средствам следующего ведомства и сообщает им о необходимости дальнейшего выполнения регламента. Далее процесс интеграции отслеживает действия уже этого ведомства. При этом гражданин или сотрудники ведомства, которое не использует собственные средства автоматизации элементов работ, будет выполнять задания экземпляра регламентной схемы напрямую, не используя процесс интеграции.

К программному обеспечению, реализующему взаимодействие с государственными органами (системе управления бизнес-процессами и административными регламентами), разумно предъявить следующие требования:

- государственные организации должны иметь возможность использования неограниченного количества экземпляров программного обеспечения без увеличения расходов при увеличении количества экземпляров
- граждане должны иметь возможность использования такого программного обеспечения бесплатно
- в случае, если разработчик программного обеспечения по каким-либо причинам перестал устраивать государство, государство должно иметь возможность сменить разработчика, не потеряв при этом программное обеспечение

Этим требованиям удовлетворяет свободное программное обеспечение с открытым кодом.

3. Свободная система управления бизнес-процессами и административными регламентами с открытым кодом

3.1 Введение

В настоящей части книги на примере системы с открытым кодом RunaWFE дано описание основных элементов СУБПиАР. Система RunaWFE свободно распространяется вместе со своими исходными кодами на условиях открытой лицензии LGPL. Система бесплатная. Скачать дистрибутивы и исходный код ее можно через интернет с портала разработчиков свободного программного обеспечения sourceforge.net по адресу:

<http://sourceforge.net/projects/runawfe>.

Система RunaWFE включена в репозиторий свободных программ Sisyphus (<http://sisyphus.ru>). На основе RunaWFE и дистрибутива ALT Linux компания Альт Линукс выпустила специализированный дистрибутив ALT Linux Workflow.

Принято считать, что современная СУБПиАР должна обеспечивать разработку бизнес-процесса в графической среде, исполнение бизнес-процесса, мониторинг состояния бизнес-процесса, ведение истории событий бизнес-процесса, интеграцию приложений при помощи используемых бизнес-процессами коннекторов, администрирование пользователей, а также возможность замещения исполнителей заданий.

Для выполнения этих функций в СУБПиАР служат следующие графические интерфейсы:

- интерфейсы для работы с заданиями исполнителей
- интерфейсы для работы с загруженными в СУБПиАР определениями бизнес-процессов
- интерфейсы для работы с выполняющимися в СУБПиАР экземплярами процессов
- интерфейсы для администрирования пользователей и групп пользователей
- интерфейсы для настройки замещений исполнителей заданий

Для разработки бизнес-процессов обычно применяются графические дизайнеры бизнес-процессов, которые являются отдельными приложениями.

В системе RunaWFE для интеграции приложений реализованы специальные сущности — боты и бот-станции.

В данной части книги на примере системы RunaWFE продемонстрирована вся перечисленная функциональность и пользовательские интерфейсы.

3.2 Основные компоненты системы

RunaWFE состоит из следующих основных компонентов:

- RunaWFE-сервер
- Внешняя бот-станция (необязательный компонент)
- Графический редактор бизнес-процессов
- Клиент-оповещатель о поступивших заданиях (необязательный компонент)

RunaWFE-сервер — это основной компонент системы. RunaWFE-сервер реализует среду исполнения экземпляра процесса в соответствии с его определением. Этот компонент содержит определения загруженных в него бизнес-процессов и выполняющиеся экземпляры бизнес-процессов. Позволяет создавать и изменять свойства пользователей. Генерирует списки заданий и визуальные формы, соответствующие заданиям. Позволяет устанавливать различные права на объекты системы.

Бот-станции содержат ботов, которые периодически опрашивают RunaWFE-сервер. Если выполняющиеся на RunaWFE-сервере экземпляры бизнес-процессов содержат задачи для ботов, загруженных в бот-станцию, то боты выполняют эти задачи и возвращают результаты работы на RunaWFE-сервер. В частности, боты могут представлять собой коннекторы к другим информационным системам. В этом случае бот-станция может служить средством интеграции автоматизированных систем предприятия.

Графический редактор бизнес-процессов служит для создания модели процесса, в которой определяются последовательность выполнения элементов работ и данные, присваиваются роли участникам процесса, вводятся правила маршрутизации, определяются графические формы заданий, используемые участниками процесса для выполнения задач. Редактор бизнес-процессов позволяет сконструировать модель в виде графической диаграммы с описанием деталей этой модели в виде свойств отдельных действий, подпроцессов или процесса в целом. Редактор процессов — средство разработчиков процессов, бизнес-аналитиков, он обеспечивает внесение изменений в бизнес-процесс путем простой модификации графической диаграммы и свойств элементов.

Клиент-оповещатель о поступивших заданиях представляет собой среду доступа пользователей к функциональности RunaWFE-сервера. В частности, отображает списки заданий и визуальные формы заданий. Позволяет пользователям выполнять задания. Позволяет администратору системы устанавливать права на объекты системы. Дает возможность осуществлять мониторинг исполнения экземпляров бизнес процессов. А также реализует оповещение пользователя о поступивших задачах.

Замечание. Если оповещение о поступивших задачах не требуется или достаточно оповещения по электронной почте, то для получения остальной описанной выше функциональности можно клиент-оповещатель не устанавливать. Эта функциональность доступна через обычный браузер.

Краткое описание функциональности графических интерфейсов системы

При помощи web-интерфейса системы пользователь может:

- Получать, фильтровать, выполнять задачи, генерируемые экземплярами бизнес-процессов
- Запускать новые экземпляры бизнес-процессов
- Просматривать состояния выполняющихся экземпляров бизнес-процессов

- Загружать файлы-архивы, содержащие определения бизнес-процессов в систему

При помощи web-интерфейса системы администратор может:

- Создавать/удалять пользователей и группы пользователей
- Включать/исключать пользователей в группы
- Раздавать права на объекты системы пользователям и группам пользователей
- Принудительно останавливать экземпляры бизнес-процессов
- Добавлять, изменять правила замещения пользователей

При помощи графического редактора бизнес-процессов аналитик может разрабатывать бизнес-процессы и экспортировать их в файлы-архивы в файловую систему.

Симулятор бизнес-процессов

Симулятор бизнес-процессов является адаптированной для клиентского компьютера версией RunaWFE-сервера. При помощи симулятора бизнес-процессов можно тестировать разработанные бизнес-процессы на условной конфигурации на клиентском компьютере аналитика, не загружая их в промышленную систему.

3.3 Где скачать исходные файлы и как установить систему

Для того, чтобы лучше познакомиться с функциональностью СУБПиАР, имеет смысл установить систему RunaWFE на компьютере и параллельно с чтением книги производить указанные действия в системе.

RunaWFE распространяется в следующих вариантах:

- В виде специализированных дистрибутивов для конкретных операционных систем
- В виде исполнимых файлов java-машины
- В исходных кодах

Проще всего установить RunaWFE при помощи специализированного дистрибутива для конкретной операционной системы. Например, в случае операционной системы ALT Linux, для того, чтобы скачать специализированный дистрибутив через интернет, можно зайти на страницу скачивания файлов проекта RunaWFE на портале разработчиков свободного программного обеспечения sourceforge — <http://sourceforge.net/projects/runawfe/files>, выбрать папку «Distributives», потом подпапки «Distributives for Linux» и «Distributives for ALTLinux», далее выбрать последнюю по номеру версию системы, войти в папку этой версии и скачать файлы-дистрибутивы в виде rpm-пакетов. Также в случае операционной системы ALTLinux систему RunaWFE можно установить из репозитория свободного программного обеспечения Sisyphus — <http://sisyphus.ru> (пакеты runawfe и runawfe-jboss).

В случае операционной системы Windows для того, чтобы скачать специализированный дистрибутив через интернет, надо зайти на страницу скачивания файлов проекта RunaWFE на портале разработчиков свободного программного

обеспечения sourceforge — <http://sourceforge.net/projects/runawfe/files>, выбрать папку «Distributives», потом подпапку «Distributives for Windows», далее выбрать последнюю по номеру версию системы, войти в папку этой версии и скачать файл-дистрибутив «RunaWFE-Installer.exe». Запуск на выполнение этого файла запустит на компьютере с ОС Windows диалог установки системы RunaWFE. Если вам удобнее устанавливать систему с CD-диска, то из той же папки надо скачать файл «runawfe-x.x.iso» и скопировать его на CD-диск. При вставке этого диска в CD-дисковод компьютера запустится диалог установки системы RunaWFE для ОС Windows.

Вариант распространения в виде исполняемых файлов java-машины используется, если в проекте RunaWFE нет специализированного дистрибутива для ОС, которую вы используете на своем компьютере. В этом случае надо обратиться к документации проекта RunaWFE (например, на сайте wf.runa.ru/rus) и установить на компьютер непосредственно исполняемые файлы java. Вариант распространения в виде исходных кодов предназначен для разработчиков программного обеспечения. Используя исходные коды, они могут модифицировать систему или встраивать ее в какое-то другое программное обеспечение.

3.3.1 Установка системы при помощи специализированных дистрибутивов для операционной системы Linux

Дистрибутивы для ОС Linux состоят из rpm или deb пакетов.

Скачайте пакеты для вашей ОС со страницы загрузки:

<http://sourceforge.net/projects/runawfe/files> (пакеты находятся в Distributives/Distributives for Linux).

В состав поставки входят следующие пакеты:

- runawfe-jboss — jboss необходимый для работы всех типов серверов runawfe
- runawfe-simulation — Симулятор WFE. Позволяет запускать и останавливать WFE сервер с использованием команд из главного меню операционной системы (подменю Офис), а так же содержит ссылку на web интерфейс системы.
- runawfe-gpd — Графический редактор бизнес-процессов.
- runawfe-notifier — Клиент-оповещатель
- runawfe-doc — Документация
- runawfe-commonlibs — Библиотеки, используемые разными типами серверов
- runawfe-common — Общие компоненты главного меню
- runawfe-client-conf — Настраивает клиентские компоненты на WFE сервер
- runawfe-client — Ссылка на web интерфейс WFE сервера
- runawfe-adminkit — Административные скрипты
- runawfe-server — WFE сервер
- runawfe-botstation — Удаленная ботстанция

Для установки пакетов наиболее целесообразно использовать какой-либо менеджер пакетов, например `apt-get` или `yum`. Использование менеджера пакетов позволит автоматически загрузить и установить зависимости для устанавливаемых пакетов.

Пакеты `runawfe-jboss`, `runawfe-commonlibs`, `runawfe-common` являются служебными и ставятся в качестве зависимостей при установке других пакетов. Установка их отдельно от остальных пакетов смысла не имеет.

Пакеты `runawfe-server` и `runawfe-botstation` имеет смысл устанавливать только при установке системы в промышленную эксплуатацию. Для ознакомления с системой рекомендуется установить пакеты `runawfe-simulation`, `runawfe-gpd`, `runawfe-notifier`, `runawfe-doc` и все их зависимости. Такая установка позволит создавать/редактировать бизнес-процессы, посмотреть работу сервера и клиента-оповещателя. Так же `runawfe-simulation` включает тестовую базу данных с предварительно загруженной демонстрационной конфигурацией (похожей на `wfdemo.runa.ru`).

Для установки рекомендуемых пакетов можно выполнить:

```
apt-get install runawfe-simulation runawfe-gpd runawfe-notifier runawfe-doc
```

3.3.2 Установка системы при помощи специализированного дистрибутива для операционной системы Windows

В данном параграфе подробно рассматривается установка системы для ОС Microsoft Windows. Описание установки системы для других ОС можно посмотреть в документации проекта RunaWFE (например, на сайте `wf.runa.ru/rus`).

Перечислим компоненты дистрибутива системы, которые можно установить при помощи специализированного дистрибутива:

Компоненты дистрибутива, относящиеся к клиентской части системы:

- Клиент (web-интерфейс)
- Графический редактор бизнес-процессов
- Симулятор бизнес-процессов
- Клиент-оповещатель о поступивших заданиях

Компоненты дистрибутива, относящиеся к серверной части системы:

- RunaWFE-сервер
- Бот-станция

Для первого знакомства с системой рекомендуется установить на компьютер только клиентские компоненты. Клиентский компонент «симулятор» является адаптированной для компьютера пользователя версией RunaWFE сервера, этот компонент содержит в себе локальную бот-станцию. Поэтому клиентских компонентов достаточно для того, чтобы познакомиться со всей функциональностью RunaWFE.

Установка системы для ОС Windows

Вставьте диск в дисковод (в случае дистрибутива на CD-диске) или запустите на выполнение файл `RunaWFE-Installer.exe` (в случае дистрибутива в виде исполняемого файла). Появится экран мастера установки системы:

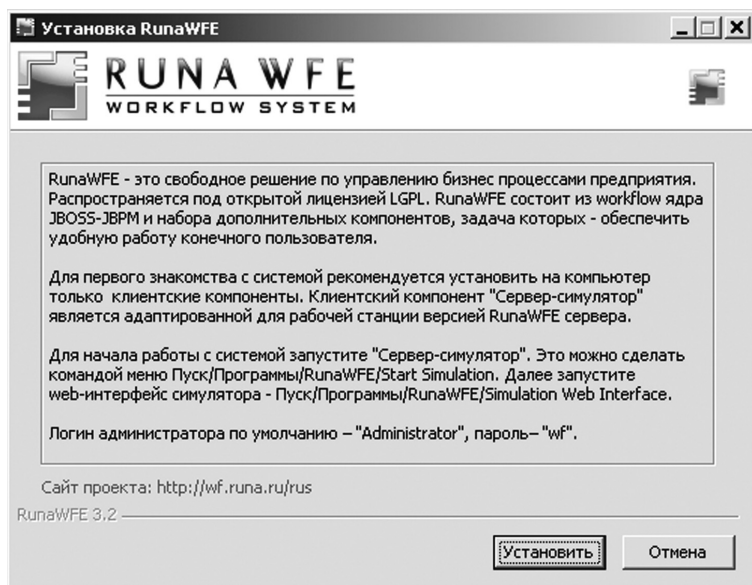


Рисунок 3.3.1

Нажмите «Установить». В следующем мастере установки появится текст свободной лицензии LGPL⁵, под которой распространяется система RunaWFE. В настоящее время не существует официального перевода лицензии LGPL на русский язык, поэтому текст лицензии представлен в оригинале — на английском языке.

⁵ Лицензия LGPL разрешает свободное использование, изменение кода и распространение программного продукта. На использование кода программы под LGPL лицензией разработчиками программного обеспечения в других коммерческих проприетарных программах налагаются некоторые ограничения.

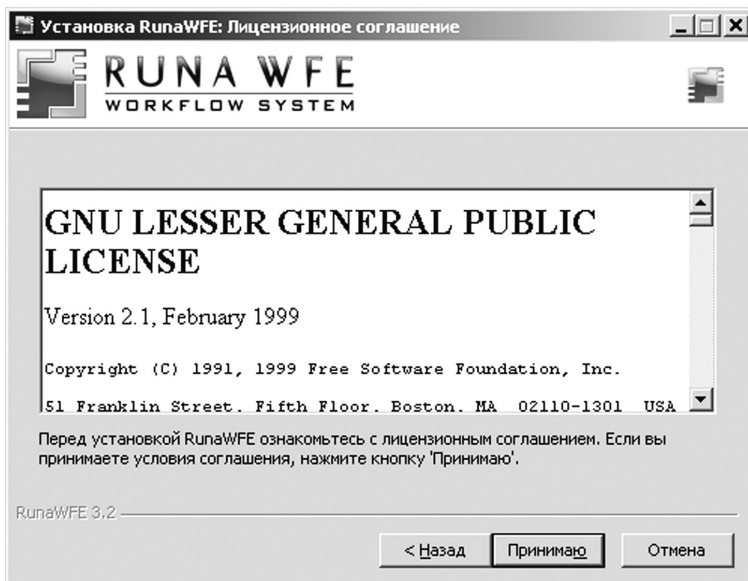


Рисунок 3.3.2

После утвердительного ответа на вопрос о принятии условий лицензии появится выбор — установить на компьютер клиентские или серверные компоненты RunaWFE

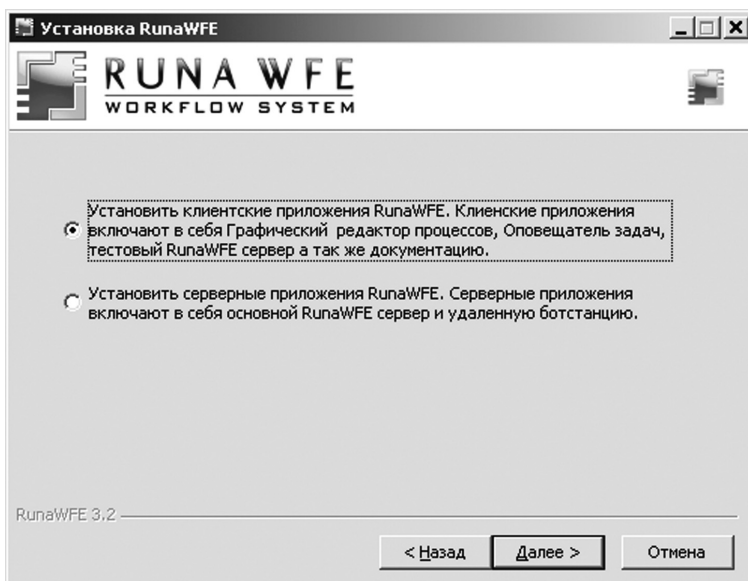


Рисунок 3.3.3

Выберите клиентские приложения и нажмите «Далее»
Появится страница выбора конкретных клиентских компонентов.

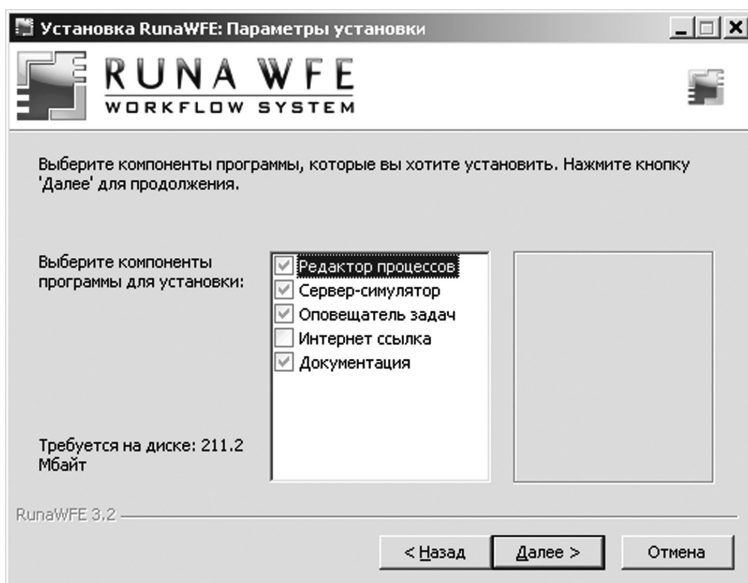
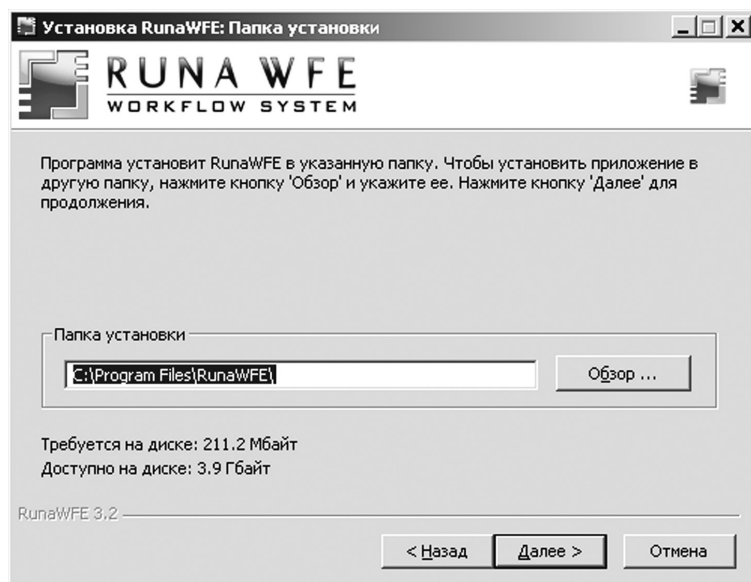


Рисунок 3.3.4



Отметьте все компоненты кроме компонента «Интернет ссылка» и нажмите «Далее».

Рисунок 3.3.5

Выберите папку для установки RunaWFE и нажмите «Далее». В появившемся окне мастера установите самую верхнюю и самую нижнюю галочки:

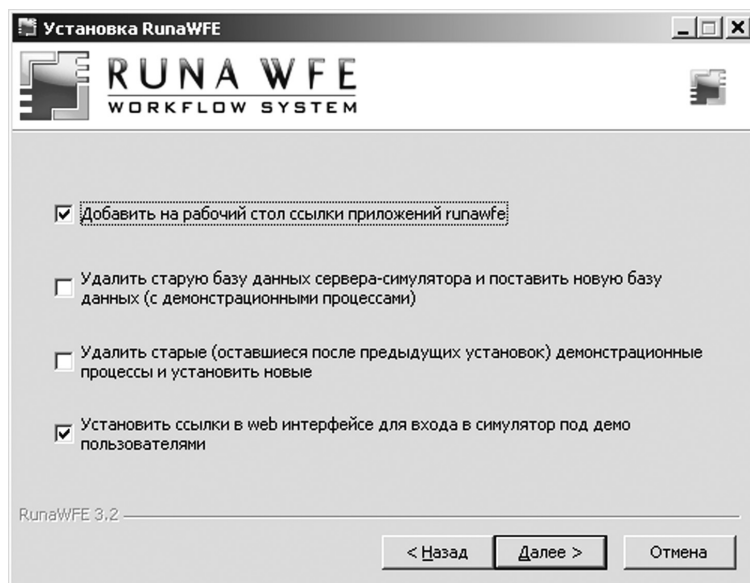
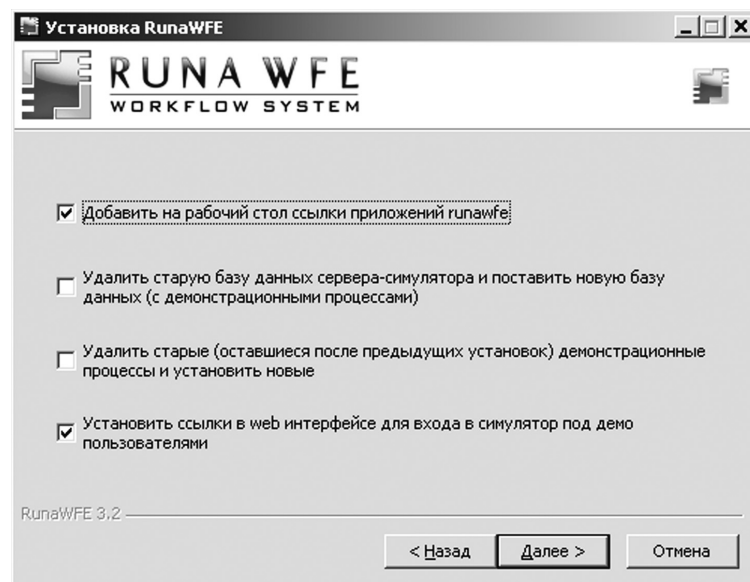


Рисунок 3.3.6



Нажмите «Далее».

Рисунок 3.3.7

В следующем окне мастера оставьте настройки, появившиеся по умолчанию, и нажмите «Далее». Начнется процесс копирования файлов системы. После того, как система будет установлена, появится следующее окно мастера:



Рисунок 3.3.8

Нажмите «Готово». Процесс установки будет завершен.

Работать с системой можно через системное меню (Пуск / Программы / RunaWFE)

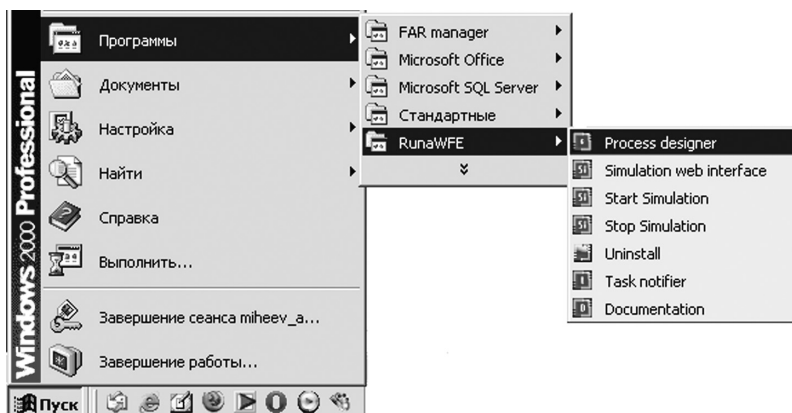


Рисунок 3.3.9

или через расположенные на рабочем столе иконки.

Для возможности запуска и выполнения бизнес-процессов необходимо запустить RunaWFE сервер (симулятор). Симулятор можно запустить, например, командой меню Пуск / Программы / RunaWFE / Start Simulation. Далее для работы с системой через web-интерфейс рекомендуется выполнить команду Пуск / Программы / RunaWFE / Simulation Web Interface. Логин администратора по умолчанию — «Administrator» (существенно, что с большой буквы), пароль администратора — «wfw».

Замечание. RunaWFE-сервер уже содержит в себе локальную бот-станцию. Поэтому специально устанавливать бот-станцию как отдельный компонент в данном случае не требуется.

3.4 Работа с системой

3.4.1 Вход в систему

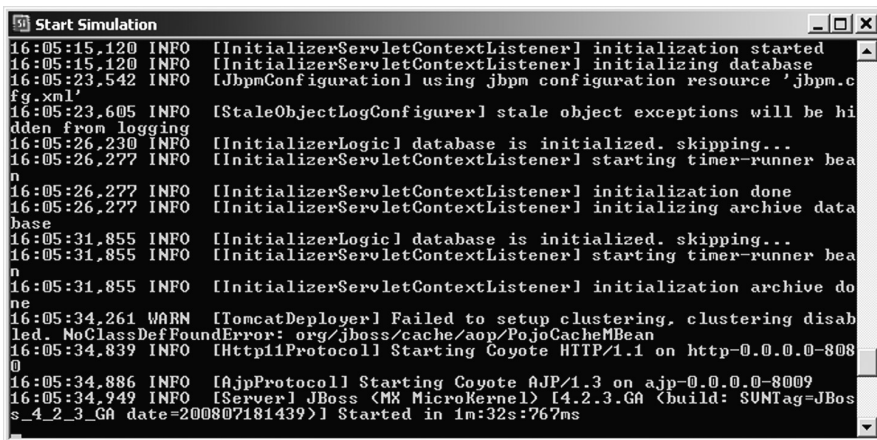
Для того, чтобы можно было войти в систему, RunaWFE-сервер (симулятор) должен быть запущен. Симулятор можно запустить командой меню, например:

/ Программы / Офис / RunaWFE / Start Simulation в случае ОС ALT

Linux или

Пуск / Программы / RunaWFE / Start Simulation в случае ОС Windows.

При запуске симулятора появится консольное окно:



```
Start Simulation
16:05:15.120 INFO [InitializerServletContextListener] initialization started
16:05:15.120 INFO [InitializerServletContextListener] initializing database
16:05:23.542 INFO [JbpmConfiguration] using jbpm configuration resource 'jbpm.c
fg.xml'
16:05:23.605 INFO [StaleObjectLogConfigurer] stale object exceptions will be hi
dden from logging
16:05:26.230 INFO [InitializerLogic] database is initialized. skipping...
16:05:26.277 INFO [InitializerServletContextListener] starting timer-runner bea
n
16:05:26.277 INFO [InitializerServletContextListener] initialization done
16:05:26.277 INFO [InitializerServletContextListener] initializing archive data
base
16:05:31.855 INFO [InitializerLogic] database is initialized. skipping...
16:05:31.855 INFO [InitializerServletContextListener] starting timer-runner bea
n
16:05:31.855 INFO [InitializerServletContextListener] initialization archive do
ne
16:05:34.261 WARN [TomcatDeployer] Failed to setup clustering, clustering disab
led. NoClassDefFoundError: org/jboss/cache/aop/PojoCacheMBean
16:05:34.839 INFO [Http11Protocol] Starting Coyote HTTP/1.1 on http-0.0.0-808
0
16:05:34.886 INFO [AjpProtocol] Starting Coyote AJP/1.3 on ajp-0.0.0-8009
16:05:34.949 INFO [Server] JBoss (MX MicroKernel) [4.2.3.GA (build: SUNTag=JBos
s_4_2_3_GA date=200807181439)] Started in 1m:32s:767ms
```

Рисунок 3.4.1

Надпись «...INFO [Server] JBoss ... Started in ...» означает, что симулятор запущен.

После этого с системой можно работать через web-интерфейс. Это можно сделать как через клиент-оповещатель о поступивших заданиях, так и через обычный браузер.

Проще всего запустить web-интерфейс системы при помощи команды Пуск / Программы / RunaWFE / Simulation Web Interface. После выполнения этой команды появится следующее окно ввода логина и пароля пользователя.

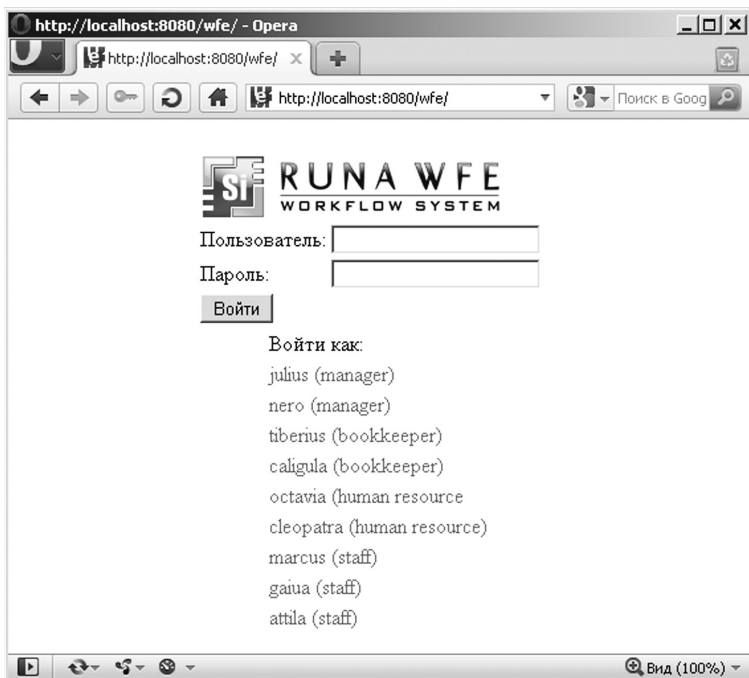


Рисунок 3.4.2

Логин администратора системы по умолчанию — «Administrator» (существенно, что с большой буквы), пароль администратора — «wfe». Также в данном варианте установки на странице входа в систему расположен набор ссылок, при клике на которые можно входить в систему как пользователи демо-конфигурации. Это сделано для удобства изучения системы, чтобы не приходилось каждый раз вводить логин и пароль.

3.4.2 Меню системы

После входа в систему на экране появится страница, в левом верхнем углу которой находится меню системы. В зависимости от прав пользователя у него могут быть показаны не все пункты меню, изображенные на рисунке.



Рисунок 3.4.3

Дадим краткое описание пунктов меню системы RunaWFE. В последующих разделах эти пункты меню будут описаны более подробно.

Меню «Список заданий». При выполнении команды меню «Список заданий» открывается форма списка заданий для данного пользователя. Здесь пользователь может, кликнув на задание, открыть форму задания, ввести в нее данные, а также отметить выполнение задания. Также в списке заданий пользователь может искать, фильтровать задания, выводить в строках задания значения переменных бизнес-процессов.

Меню «Запустить процесс». На странице, соответствующей пункту меню «Запустить процесс» находится список определений бизнес-процессов. Здесь пользователь может запустить бизнес-процесс, посмотреть схему и другие свойства бизнес-процесса, посмотреть описание бизнес-процесса. Если у пользователя есть соответствующие права, он может загрузить новый бизнес-процесс в систему или загрузить новую версию уже существующего процесса.

Меню «Запущенные процессы». На странице, соответствующей пункту меню «Запущенные процессы», находится список экземпляров бизнес-процессов, доступных для чтения данному пользователю. Здесь пользователь может посмотреть состояния выполняющихся экземпляров бизнес-процессов, в частности — положение текущих точек управления на схеме бизнес-процесса, текущие значения переменных и ролей экземпляра бизнес-процесса, а также историю событий экземпляра бизнес-процесса. Если у пользователя есть соответствующие права, он может остановить выполнение экземпляра бизнес-процесса. Также в списке экземпляров бизнес-процессов пользователь может искать, группировать, фильтровать экземпляры бизнес-процессов, выводить в строках значения переменных бизнес-процессов.

Меню «Исполнители». На странице, соответствующей пункту меню «Исполнители», находится список потенциальных исполнителей заданий (пользователей и групп пользователей), доступных для чтения данному пользователю. На этой странице можно завести или удалить исполнителя, завести или удалить группу исполнителей, включить (исключить) исполнителя или группу исполнителей в другую группу. Также для исполнителя можно установить статус (Активен / Не активен) настроить список замещений.

Меню «Бот станции». Боты в системе RunaWFE — это специальные компьютерные приложения, которые также как и люди могут быть исполнителями заданий. Бот-станция — это компьютерная среда, в которой функционируют боты. Находящиеся в бот-станции боты периодически опрашивают RunaWFE-сервер. Если выполняющиеся на сервере экземпляры бизнес-процессов содержат задачи для исполнителей ботов, то боты выполняют эти задачи и возвращают результаты работы на RunaWFE-сервер. На странице, соответствующей пункту меню «Бот станции», находится список зарегистрированных бот-станций. Здесь пользователь может посмотреть свойства бот-станций состояния бот-станций, свойства входящих в бот-станцию ботов, а также задач, которые они могут выполнять. Также в меню «Бот станции» можно завести новую бот-станцию, изменить параметры бот-станции, запустить/остановить периодическую активацию бот-станции, а также изменить свойства входящих в бот-станцию ботов. В частности можно добавить новое задание боту или изменить/удалить уже существующее задание.

Меню «Система». На странице, соответствующей пункту меню «Система» находится список полномочий исполнителей на действия с системой, которые настраивает администратор.

3.4.3 Список заданий

На странице, соответствующей пункту меню «Список заданий», находится список заданий данному пользователю. В строке задания по умолчанию показывается имя задания, совпадающее с именем узла-действия, в котором находится точка управления, соответствующая данному заданию. Также в строке задания показывается имя бизнес-процесса, к которому относится задание, номер экземпляра бизнес-процесса данные которого могут содержаться в форме задания, имя роли, соответствующей заданию и еще некоторая информация.

На рисунке изображен список заданий, содержащий одно задание пользователю Волков.

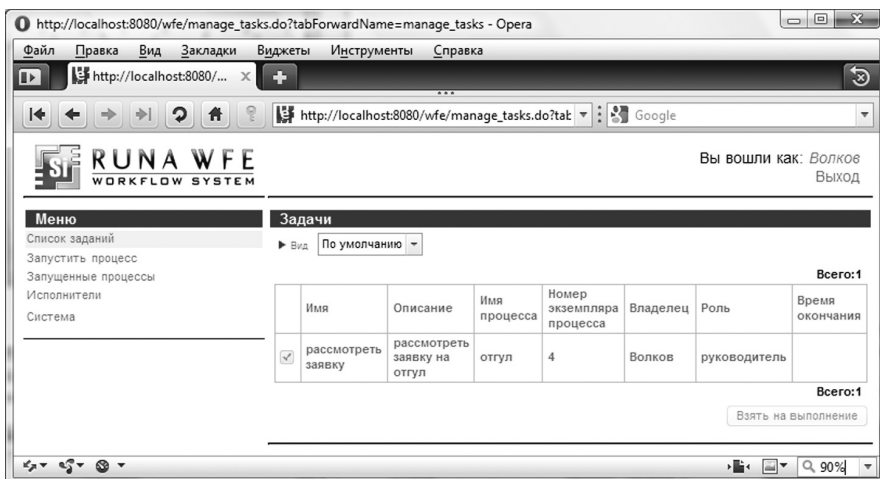


Рисунок 3.4.4

Если пользователь еще не просматривал задание, то в строке задания используется жирный шрифт. Справа под списком заданий расположена кнопка «Взять на выполнение». Если задание является заданием группе сотрудников, то один из них может сразу не выполнять задание, а взять его себе на выполнение. Для этого надо кликнуть на кнопку «Взять на выполнение». После этого действия у всех остальных членов группы это задание пропадет из их списков заданий.

3.4.4 Проигрыватель форм

При клике на имя задания вызывается проигрыватель форм, в котором открывается связанная с заданием форма.



Рисунок 3.4.5

Сортировка, группировка и фильтрация заданий

При клике на красном треугольнике в верхней части формы или надписи «Вид» рядом с ним открывается подформа настройки сортировки, группировки и фильтрации. Закрыть подформу можно, повторно кликнув на треугольнике или надписи «Вид».

Пример настройки группировки:

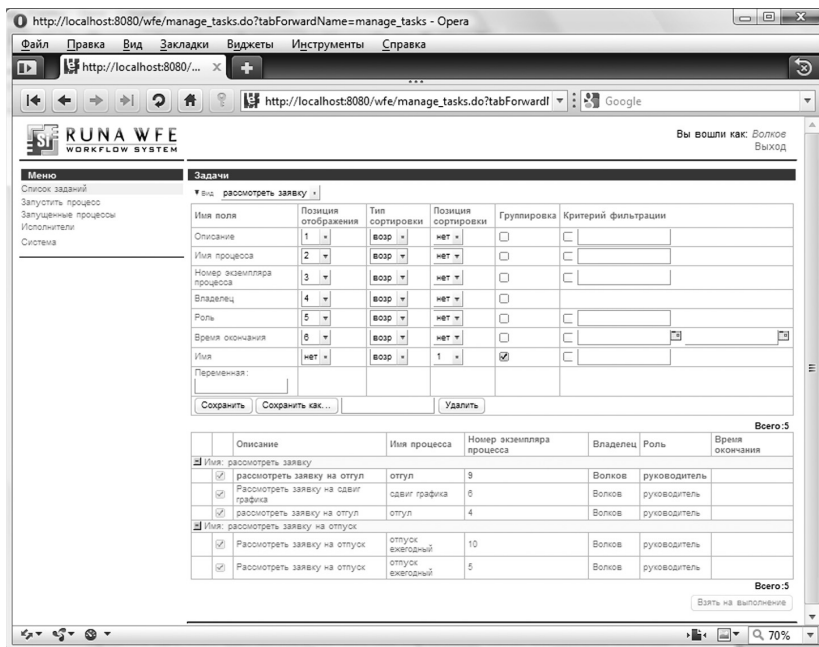


Рисунок 3.4.6

Подформа также содержит пустое поле «Переменная». При помощи этого поля задания можно сортировать, группировать и фильтровать по значениям переменных бизнес-процесса. Для того, чтобы добавить в фильтр переменную, надо ввести в это поле имя переменной. После выполнения команды «сохранить» в строке фильтра появится имя переменной. Теперь в поле можно вводить имя следующей переменной и т.д. После того, как введенная переменная появится в строке фильтра, для нее можно устанавливать сортировку и группировку. Фильтр будет применен ко всем задачам, соответствующим запущенным процессам, содержащих переменную с таким именем.

3.4.5 Определения процессов и запуск процессов

На странице, соответствующей пункту меню «Запустить процесс» находится список определений бизнес-процессов. Имена и иконки бизнес-процессов, доступных для запуска данному пользователю, являются одновременно ссылками, при клике на которые открывается стартовая форма этого бизнес-процесса или же сразу создается экземпляр процесса (в случае отсутствия стартовой формы).

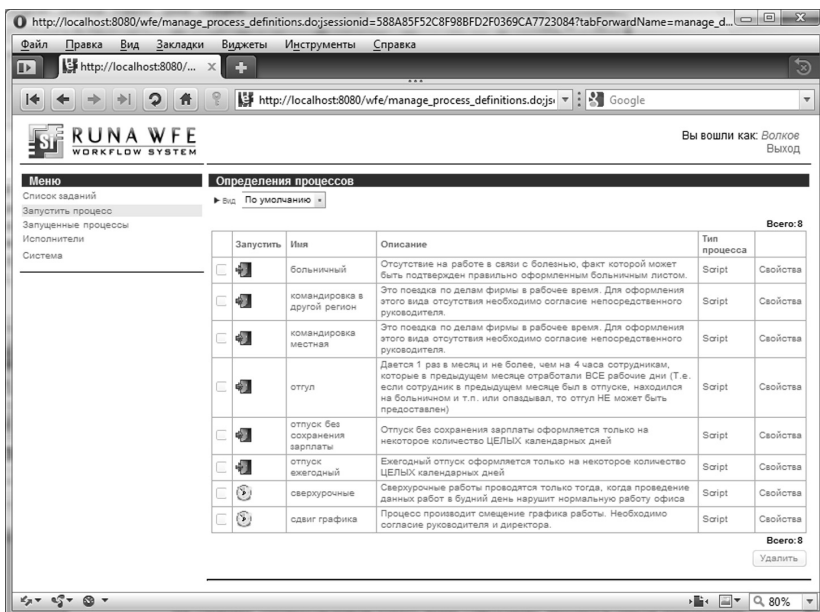


Рисунок 3.4.7

В поле «Описание» находится краткое описание процесса. Если кликнуть на него, то откроется полный вариант описания:

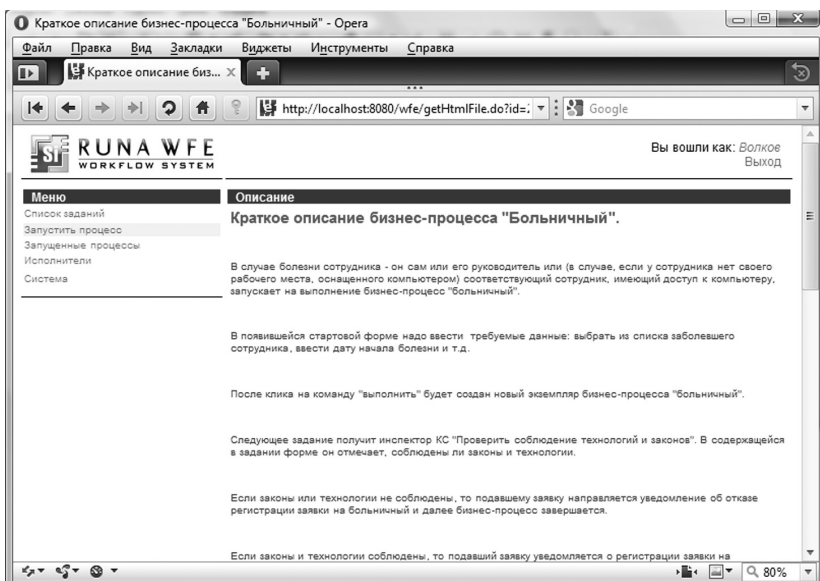


Рисунок 3.4.8

Сортировка, группировка и фильтрация определений бизнес-процессов

При клике на красном треугольнике в верхней части формы или надписи «Вид» рядом с ним открывается подформа настройки сортировки, группировки и фильтрации (заккрыть подформу можно, повторно кликнув на треугольнике или надписи «Вид»).

Настройка подформы аналогична настройке подформы для заданий

Как запустить бизнес-процесс.

При клике на имени бизнес-процесса (или на иконке бизнес-процесса) появляется стартовая форма, если она содержится в определении бизнес-процесса. Если у бизнес-процесса нет стартовой формы, то сразу создается и запускается на выполнение новый экземпляр бизнес-процесса.

Пример заполненной стартовой формы:

The screenshot shows a web browser window with the URL `http://localhost:8080/wfe/submit_start_process_instance.do?id=25`. The browser is Opera. The page title is "RUNA WFE WORKFLOW SYSTEM". The user is logged in as "Волков" (Volkov) and can click "Выход" (Exit). The left sidebar contains a "Меню" (Menu) with items: "Список заданий" (Task List), "Запустить процесс" (Start Process), "Запущенные процессы" (Running Processes), "Исполнители" (Executors), and "Система" (System). The main content area is titled "Стартовая форма" (Start Form) and contains a form titled "Заявка на местную командировку" (Local Business Trip Request). The form fields are: "Сотрудник:" (Employee) with a dropdown menu showing "Волков"; "Дата время с (дд.мм.гггг чч:мм):" (Date and time from) with a text input showing "05.03.2010 14:29"; "Количество часов:" (Number of hours) with a dropdown menu showing "4 часа"; "Количество минут:" (Number of minutes) with a dropdown menu showing "0 минут"; "Причина:" (Reason) with a text input showing "доклад"; and "Комментарий:" (Comment) with a text input showing "конференция". At the bottom of the form is a "Запустить" (Start) button.

Рисунок 3.4.9

При клике на командную кнопку «запустить» стартовой формы создается и запускается на выполнение экземпляр бизнес-процесса.

Свойства бизнес-процесса.

При клике на команду «свойства» в строке, содержащей определение бизнес-процесса открывается страница свойств бизнес-процесса:

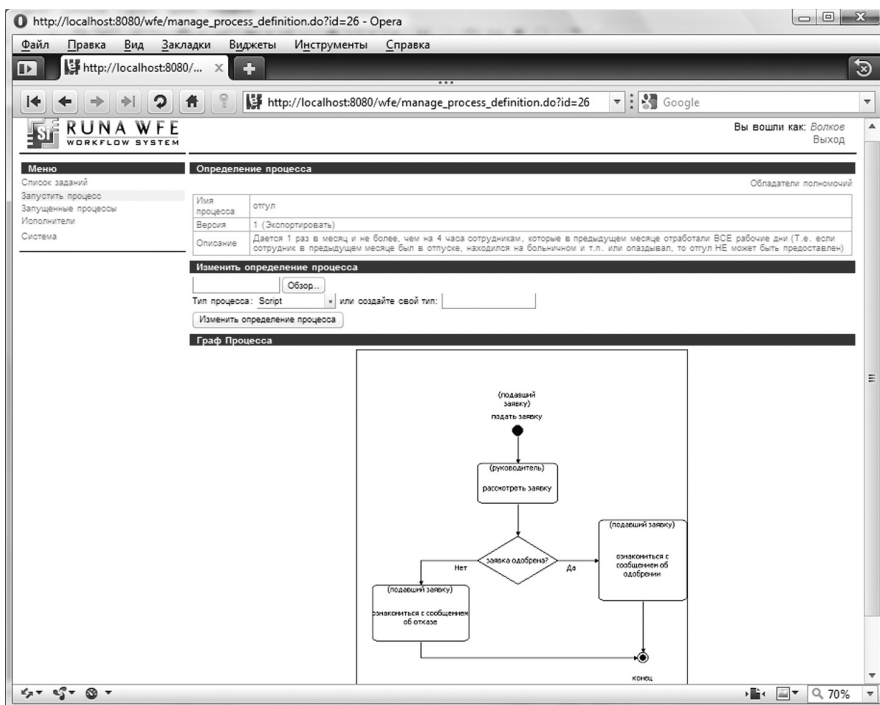


Рисунок 3.4.10

На этой странице находится описание процесса. Кликнув в строке «Версия» на ссылку «Экспортировать», можно записать этот бизнес-процесс в виде файла *.par. Можно также использовать уже готовый файл *.par для изменения определения процесса. Для этого нужно выбрать этот файл с помощью кнопки «Обзор». Далее нажать на кнопку «Изменить определение процесса». Кликнув на ссылку в правом верхнем углу «Обладатели полномочий», увидим список обладателей полномочий этого процесса:

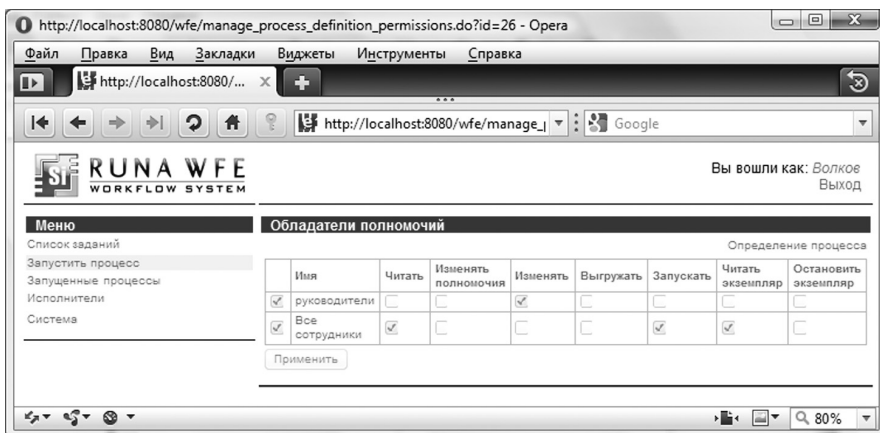


Рисунок 3.4.11

Как загрузить новый бизнес-процесс.

Надо войти в меню «Определения процессов» и выполнить команду (нажать на ссылку) «Загрузить определение процесса». На появившейся странице надо выбрать тип процесса из списка существующих типов (или ввести новый тип), ввести путь к файлу, содержащему определение бизнес-процесса и кликнуть на команде «выполнить». В случае, если определение бизнес-процесса не содержит ошибок, оно будет загружено в систему.

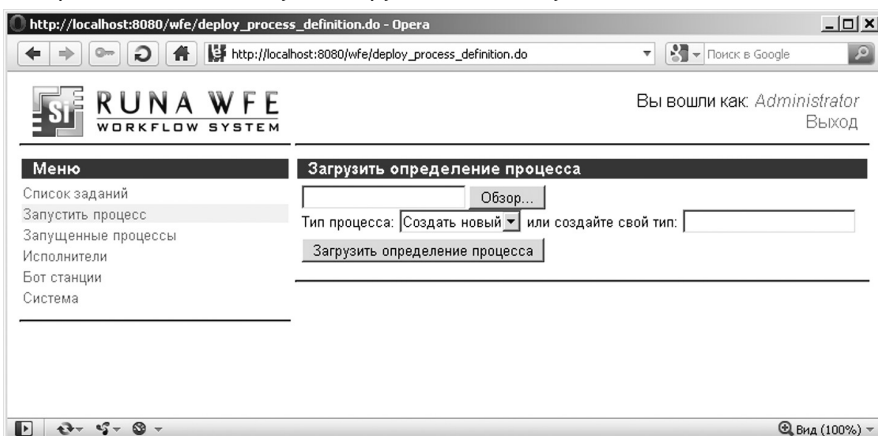


Рисунок 3.4.12

Как выгрузить бизнес-процесс.

Надо войти в меню «Определения процессов», поставить галочки напротив бизнес-процессов, которые надо выгрузить из системы и кликнуть на командной кнопке «выгрузить». Отмеченные бизнес-процессы будут выгружены из системы вместе со всеми своими экземплярами.

Как дать права Исполнителю на определение бизнес-процесса.

Надо войти в меню «Определения процессов», выбрать строку, содержащую требуемый бизнес-процесс, и выполнить в ней команду «свойства». На появившейся странице надо кликнуть на ссылку «Обладатели полномочий». Откроется страница, содержащая таблицу прав на определение бизнес-процесса.

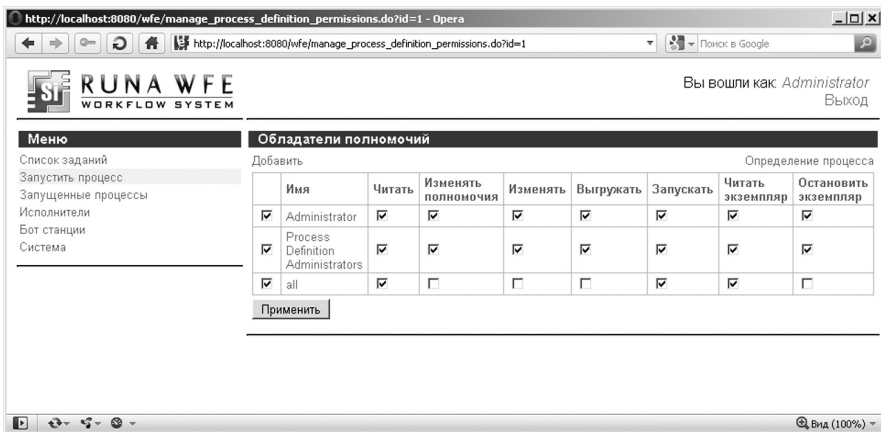


Рисунок 3.4.13

Если Исполнителя, которому надо дать права, нет в таблице, то его надо добавить при помощи команды «Добавить». Далее в строке данного исполнителя надо поставить галочки в соответствующих позициях, после чего выполнить команду «Применить».

Замечание. Таблица прав Исполнителей на определение бизнес-процесса содержит два особых права:

- Читать экземпляр
- Отменять экземпляр

Наличие галочки в данном поле означает, что для любого вновь созданного экземпляра этого бизнес-процесса данный пользователь получит на него права по умолчанию — соответственно читать экземпляр, отменять экземпляр.

Как загрузить новую версию определения бизнес-процесса.

Надо войти в меню «Определения процессов», найти строку, содержащую определение бизнес-процесса и кликнуть на «свойства» в этой строке. На появившейся странице в разделе «Изменить определение процесса» надо ввести путь к файлу, содержащему определение бизнес-процесса и затем выбрать тип процесса или создать новый (по умолчанию тип процесса установлен на текущий тип изменяемого процесса), а затем кликнуть на команде «выполнить». В случае, если определение бизнес-процесса не содержит ошибок, оно будет загружено в систему.

Если новое определение процесса в поле выбора файла не выбрано, то определение процесса изменено не будет, а тип процесса будет изменен на выбранный тип.

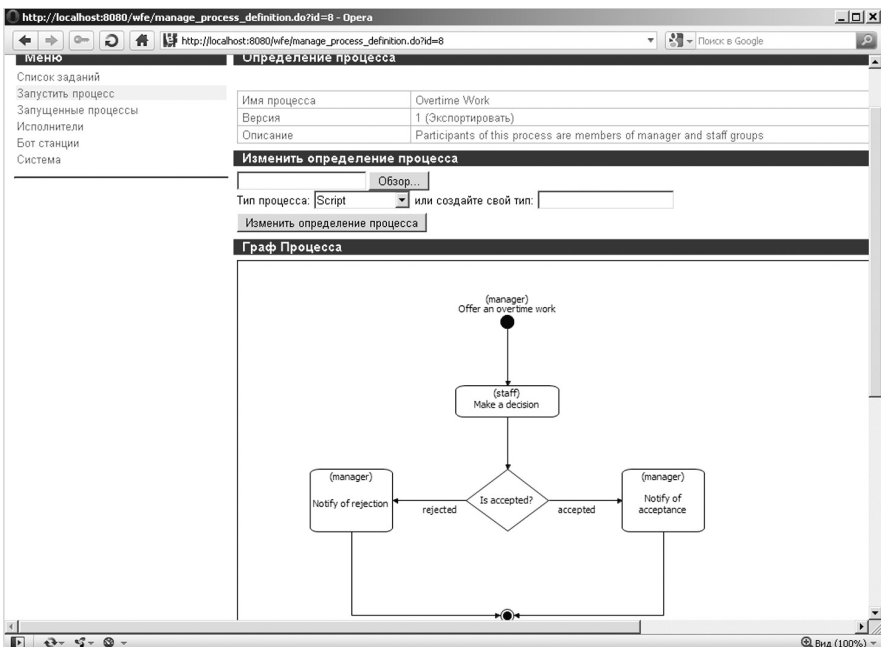


Рисунок 3.4.14

Замечание. Все уже существующие экземпляры данного бизнес-процесса продолжают выполняться в соответствии со старым определением бизнес-процесса, а вновь созданные экземпляры будут соответствовать новому определению.

Замечания. Права, соответствующие новому определению бизнес-процесса будут точно такими же, как и права, соответствующие старому.

3.4.6 Запущенные процессы

На странице, соответствующей пункту меню «Запущенные процессы», находится список экземпляров бизнес-процессов, доступных для чтения данному пользователю:

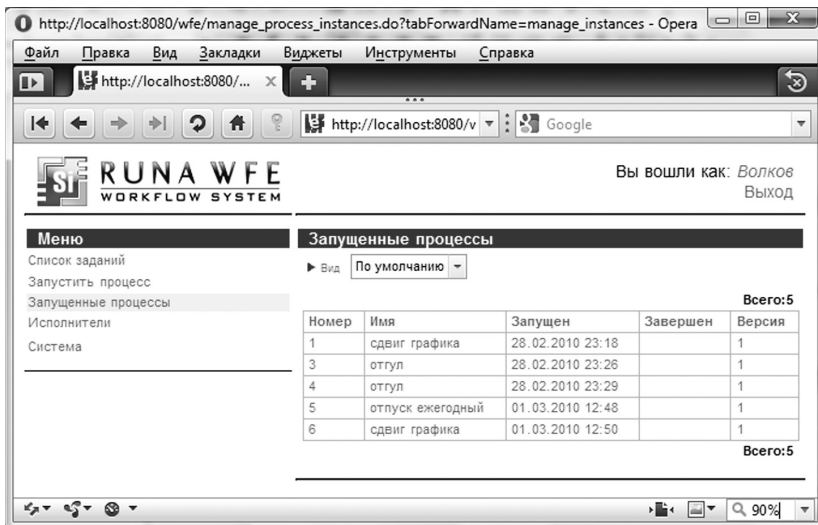


Рисунок 3.4.15

При клике на экземпляр бизнес-процесса открываются свойства экземпляра:

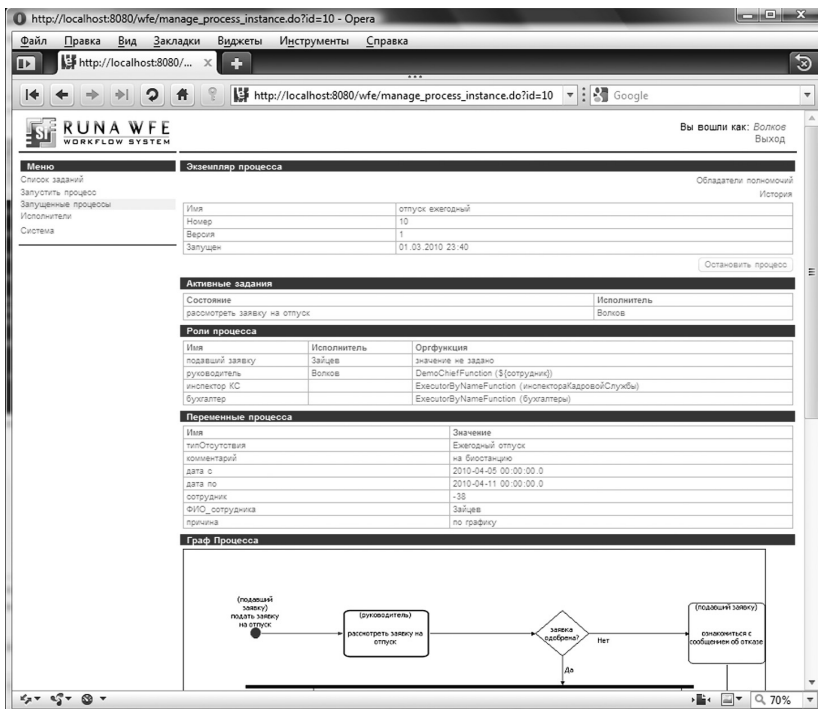


Рисунок 3.4.16

Можно остановить процесс, если пользователь обладает правом на это действие.

Также на этой странице можно увидеть, кто сейчас является исполнителем Активных заданий этого процесса, Роли, Переменные Граф процесса. Кликнув на Историю в правом верхнем углу, получим историю процесса:

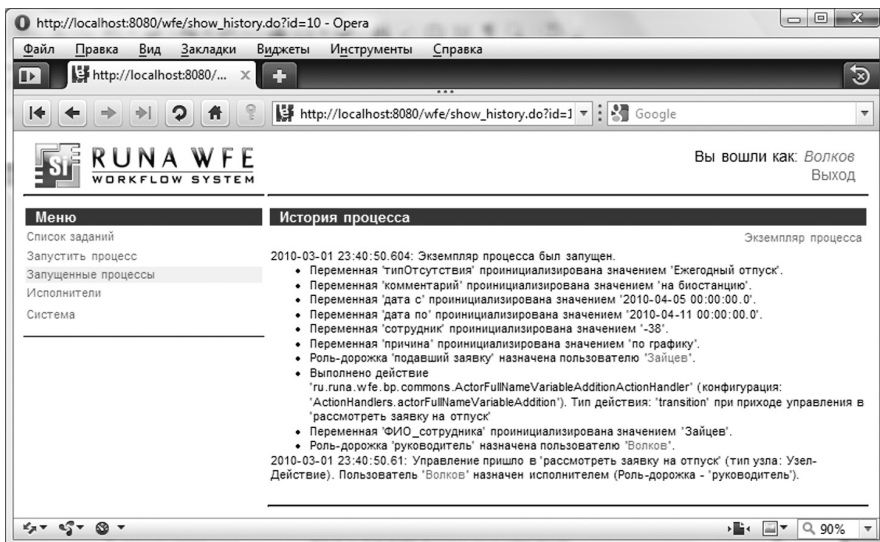


Рисунок 3.4.17

Сортировка, группировка и фильтрация экземпляров бизнес-процессов

При клике на красном треугольнике в верхней части формы или на пикси «Вид» рядом с ним открывается подформа настройки сортировки, группировки и фильтрации. Закрыть подформу можно, повторно кликнув на треугольнике или надписи «Вид».

Настройка подформы аналогична настройке подформы для заданий.

Как посмотреть состояние и значения переменных бизнес-процесса.

Надо войти в меню «Экземпляры процессов», найти строку, содержащую соответствующий экземпляр бизнес-процесса и кликнуть на этой строке.

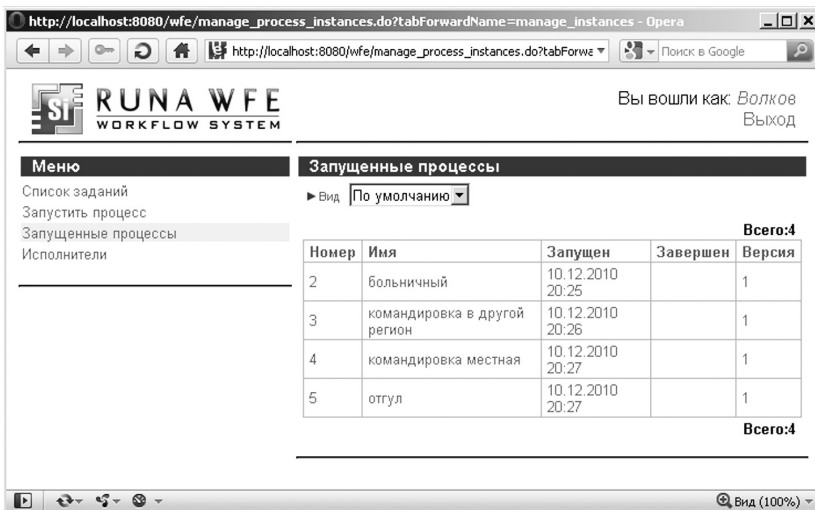


Рисунок 3.4.18

Появится страница, содержащая название узлов-действий, в которых находятся точки управления экземпляра бизнес-процесса, значения всех переменных бизнес-процесса, таблицу ролей-дорожек и ссылку («Обладатели полномочий») на таблицу прав на экземпляре бизнес-процесса.

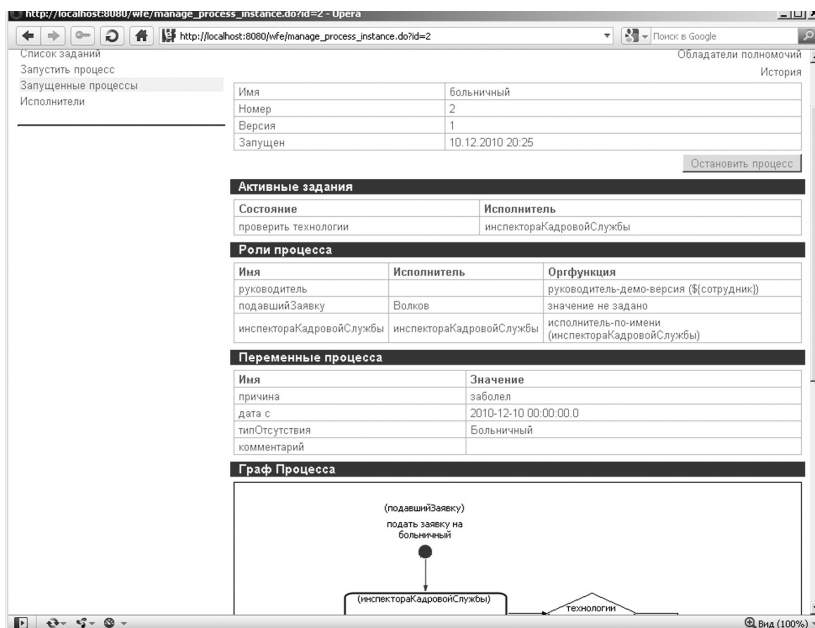


Рисунок 3.4.19

Как отменить выполнение экземпляра бизнес-процесса.

Надо войти в меню «Экземпляры процессов», найти строку, содержащую соответствующий экземпляр бизнес-процесса, и кликнуть на этой строке. На появившейся странице кликнуть на командной кнопке «Остановить процесс».

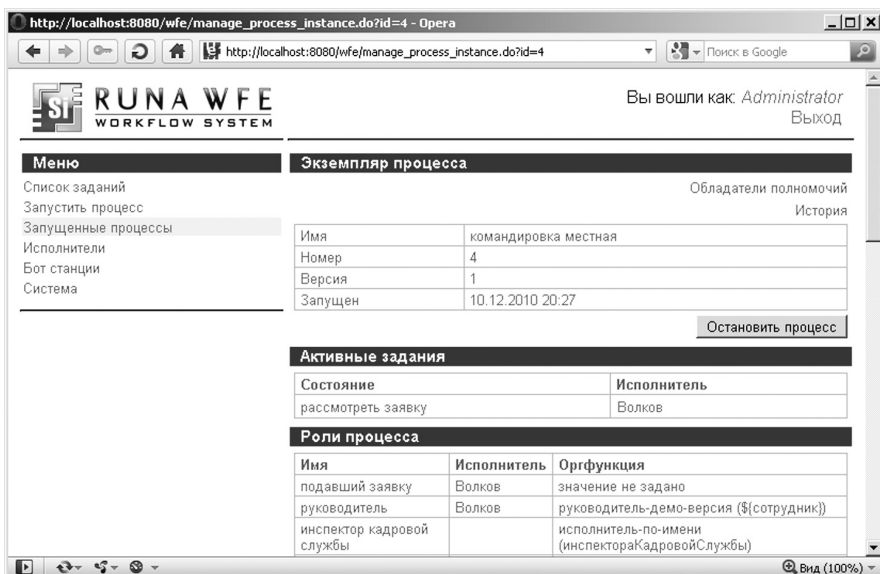


Рисунок 3.4.20

3.4.7 Управление пользователями и группами пользователей

На странице, соответствующей пункту меню «Исполнители», находится список потенциальных исполнителей заданий (пользователей и групп пользователей), доступных для чтения данному пользователю.

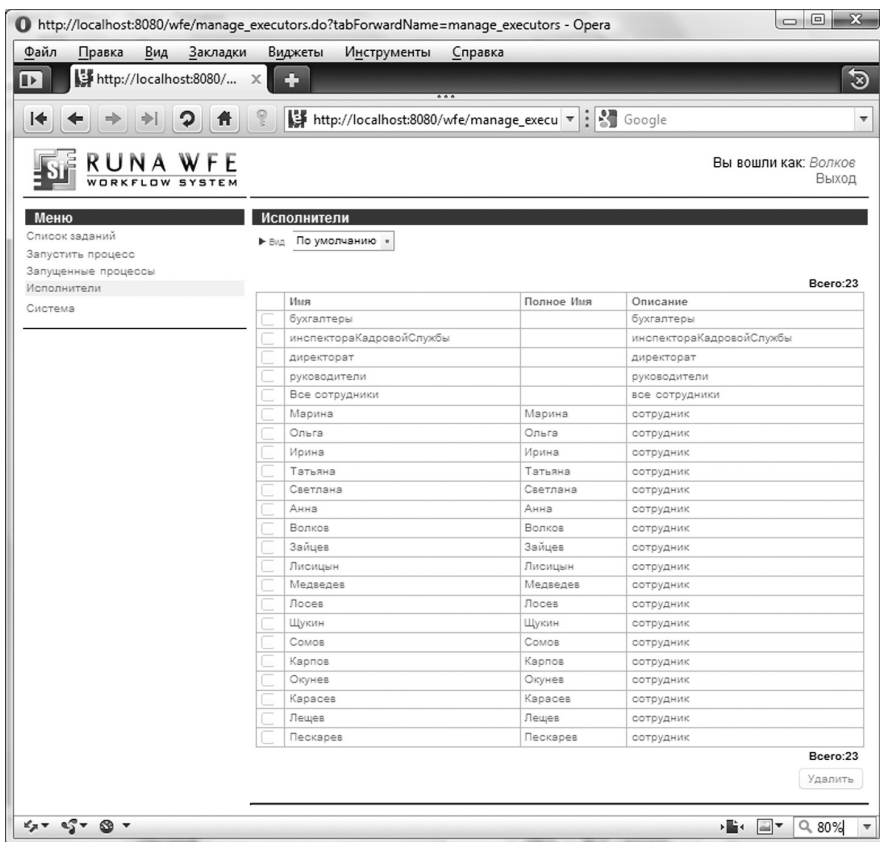


Рисунок 3.4.21

При клике на Имя исполнителя открывается страница свойств исполнителя:

В случае пользователя на этой странице находятся:

- Свойства исполнителя, где отражена информация о пользователе
- Статус, где проставляется признак активности пользователя
- Поле для ввода пароля
- Группы исполнителя, в которые входит пользователь
- Заместители. Тут находится список правил замещения

Замечания.

1. Пользователь может видеть только те данные, на которые у него есть права на чтение.
2. Если признак активности пользователя снят, то для пользователя активизируются правила замещения, которые перенаправляют его задания заместителям. Подробнее об этом написано в разделе «Назначение заместителей»

В правом верхнем углу можно посмотреть список обладателей полномочий, связанных с данным пользователем, кликнув на ссылку «Обладатели полномочий».

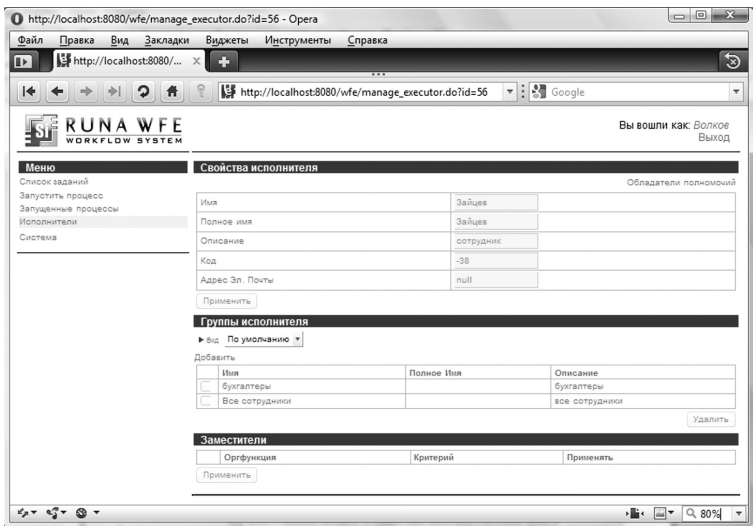
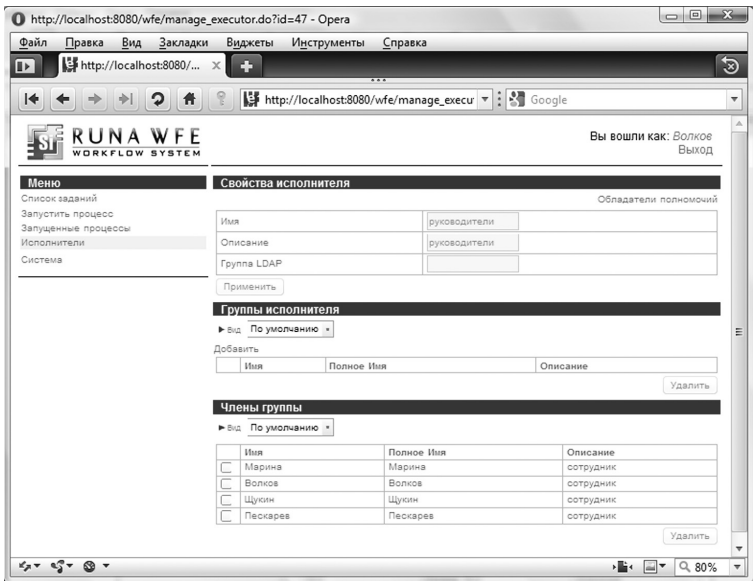


Рисунок 3.4.22



В случае группы пользователей на этой странице находится информация о пользователе, список обладателей полномочий, связанных с данной группой, список групп, входящих в данную группу и список групп, в которые входит данная группа:

Рисунок 3.4.23

Сортировка, группировка и фильтрация исполнителей

При клике на красном треугольничке в верхней части формы или надписи «Вид» рядом с ним открывается подформа настройки сортировки, группировки и фильтрации (заккрыть подформу можно, повторно кликнув на треугольничке или надписи «Вид»).

Настройка подформы аналогична настройке подформы для заданий

Как завести нового Пользователя.

В меню «Исполнители» надо выполнить команду (нажать на ссылку) «Создать пользователя». Появится страница заведения нового Пользователя:

The screenshot shows a web browser window with the address `http://localhost:8080/wfe/create_executor.do?executorType=actor`. The page title is "RUNA WFE WORKFLOW SYSTEM". The user is logged in as "Administrator" and can click "Выход" (Logout). On the left is a "Меню" (Menu) with links: "Список заданий" (Task list), "Запустить процесс" (Start process), "Запущенные процессы" (Running processes), "Исполнители" (Executors - selected), "Бот станции" (Bot stations), and "Система" (System). The main area is titled "Создать нового пользователя" (Create new user) and contains a form with the following fields: "Имя *" (Name - required), "Полное имя" (Full name), "Описание" (Description), "Код" (Code), and "Адрес Эл. Почты" (Email address). There is a "Применить" (Apply) button at the bottom of the form. The browser's status bar at the bottom shows "Вид (100%)" (View (100%)).

Рисунок 3.4.24

На этой странице надо заполнить поля и выполнить команду «Применить».

Поле «Имя» соответствует логину пользователя, оно обязательно для заполнения. Все остальные поля необязательны. Поле «Код» соответствует табельному номеру сотрудника и должно быть уникально в рамках системы.

После того, как Пользователь заведен, для того, чтобы он мог войти в систему, ему еще необходимо задать пароль (см. раздел «Как изменить параметры Пользователя») и дать права на вход в систему (см. раздел «Как дать права Исполнителю на объект система»).

Как изменить параметры Пользователя

Надо войти в меню «Исполнители» и кликнуть на Пользователя, параметры которого требуется изменить. На появившейся странице, если у вас есть соответствующие права, вы сможете изменить параметры пользователя и его пароль. Для изменения статуса пользователя необходимо иметь права на изменение статуса.

http://localhost:8080/wfe/manage_executor.do?id=32 - Opera

http://localhost:8080/wfe/manage_executor.do?id=32

Панель меню: Меню, Список заданий, Запустить процесс, Запущенные процессы, Исполнители, Бот станции, Система

Свойства исполнителя

Имя * Волков

Полное имя Волков

Описание сотрудник

Код 19

Адрес Эл. Почты null

Применить

Статус

Активен ☒

Применить

Пароль

Новый пароль *

Повторите пароль *

Применить

Группы исполнителя

Вид По умолчанию

Добавить

	Имя	Полное Имя	Описание
☐	директорат		директорат
☐	руководители		руководители
☐	Все сотрудники		все сотрудники

Удалить

Заместители

Добавить правило Добавить терминатор

Рисунок 3.4.25

Как завести новую Группу Пользователей.

В меню «Исполнители» надо выполнить команду «Создать группу». Появится страница заведения новой Группы Пользователей:

На этой странице надо заполнить поля и выполнить команду «Применить».

Как изменить параметры Группы Пользователей.

Надо войти в меню «Исполнители» и кликнуть на Группу Пользователей, параметры которой требуется изменить. На появившейся странице, если у вас есть соответствующие права, вы сможете изменить параметры Группы Пользователей.

Как включить Исполнителя в Группу Пользователей

Надо войти в меню «Исполнители» и кликнуть на Группу Пользователей, в которую надо включить Исполнителя. В таблице «члены группы» надо выполнить команду «добавить». На появившейся форме надо поставить галочки напротив Исполнителей, которых надо добавить в группу, и выполнить команду «Добавить» (в нижней части страницы).

Как дать права Исполнителю на объект «система».

Надо войти в меню «Система». Если Исполнитель отсутствует в появившейся таблице, то его надо добавить при помощи команды «Добавить» в верхней части таблицы. Далее в строке данного исполнителя надо поставить галочки в соответствующих позициях, после чего выполнить команду «Применить».

Как дать права Исполнителю на другого Исполнителя.

Надо войти в меню «Исполнители», кликнуть на исполнителе, на которого надо дать права. Если в таблице «Обладатели полномочий» на появившейся странице нет Исполнителя, которому надо дать права, то необходимо его добавить при помощи команды «Добавить». Далее в строке данного исполнителя надо поставить галочки в соответствующих позициях, после чего выполнить команду «Применить».

3.4.8 Пункт меню «Система».

На странице, соответствующей пункту меню «Система» находится список полномочий исполнителей на действия с системой. Список полномочий настраивает администратор.

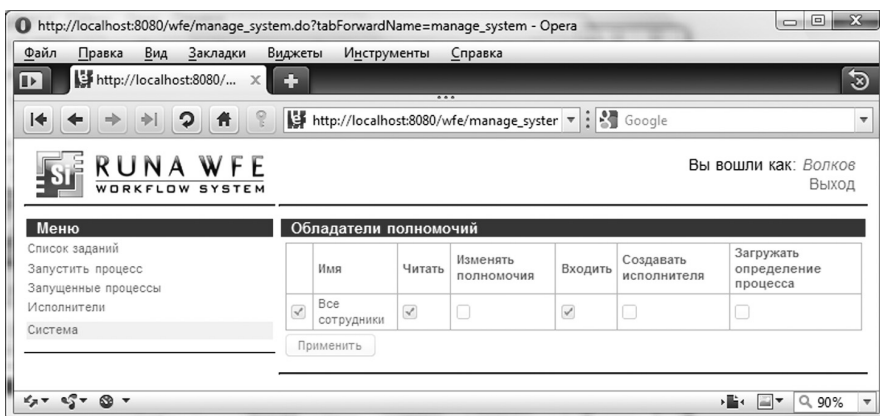


Рисунок 3.4.26

3.4.9 Бот-станции

Концепция ботов и бот-станций

В системе RunaWFE исполнителями заданий могут быть как люди, так и специальные компьютерные приложения — боты. Бот-станция — это среда, в которой работают боты. Находящиеся в бот-станции боты периодически опрашивают RunaWFE-сервер. Если выполняющиеся на сервере экземпляры бизнес-процессов содержат задачи для ботов, то боты выполняют эти задачи и возвращают результаты работы на RunaWFE-сервер.

Используя ботов, можно при помощи системы управления бизнес-процессами также решить задачу интеграции разнородных приложений предприятия в единую корпоративную информационную систему: В разных узлах бизнес-процесса через соответствующую задачу боты, связанные с различными информационными системами предприятия, получают из экземпляра бизнес-процесса значения переменных, а также изменяют значения переменных. Таким образом, через значения переменных экземпляра бизнес-процесса разнородные информационные системы предприятия могут обмениваться между собой информацией.

Рассмотрим реализацию бот-станций более подробно.

Конфигурирование бот-станций.

Настройка бот-станций и ботов производится через меню «Бот станции». Конфигурация RunaWFE сервера, поставляемая по умолчанию, содержит одну бот-станцию (localbotstation) и двух ботов, используемых в демонстрационных процессах.

У бот-станции можно устанавливать статус — «запущена» или «остановлена». При этом соответственно запускается или останавливается периодическая активация ботов.

С бот-станцией должен быть связан один из исполнителей. От имени этого исполнителя бот-станция будет обращаться к RunaWFE серверу для того, чтобы прочитать свои параметры. С каждым ботом также должен быть связан один из исполнителей, от имени которого бот будет выполнять задания в системе.

Права на работу с бот-станциями устанавливаются в меню бот-станции по клику на ссылку «Владельцы прав»:



Рисунок 3.4.27

В частности у пользователя — бот-станции должны быть права на чтение на работу с бот-станциями:

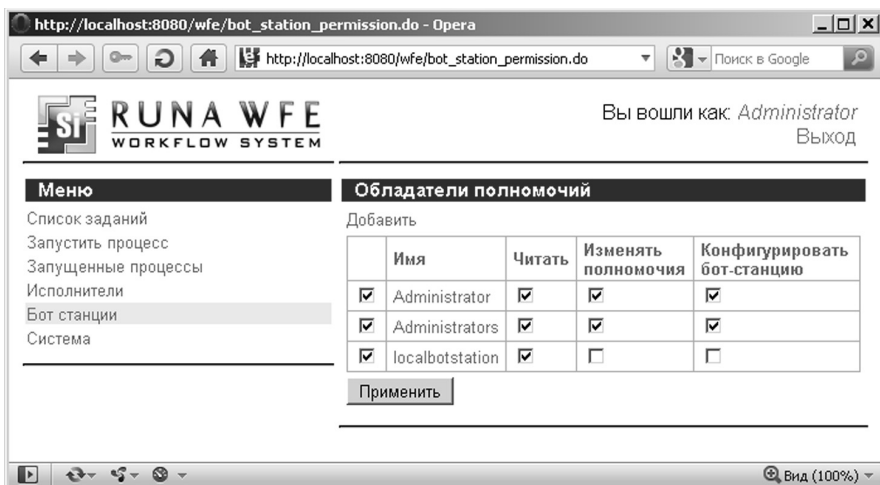


Рисунок 3.4.28

Как посмотреть список бот-станций.

Надо войти в меню «Бот-станции»:

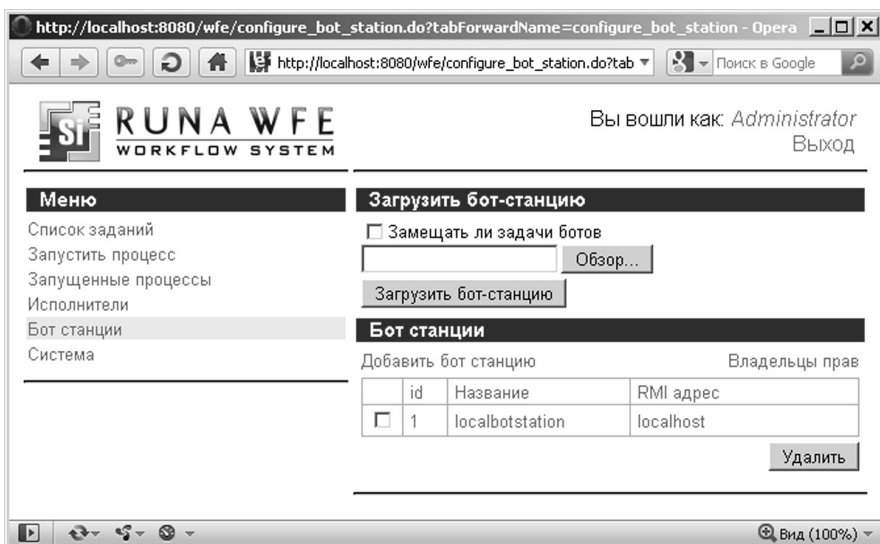


Рисунок 3.4.29

и кликнуть на имени бот-станции:

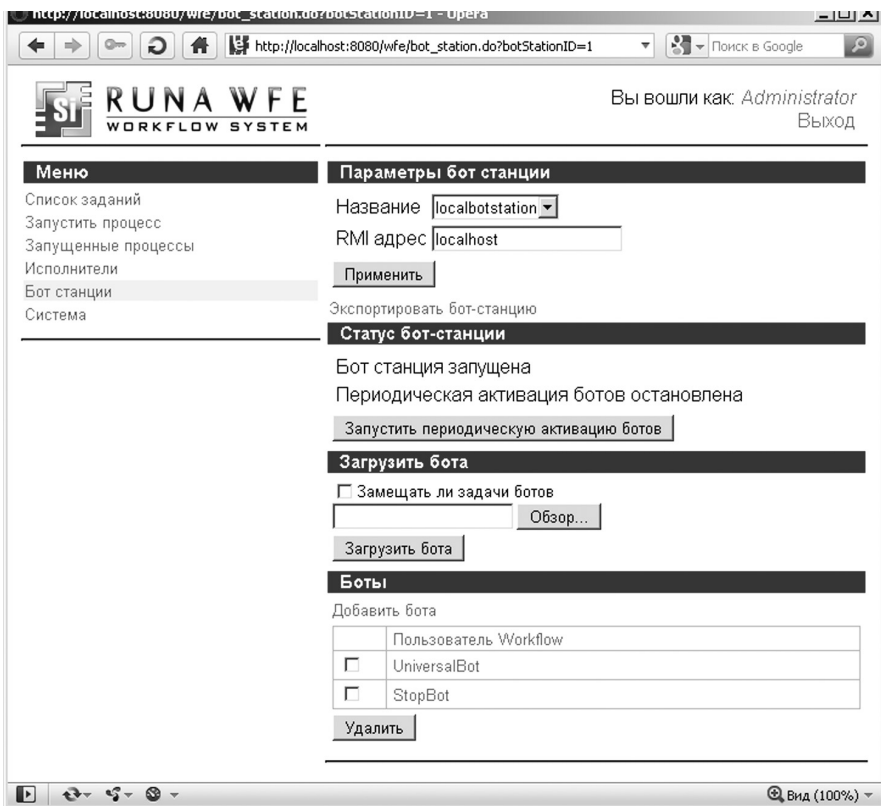


Рисунок 3.4.30

Кликнув на имени бота, можно редактировать его параметры.

Изменение параметров ботов

Для изменения параметров бота необходимо выбрать изменяемого бота на странице информации по бот-станции, перейдя по ссылке с именем бота. Изменение параметров бота предлагается производить в секции «Параметры бота». После выполнения команды «Применить» новые параметры вступят в силу немедленно без перезапуска системы и будут использованы при очередном вызове ботов.

Параметрами бота являются: имя бота (соответствует логину пользователя), пароль бота, список заданий, выполняемых ботом. Список заданий состоит из имени задания, ссылки на обработчик и конфигурации задания.

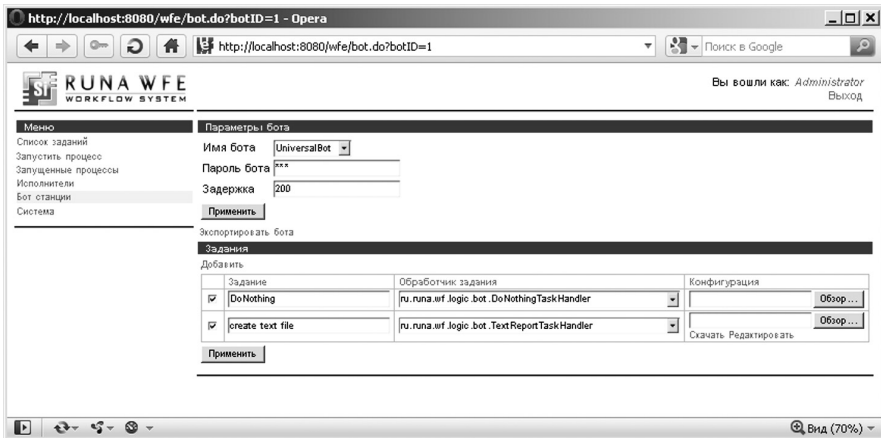


Рисунок 3.4.31

Как завести бота.

Бота надо сначала завести как обычного пользователя системы, задать ему пароль и дать ему права на логин в систему.

Потом бота надо завести как бота, указать при этом заведенного для бота пользователя и еще раз вписать его пароль в соответствующее поле.

Для добавления бота в бот-станцию перейдите на страницу с информацией по бот-станции и перейдите по ссылке «Добавить бота».

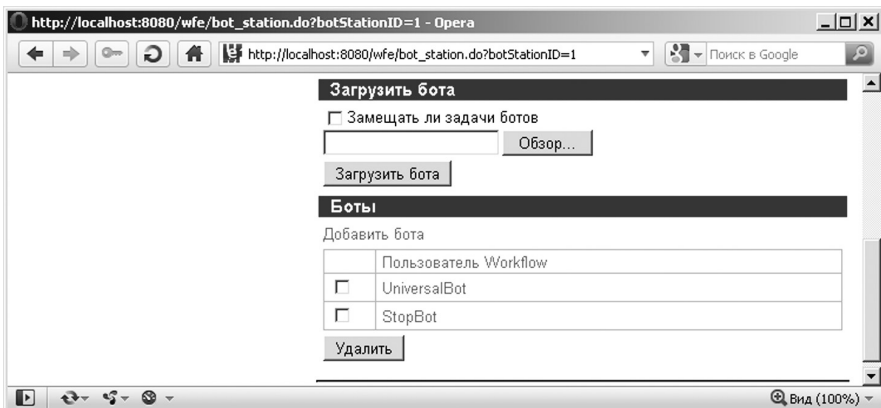


Рисунок 3.4.32

На странице добавления бота необходимо выбрать пользователя, чьи задания будет выполнять бот и указать пароль выбранного пользователя.

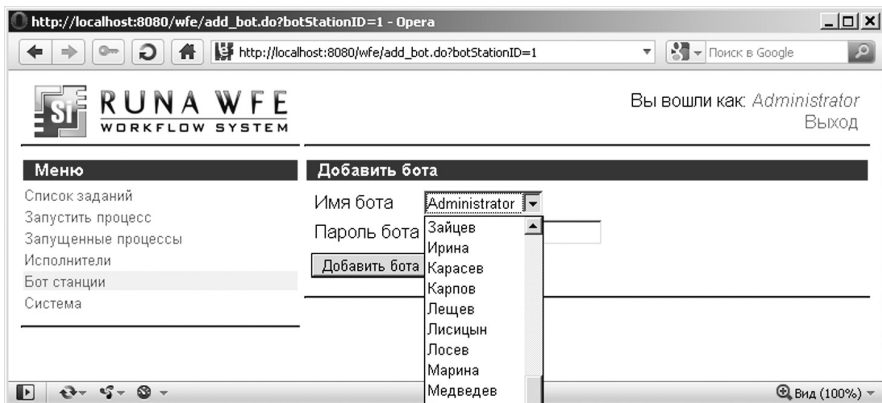


Рисунок 3.4.33

Удаление ботов

Для удаления бота из бот-станции перейдите на страницу с информацией по бот-станции и поставьте галочки напротив удаляемых ботов, после чего нажмите на кнопку «Удалить». Боты будут удалены вместе со всеми своими заданиями.

Изменение конфигурации обработчиков ботов

Перейдите на страницу информации бота, конфигурацию которого необходимо изменить. Изменение конфигурации возможно следующими путями:

- Загрузка предварительно написанного файла конфигурации с использованием кнопки «Browse» напротив задания, для которого меняется конфигурация
 - Ссылка «Редактировать» напротив задания, для которого меняется конфигурация позволит отредактировать текущую конфигурацию.
- Для корректной работы необходима поддержка javascript со стороны браузера

Если нужно удалить задание, то необходимо снять галочку напротив удаляемого задания и нажать на кнопку «Применить» в секции «Задания». После этого удаленное задание исчезнет из списка заданий для бота.

Если необходимо создать новое задание для бота, то необходимо перейти по ссылке «Добавить», после чего у бота появится новое пустое задание. Необходимо ввести Задание, Обработчик и конфигурацию для обработчика.

После нажатия на кнопку «Применить» новые параметры вступают в силу немедленно без перезапуска системы и будут использованы при очередном вызове ботов.

Как добавить задание для бота.

Сразу после создания бота у него не заведено никаких заданий (то есть не будет выполнять никакие задания, пришедшие выбранному пользователю). Поэтому после создания бота для него надо завести задания, которые он может выполнять.

Для добавления задания боту надо кликнуть на ссылку «Добавить» в свойствах бота.

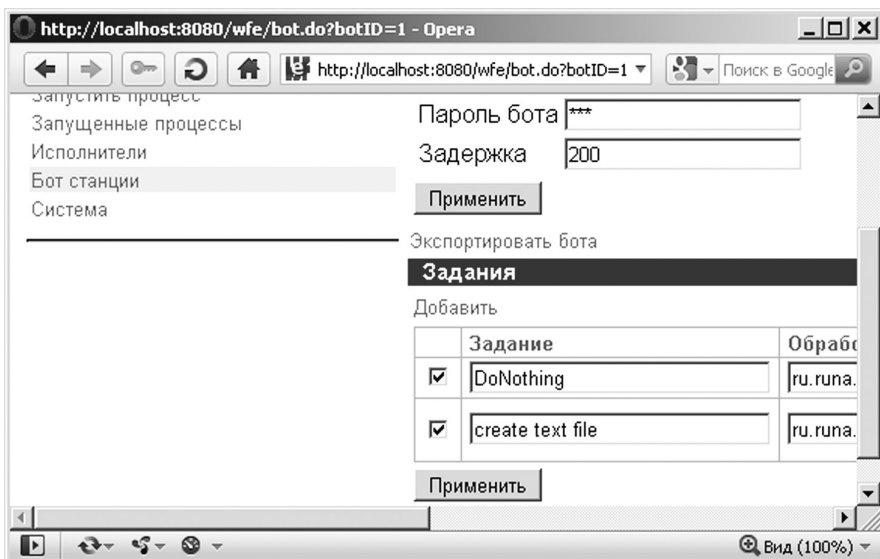


Рисунок 3.4.34

У задания есть имя задания, класс-обработчик задания и конфигурация. Имя задания — произвольная строка. Обработчик задания — Java-класс, реализующий определенный интерфейс. Класс-обработчик задания можно выбрать из выпадающего списка. Эти классы ищутся в wfe-bots.jar, а также в jar-файлах, указанных в специальном настройном файле. Конфигурация — это текстовый файл, содержащий настройки. Например, для SQL-бота в этом файле содержится текст запроса, а также соответствие параметров этого запроса переменным экземпляра бизнес-процесса.

В появившемся поле:

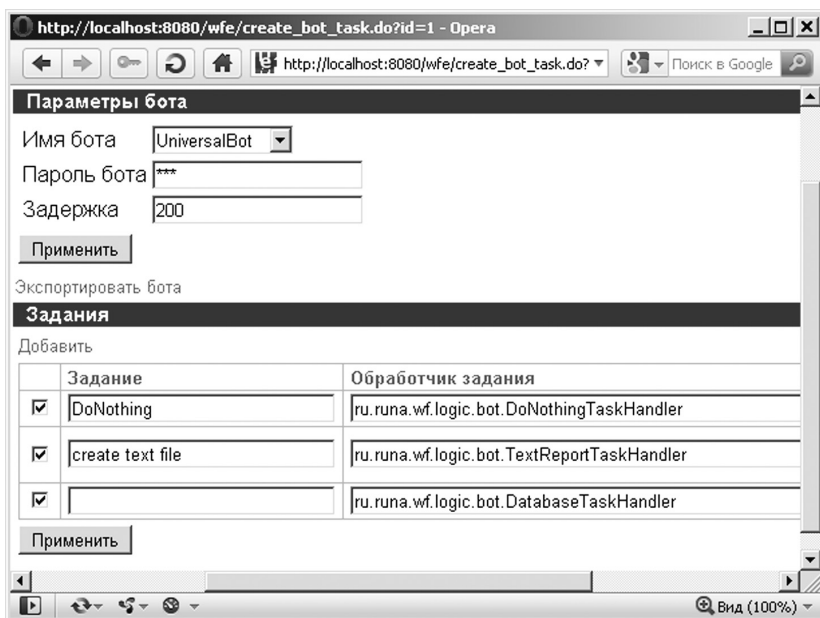


Рисунок 3.4.35

Надо ввести имя задания, которое в точности должно совпадать с именем узла-действия в создаваемом в редакторе процессе, в котором эта задача должна выполняться. Потом надо выбрать обработчик задания:

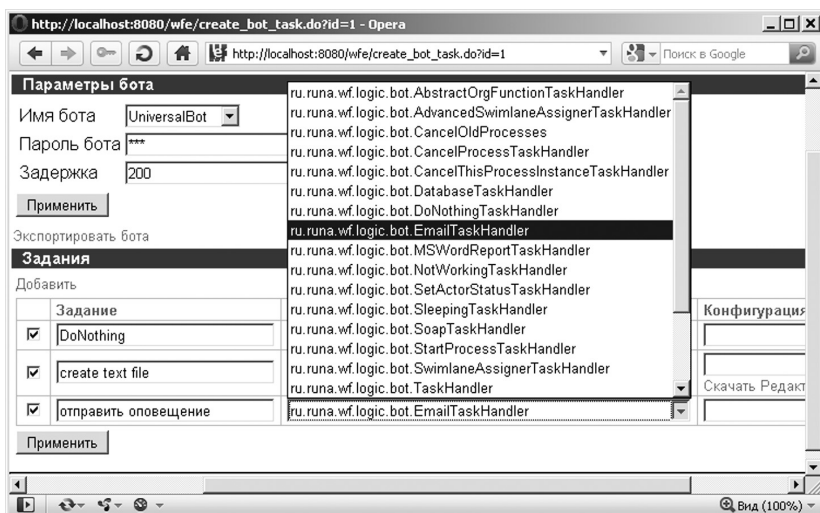


Рисунок 3.4.36

и ввести для этого задания конфигурацию из файла:

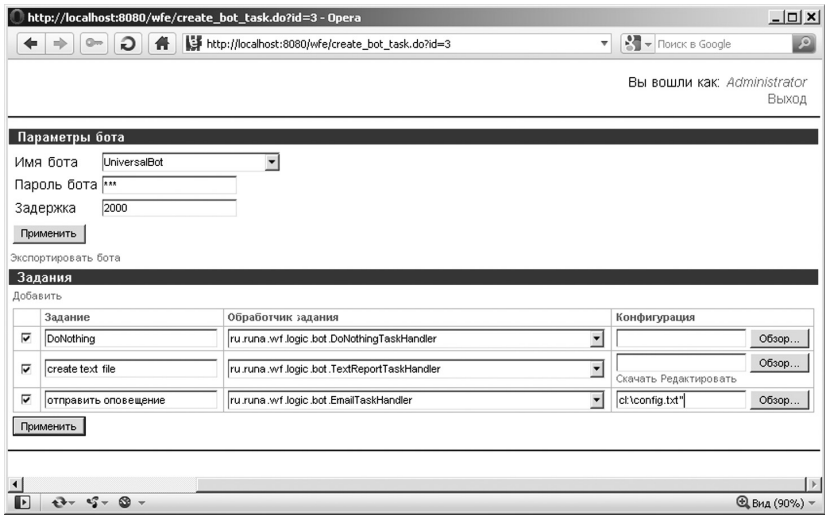


Рисунок 3.4.37

и кликнуть «применить».

Параметры задания задается в конфигурации:



Рисунок 3.4.38

3.4.10 Подпроцессы

В системе RunaWFE бизнес-процессы могут запускать другие бизнес-процессы в качестве подпроцессов. В графическом редакторе бизнес-процессов для запуска подпроцессов есть специальный элемент (со значком в правом нижнем углу в случае UML-нотации) — узел-подпроцесс, в котором можно настроить имя запускаемого процесса, а также соответствие передаваемых в подпроцесс значений переменных и значений, возвращаемых в родительский процесс.

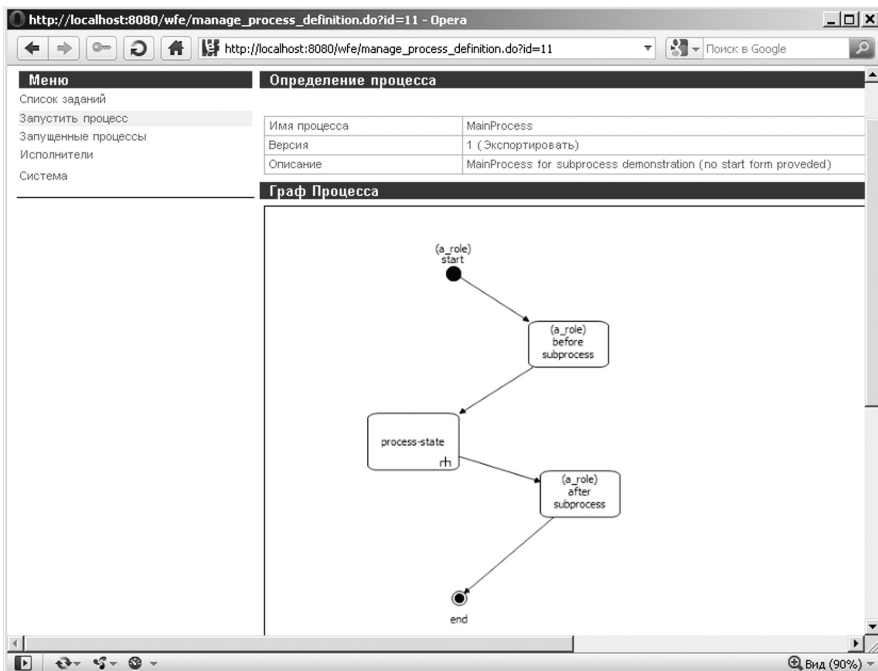


Рисунок 3.4.39

В момент прихода точки управления в узел-подпроцесс будет порожден и начнет выполняться экземпляр подпроцесса. Родительский процесс при этом будет ждать окончания подпроцесса и только после этого точка управления переместится в следующий узел.

В системе RunaWFE бизнес-процессы могут запустить другие бизнес-процессы и другим способом — при помощи бота. В задании боту надо указать обработчик, запускающий другой процесс, а в конфигурации — соответствие переменных процесса и подпроцесса. В этом случае родительский процесс не будет ждать окончания запущенного процесса, точка управления сразу переместится в следующий узел.

3.4.11 Описание системы безопасности.

Система безопасности состоит из субъектов, объектов и прав на действия: субъектам даются права на действия с определенными объектами.

Типы субъектов системы безопасности:

- Исполнители
- о Пользователи
- о Группы пользователей

Типы объектов системы безопасности:

- Система
- Исполнители
- о Пользователи
- о Группы пользователей
- Определения бизнес-процессов
- Экземпляры бизнес-процессов
- Бот-станции

Замечание. В системе безопасности присутствует только один объект типов «Система» и «Бот станция», ограничений на количество объектов других типов не предусмотрено.

Замечание. Права на действия для субъектов суммируются: Если исполнитель является членом какой-то группы пользователей, то исполнитель обладает правами как явно прописанными для этого субъекта (собственными правами), так и правами, которыми обладает группа, в которую он входит. Такие права называются унаследованными правами.

Типы прав на действия с объектами

Для всех объектов предусмотрены следующие типы прав на действия с объектами:

- Читать
- Изменять полномочия (права на объект)

Типы прав, зависящие от типов объектов

Объект «Система»:

- Входить в систему
- Создавать исполнителей
- Загружать определения бизнес-процессов

Объект «Определение бизнес-процесса»:

- Запускать бизнес-процесс
- Выгружать бизнес-процесс
- Изменять бизнес-процесс
- Читать экземпляр бизнес-процесса
- Останавливать экземпляр бизнес-процесса

Объект «Экземпляр бизнес-процесса»:

- Останавливать экземпляр бизнес-процесс

Объект «Исполнитель»:

- Изменять свойства исполнителя (имя, описание и т.п.)

Дополнительные права на пользователей:

- Изменять статус исполнителя (активен/не активен)

Дополнительные права на группы пользователей:

- Просматривать членов группы
- Добавлять в группу
- Удалять из группы

3.4.12 Работа с заданиями.

В системе RunaWFE деятельность предприятия представляется в виде множества бизнес-процессов. Пример графа бизнес-процесса «Отгул», который можно посмотреть, кликнув в меню «Запущенные процессы» на этот процесс:

Граф Процесса

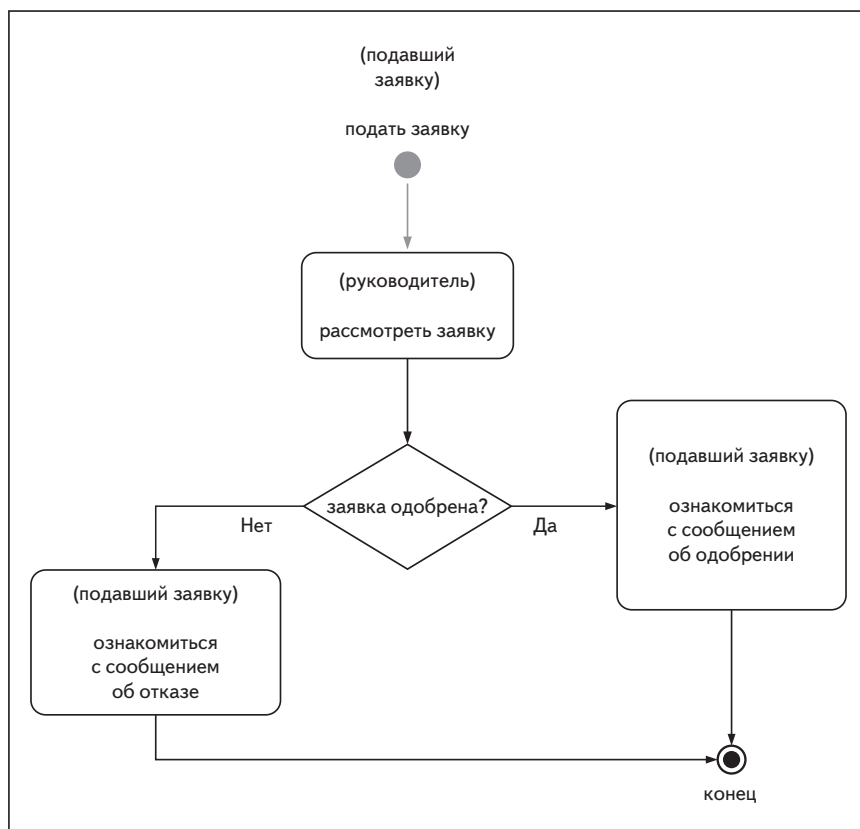


Рисунок 3.4.40

Здесь точка управления перешла в состояние «рассмотреть заявку». Соответствующий узел на графе выделен жирной рамкой. В момент перехода в это состояние исполнитель, в данном случае «руководитель», получает задание в списке заданий. При клике на задание будет отображена соответствующая ему форма. После реального выполнения задания сотрудник должен заполнить поля формы, предназначенные для ввода данных, кликнуть на командной кнопке «Задание исполнено»:

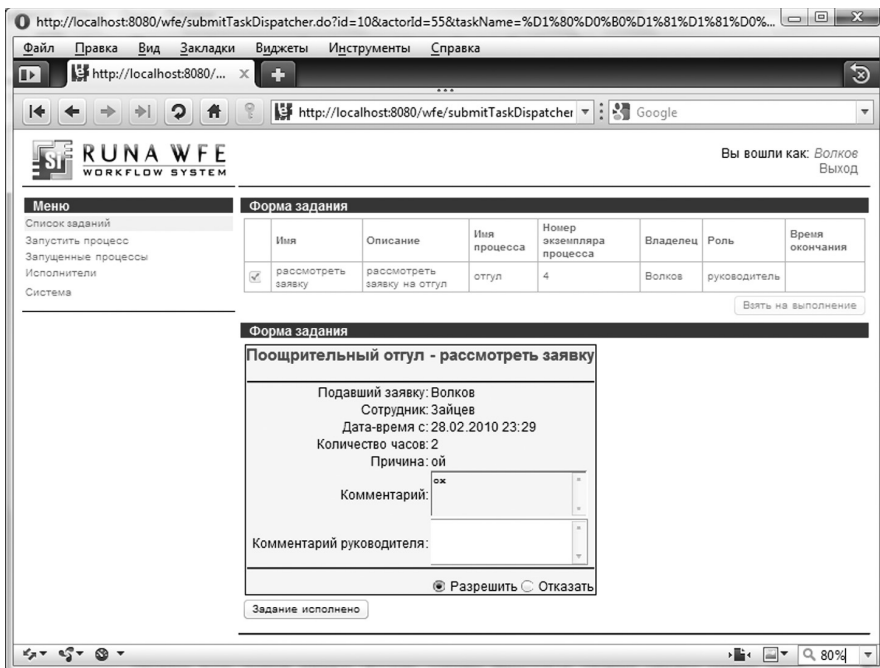


Рисунок 3.4.41

После того, как задание выполнено, точка управления переместится в следующее Действие бизнес-процесса.

Если задание соответствует группе пользователей, то это задание появится в списках заданий всех членов данной группы. Однако выполнить задание сможет только один пользователь — тот, который сделает это первым. Пользователь может не выполнять задание сразу, а взять его на выполнение, кликнув на команду «взять на выполнение». В этом случае задание пропадет в списках заданий всех остальных членов группы.

3.4.13 Система заместителей

В системе RunaWFE существует возможность, если исполнитель текущего задания не является «Активным» (например, болеет или находится в командировке) перенаправлять задания другим исполнителям. Для этого используется механизм правил замещения. Правила замещения последовательно просматриваются до тех пор, пока либо не будет найдено подходящее правило замещения, либо будет выяснено, что ни одного подходящего правила нет.

«Активность» пользователя устанавливается на странице «Исполнители». Нужно кликнуть по имени исполнителя. Под заголовком «Статус» снять галочку в строке «Активен» и кликнуть на кнопку «Применить», расположенную чуть ниже.

Замечание. Если пользователю не видно заголовка «Статус», значит он не обладает достаточными правами, и нужно обратиться к администратору. Механизм раздачи прав подробно описан в Руководстве администратора.

Рисунок 3.4.42

Правила замещения вводятся для каждого пользователя в его свойствах.

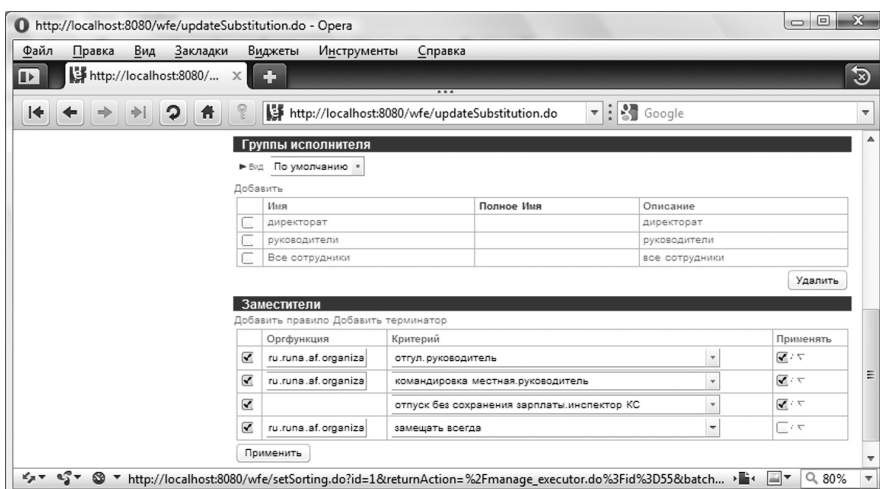


Рисунок 3.4.43

Правила замещения могут быть двух видов:

- Правило — определяет кем надо замещать Исполнителя в определенном случае
- Терминатор — определяет, что в определенном случае Исполнителя замещать не надо

На рисунке выше для пользователя применены три правила и один терминатор.

При задании правила необходимо определить функцию над оргструктурой и выбрать из списка критерий замещения.

Выбираемый критерий замещения либо соответствует бизнес-процессу и Роли-Дорожке, в которых будет происходить замещение, либо является критерием «замещать всегда».

При задании терминатора необходимо только выбрать из списка критерий замещения.

Правила замещения «просматриваются» последовательно сверху-вниз. Правила замещения, для которых не установлена галочка «Применить», игнорируются.

Для каждого правила замещения с установленной галочкой «Применить» проверяется следующее:

- Если выполнен критерий замещения и правило замещения является терминатором, то замещение исполнителя не производится и следующие правила не рассматриваются. В нашем примере третье правило замещения является терминатором, поэтому правило из четвертой строки рассматриваться не будет.
- Если выполнен критерий замещения, правило замещения не является терминатором и функция над оргструктурой вернула активного исполнителя, то этот исполнитель и будет использоваться в качестве заместителей, другие правила замещения рассматриваться не будут.

- Если оба предыдущих пункта не выполнены, то рассматривается следующее правило замещения (если текущее правило замещения оказалось последним в списке, то замещения не производится)

Замечание. При помощи треугольников столбца «применить» можно изменять положение правила замещения в списке. Сняв галочку в первом столбце и выполнив команду «применить», можно удалить правило замещения.

3.5 Графический редактор бизнес-процессов

RunaWFE содержит графический редактор, в котором можно создавать и изменять бизнес-процессы. Графический редактор запускается через меню операционной системы (Пуск / Программы / RunaWFE / Process designer) или кликом на иконку на рабочем столе.

Проиллюстрируем работу с графическим редактором бизнес-процессов при помощи описания последовательности действий пользователя по созданию нескольких демонстрационных бизнес-процессов.

3.5.1 Создание нового проекта процессов

Выберите пункт меню Файл > Создать > Новый проект. Появится мастер заведения нового проекта процессов.

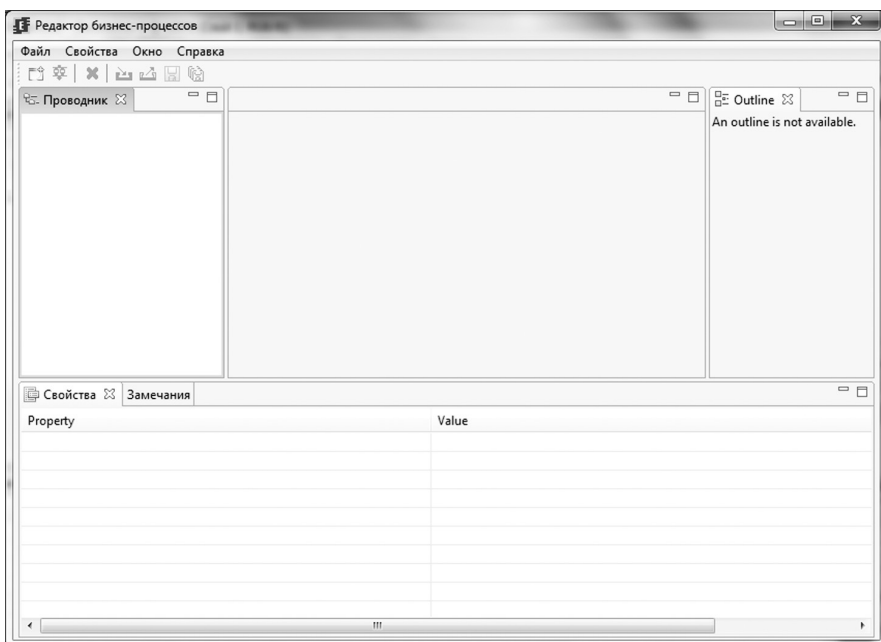


Рисунок 3.5.1

Правила замещения вводятся для каждого пользователя в его свойствах.

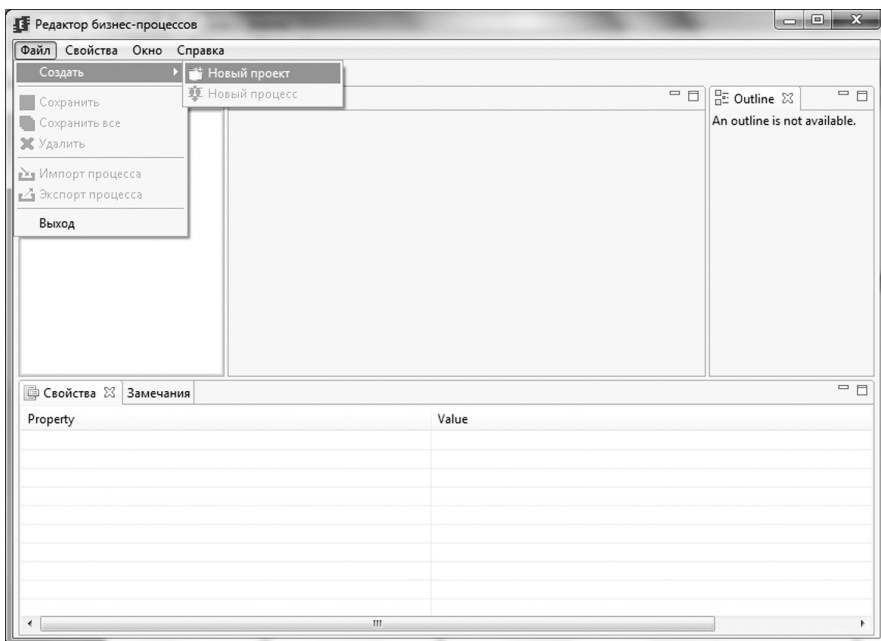


Рисунок 3.5.2

Правила замещения вводятся для каждого пользователя в его свойствах.

2. Введите имя проекта «HelloWorldProject».

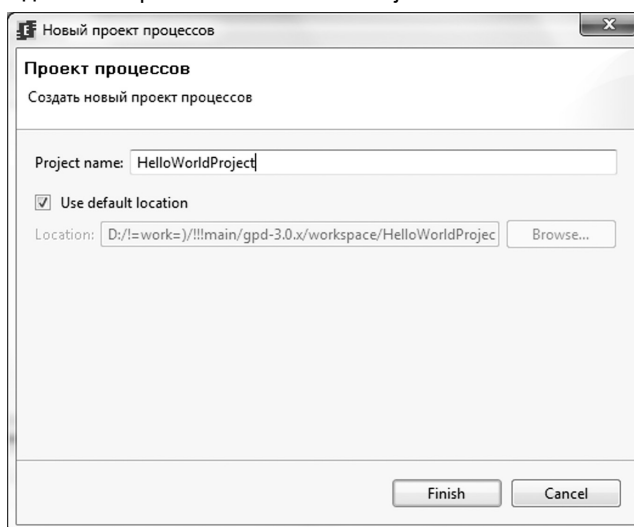


Рисунок 3.5.3

Правила замещения вводятся для каждого пользователя в его свойствах.

Будет создан проект HelloWorldProject.

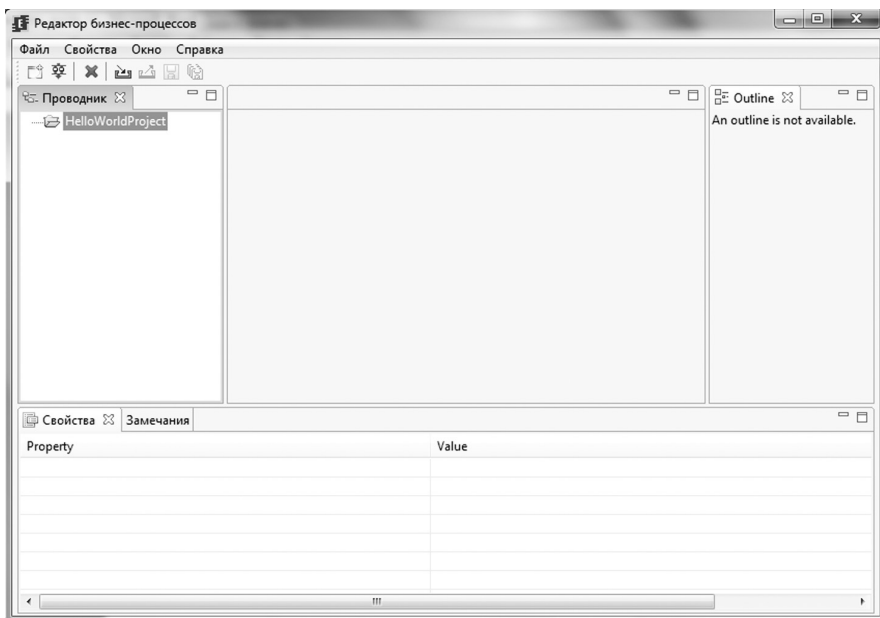


Рисунок 3.5.4

Создание процесса HelloWorldProcess

Сценарий

1. При запуске процесса появляется стартовая форма HelloWorld.
2. После клика на команду «запустить» процесс запускается и тут же переходит в состояние завершения.

Этот простейший бизнес-процесс состоит только из двух узлов: Start-state и Stop-state.

Разработка схемы бизнес-процесса

Откройте контекстное меню правой кнопкой мыши, кликнув на HelloWorldProject, затем кликните на «Новый процесс».

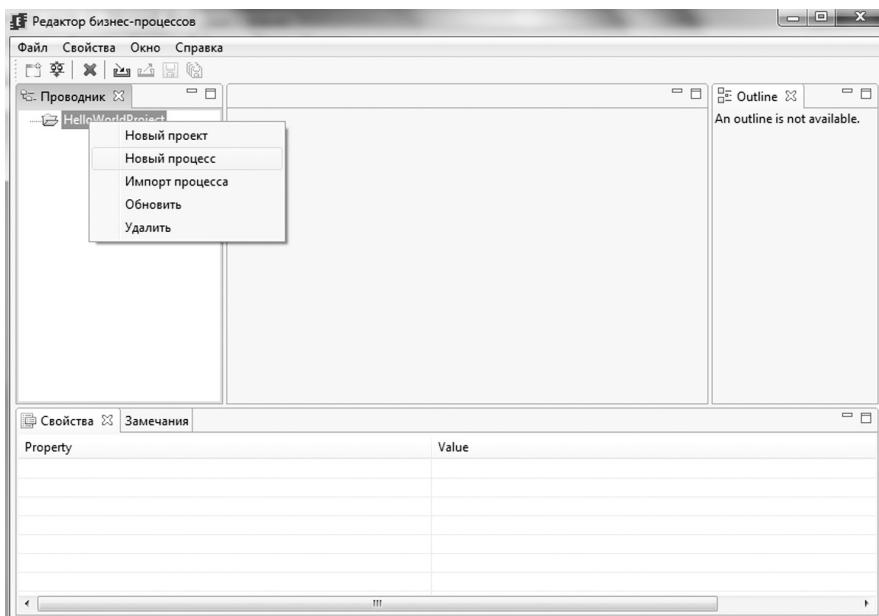


Рисунок 3.5.5

Введите HelloWorldProcess в качестве имени процесса:

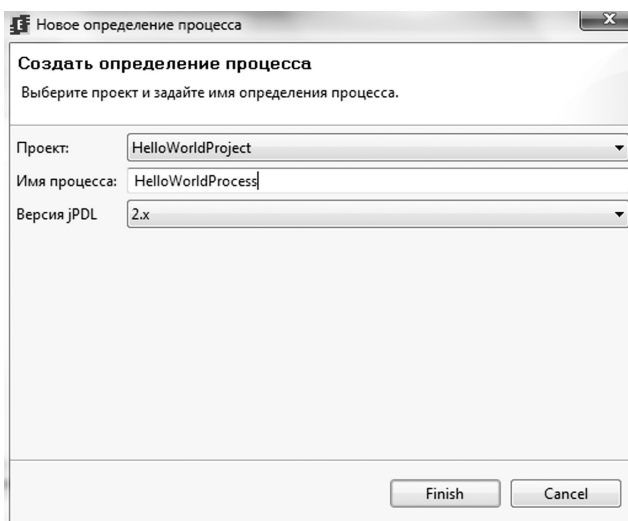


Рисунок 3.5.6

и кликните «Finish». Будет создан процесс HelloWorldProcess.

Щелкните двойным кликом на HelloWorldProcess. Откроется диаграмма процесса:

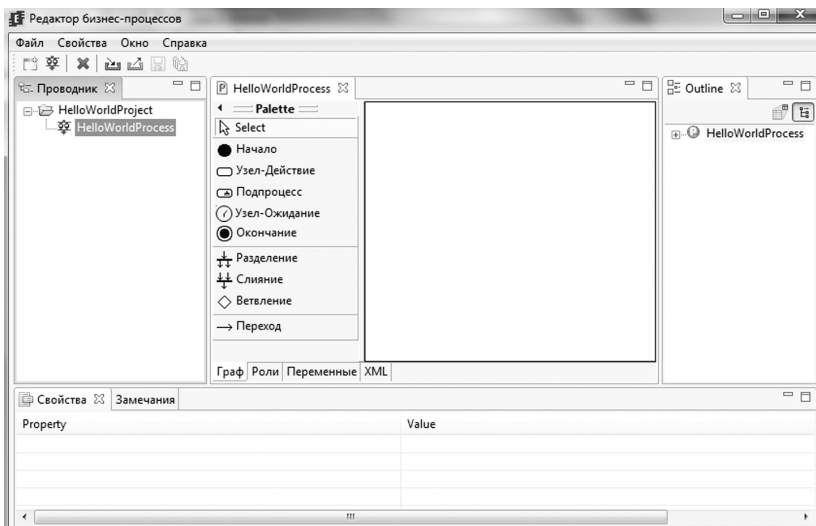


Рисунок 3.5.7

Кликните на свойства/показать сетку, появится сетка. Кликните на элементе «Начало» палитры, затем кликните на окне диаграммы. Стартовый узел процесса появится в окне диаграммы. Аналогично поместите на диаграмму завершающий узел процесса «Окончание», кликните на элемент «Переход» и соедините узлы «Начало» и «Окончание».

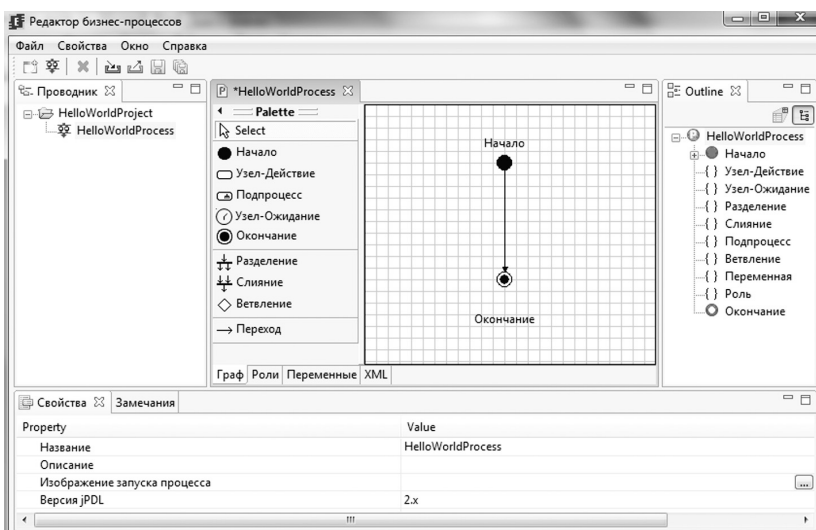


Рисунок 3.5.8

Кликните на свойства/показать сетку — сетка исчезнет. В соответствующих полях свойств введите краткое описание процесса и пиктограмму изображения процесса. Схема процесса готова.

3.5.2 Создание ролей

В RunaWFE роли являются специальным видом переменных бизнес-процесса. Каждому узлу-действию в бизнес-процессе должна быть поставлена в соответствие роль. Начальный узел процесса имеет отличное от остальных узлов поведение в части инициализации ролей. Начальный узел не использует инициализатор. Роль, указанная в начальном узле, инициализируется пользователем, запустившим бизнес-процесс. В отличие от этого остальные узлы-действия используют роли для того, чтобы определить, кто может выполнять задание в данном узле-действии.

Задание роли для процесса HelloWorldProcess

Процесс состоит только из двух узлов: Начало и Окончание. Для состояния окончания процесса роль не нужна. Поэтому в процессе будет только одна роль.

Выберите вкладку «Роли». Появится следующая форма:

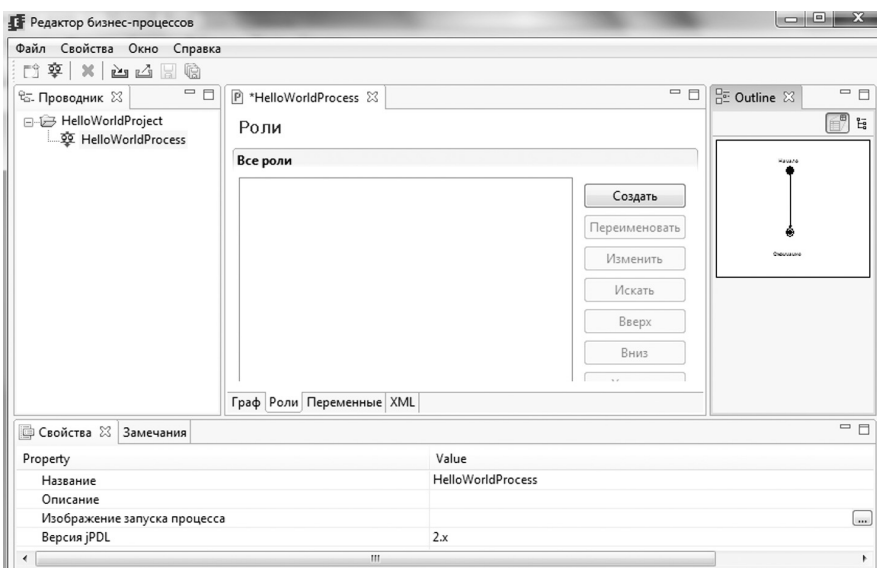


Рисунок 3.5.9

Кликните на кнопке «Создать». В появившейся форме введите «requester».

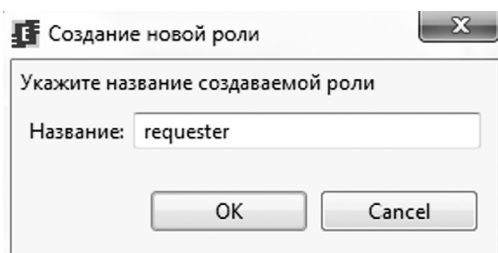


Рисунок 3.5.10

Кликните на ОК.

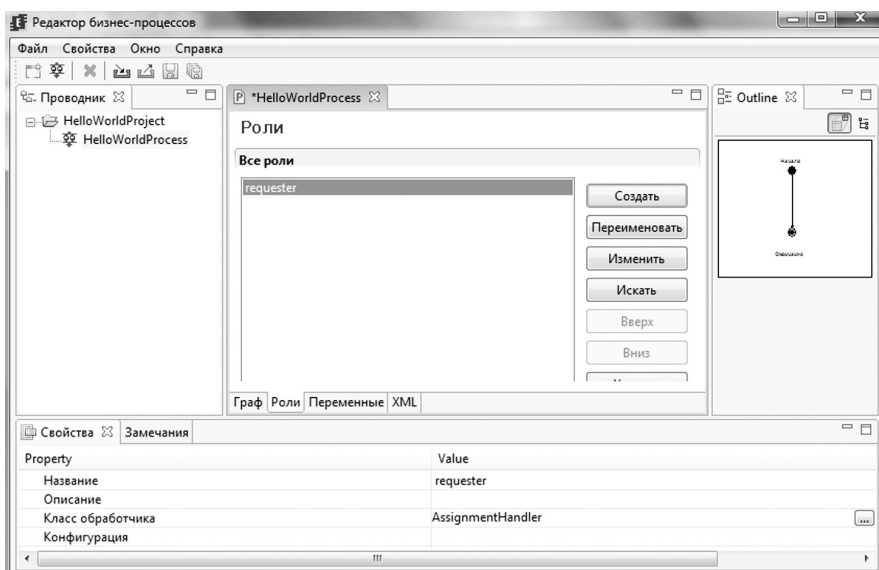


Рисунок 3.5.11

Кликните на закладке «Граф». На графе процесса кликните на узле «Начало», в свойствах узла кликните на поле, находящееся на пересечении Роль-Value, откройте появившейся список и выберите в нем строку requester:

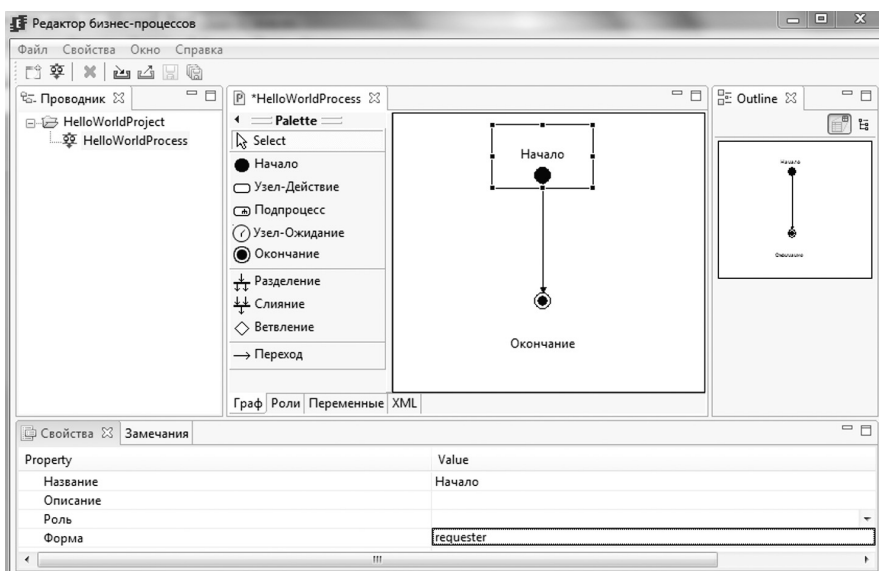


Рисунок 3.5.12

Имя роли появится в круглых скобках над названием узла процесса:

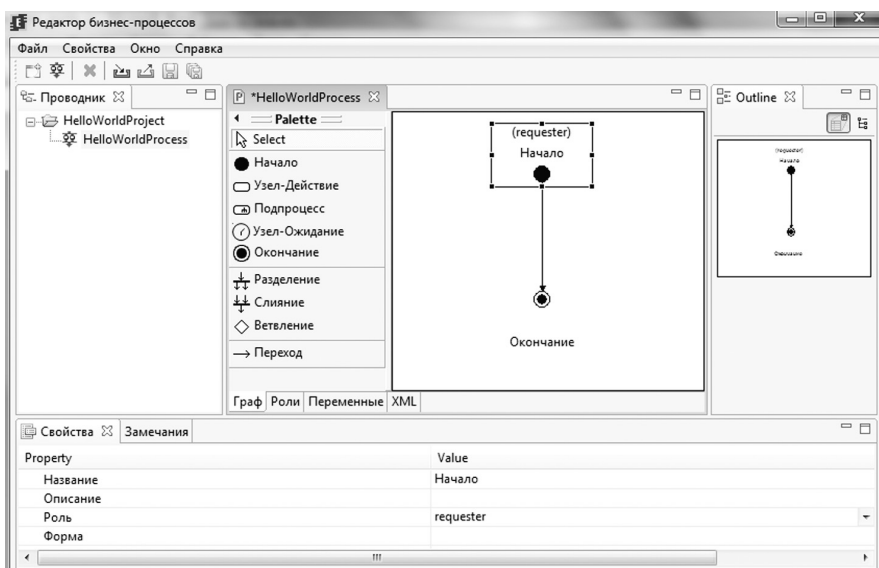


Рисунок 3.5.13

3.5.3 Создание графических форм

У процесса HelloWorldProcess нет переменных и есть только одна форма — стартовая форма. Кликните на узел «start» графа бизнес-процесса правой кнопкой мыши и выберите команду «Форма» → «Создать форму» (левой кнопкой мыши):

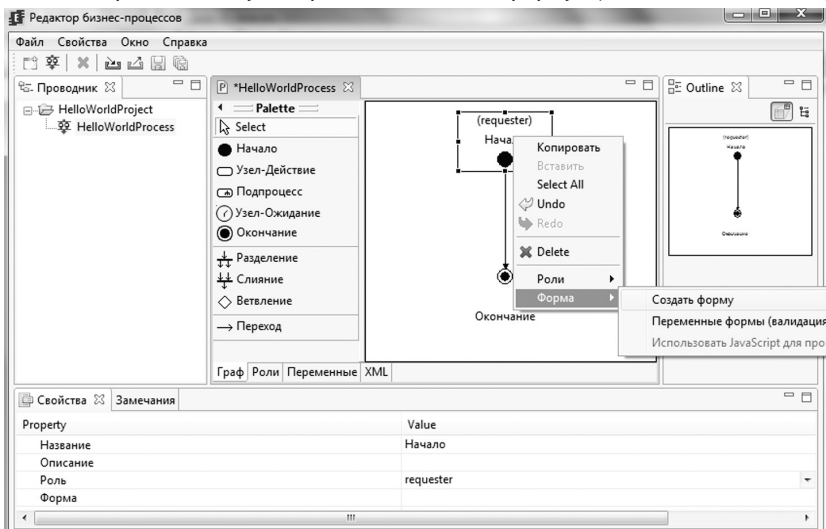


Рисунок 3.5.14

В появившемся выпадающем списке выберите тип формы «ftl».

После появления окна редактирования кликните на вкладку дизайн:

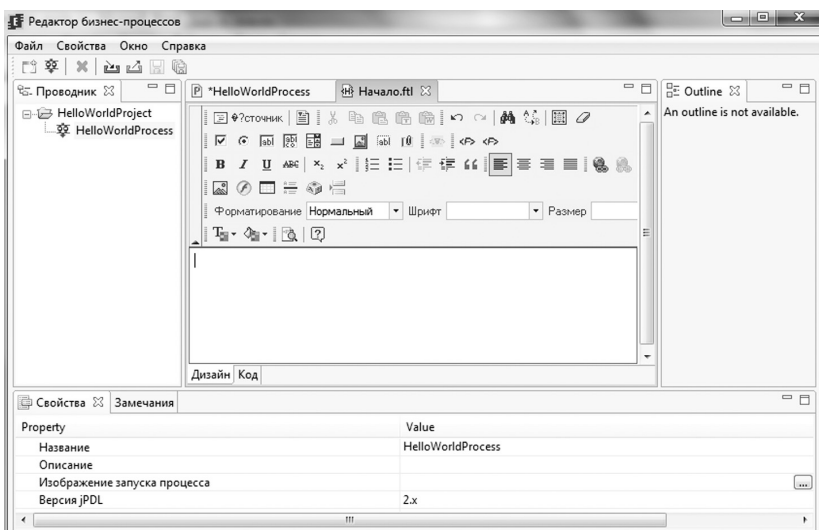


Рисунок 3.5.15

В окне конструктора форм введите Hello World!
 Выберите соответствующие шрифт, размер, положение на странице:

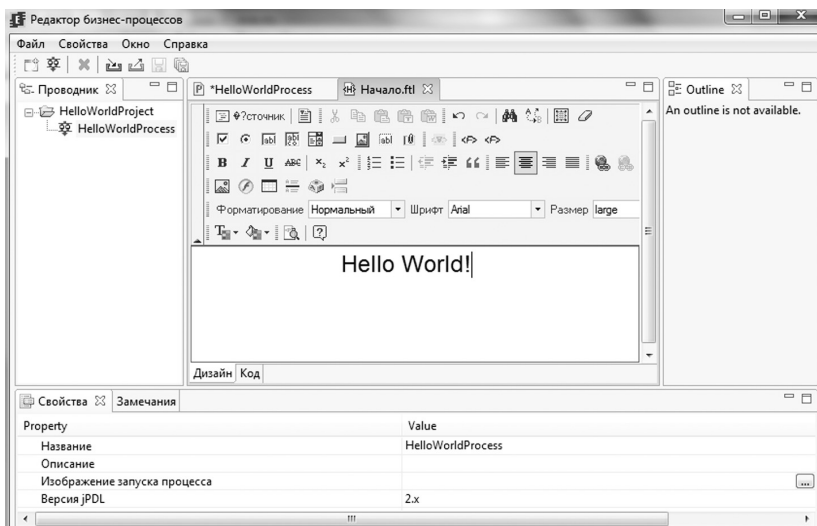


Рисунок 3.5.16

Закройте вкладку формы, подтвердив ее сохранение в диалоге.

3.5.4 Создание файла-архива бизнес-процесса

Выберите HelloWorldProcess, выполните команду Файл/Экспорт.

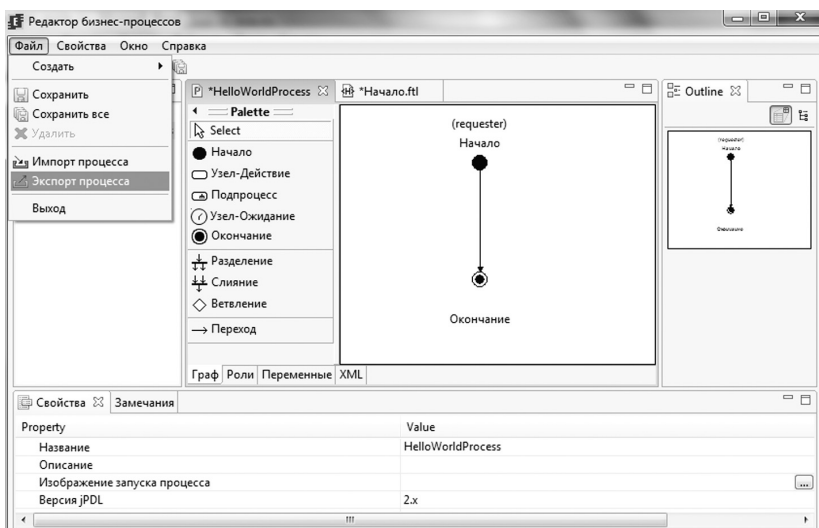


Рисунок 3.5.17

В появившейся форме нажмите кнопку «Browse...», укажите каталог и введите имя файла архива бизнес-процесса:

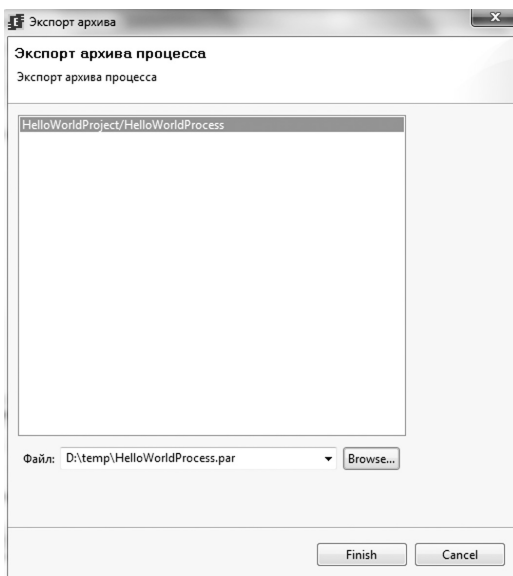


Рисунок 3.5.18

Выполните «finish». В файловой системе будет сформирован файл-архив бизнес-процесса HelloWorldProcess: HelloWorldProcess.par.

3.5.5 Загрузка бизнес-процесса в систему

Войдите в систему управления бизнес-процессами RunaWFE как Administrator.

Войдите в меню «Запустить процесс».

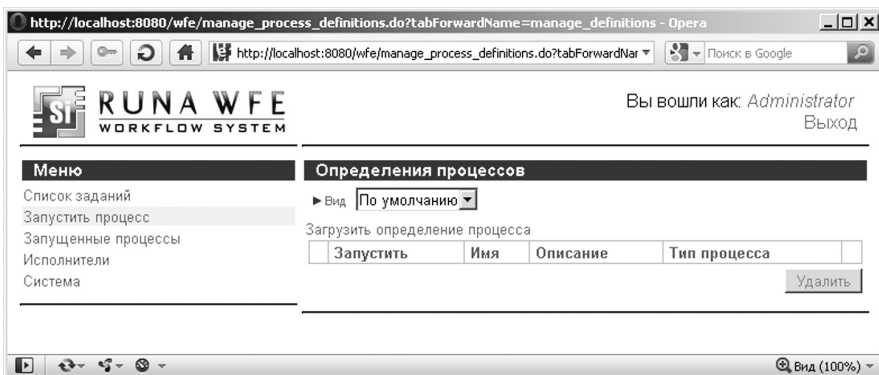


Рисунок 3.5.19

Кликните на ссылку «Загрузить определение процесса»⁶. В появившемся диалоге выберите в файловой системе сформированный в редакторе файл-архив бизнес-процесса — HelloWorldProcess.par, впишите тип процесса — «демо» и нажмите «выполнить».

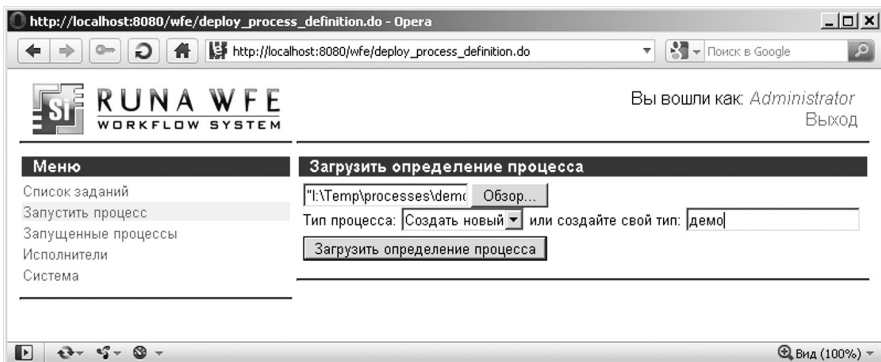


Рисунок 3.5.20

Процесс HelloWorldProcess будет загружен в систему:

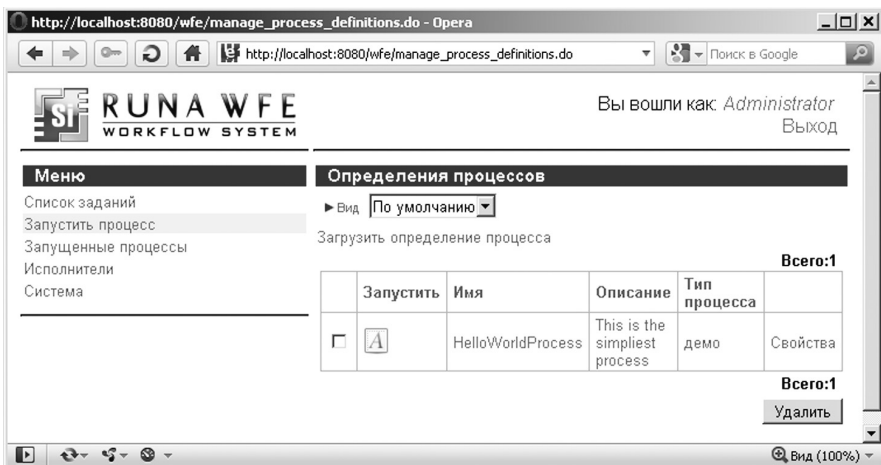


Рисунок 3.5.21

3.5.6 Выполнение процесса

Кликните на имя процесса. Вы увидите стартовую форму:

⁶ Замечание: Для того, чтобы загрузить определение бизнес-процесса в систему у вас должны быть права «Загружать определение процесса» (могут быть даны в меню система).

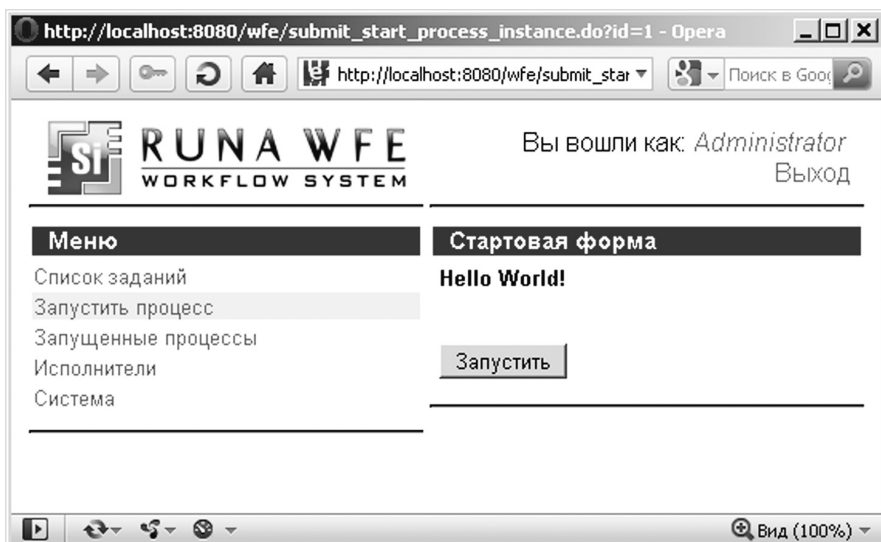


Рисунок 3.5.22

Кликните на команду «Запустить». Процесс будет запущен и тут же завершится. Можно будет посмотреть экземпляр процесса в меню «Экземпляры процессов»:

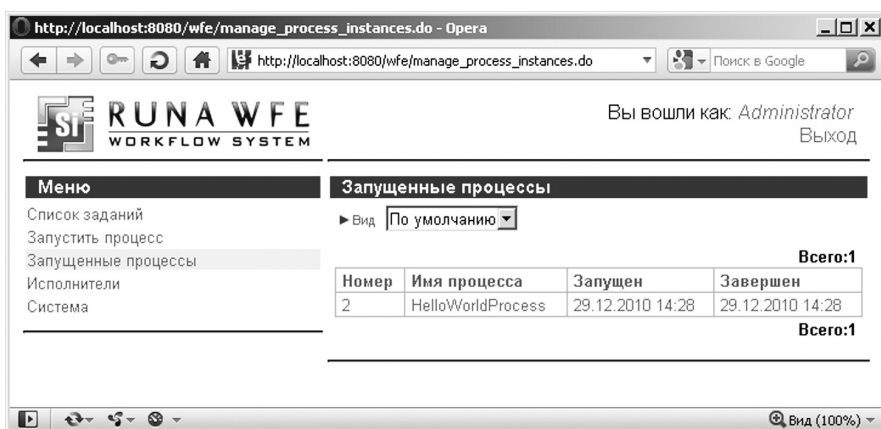


Рисунок 3.5.23

Кликните на экземпляр процесса — вы увидите свойства процесса:

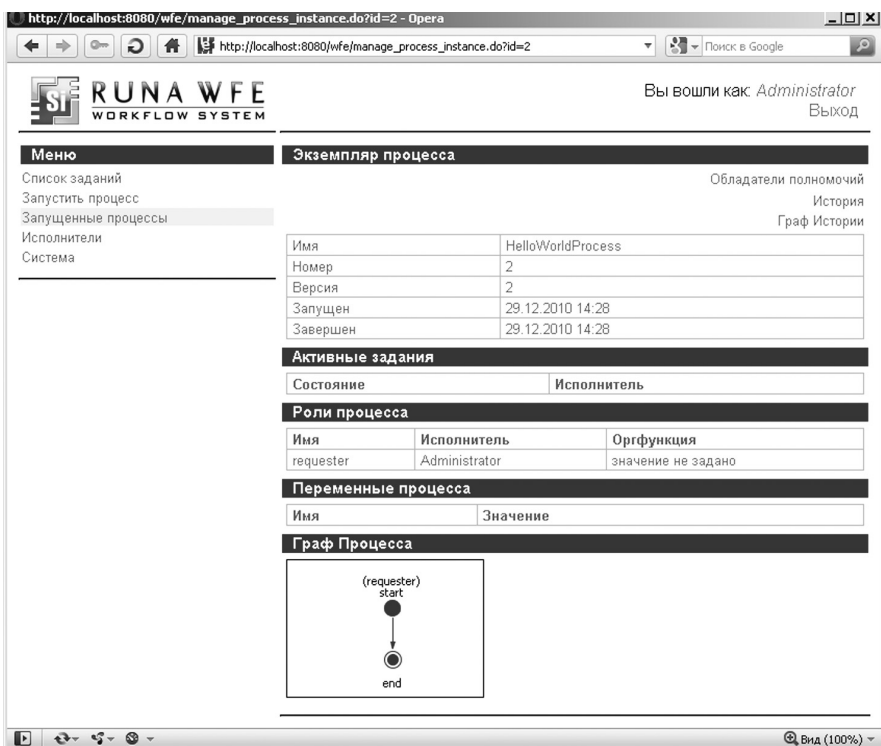


Рисунок 3.5.24

Создание процесса «Overtime Work»

Рассмотрим более сложные конструкции графического редактора бизнес-процессов на примере разработки процесса «Overtime Work» (сверхурочные работы).

Сценарий процесса

Руководитель предлагает сотруднику выйти на сверхурочную работу. Сотрудник соглашается или отказывается. Далее руководитель знакомится с решением сотрудника.

Предполагается, что руководитель является членом группы «manager», а сотрудник является членом группы «staff».

Разработка графа бизнес-процесса

Создайте новый проект «OvertimeWork», затем кликните левой кнопкой мыши на команду «Новый процесс». Введите в появившемся диалоге «Overtime Work» в качестве имени процесса. Сделайте двойной клик на появившейся строке «Overtime Work». Появится окно диаграммы бизнес-процесса. Выбирая элементы палитры, нарисуйте следующий граф бизнес-процесса:

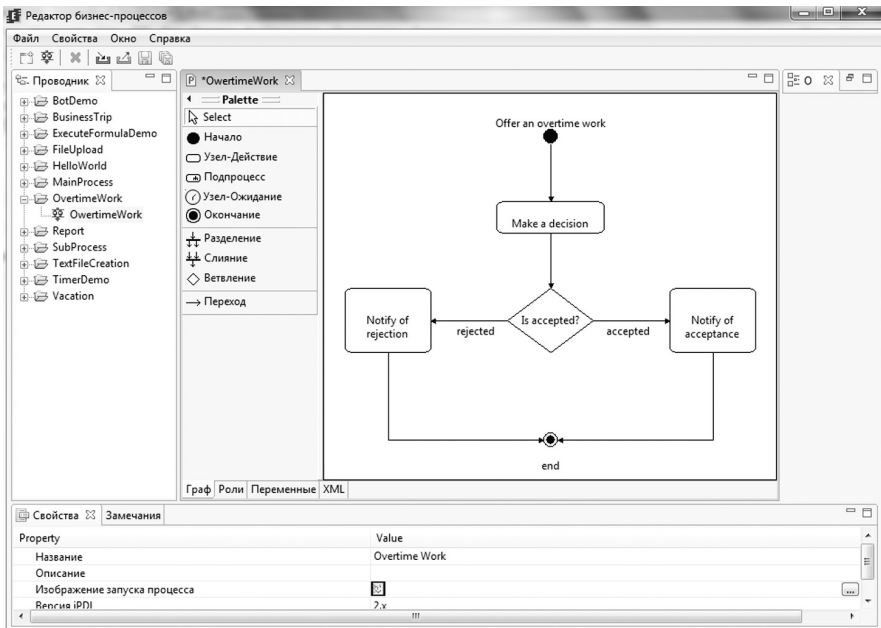


Рисунок 3.5.25

Для задания подписи элемента, отличной от подписи по умолчанию, выделите элемент кликнув на него, и еще раз кликните на элементе.

Для переходов, выходящих из узла исключающего выбора «Is accepted?» в их свойствах задайте явные имена «accepted» и «rejected»:

Создание ролей Роли

Поведение ролей в начальном узле бизнес-процесса отличается от поведения в остальных узлах. В начальном узле бизнес-процесса роль, поставленная в соответствие начальному узлу, инициализируется пользователем, запустившим процесс. Во всех других узлах роль инициализируется при помощи выбранного в редакторе бизнес-процессов инициализатора.

Роль (или роль-дорожка) — это специальный тип переменной бизнес-процесса. Роль может быть инициализирована пользователем или группой пользователей в любой момент выполнения бизнес-процесса. К моменту перехода управления в некоторый узел роль, связанная с этим узлом, должна быть инициализирована. Если роль инициализирована пользователем, то после прихода управления в этот узел только этот пользователь получит задание. Если роль инициализирована группой пользователей, то после прихода управления в этот узел все члены группы получают задание. Однако выполнить задание сможет только один член группы, — тот, который первым кликнет на «выполнить».

Для автоматической инициализации роли в момент прихода управление в узел-действие служит специальный механизм инициализации, основанный на функциях над организационной структурой, которые возвращают пользователя или группу пользователей. В системе можно определить набор таких функций. Эти функции также могут зависеть от параметров. В частности параметром может быть переменная бизнес-процесса.

Также можно явно инициализировать роль некоторым значением, например, присвоить значение роли как переменной бизнес-процесса через графическую форму.

«Внутри» бизнес-процесса функции над организационной структурой (далее — оргфункции) должны быть определены в соответствии с используемым в системе RunaWFE языком jPdl.

Если параметрами оргфункций являются значения переменных бизнес-процесса, то имена этих переменных должны быть «окружены» следующей конструкцией: `${<имя переменной>}`.

Задание ролей для процесса «Overtime Work».

В бизнес-процессе есть две роли:

- manager
- staff

Описание ролей:

Таблица 3.5.1

Роль	Описание
Manager	Руководитель, который запускает процесс
Staff	Сотрудник, которого руководитель выбирает в стартовой форме и которому направляется предложение о сверхурочной работе

Связь узлов графа бизнес-процесса и ролей:

Таблица 3.5.2

Узел	Роль
Offer an overtime work (предложить сверхурочные работы)	manager
Make a decision (принять решение)	staff
Notify for declining (ознакомиться с сообщением об отклонении)	manager
Notify for acceptance (ознакомиться с сообщением о согласии)	manager

Создание ролей

Кликните на вкладку «роли».

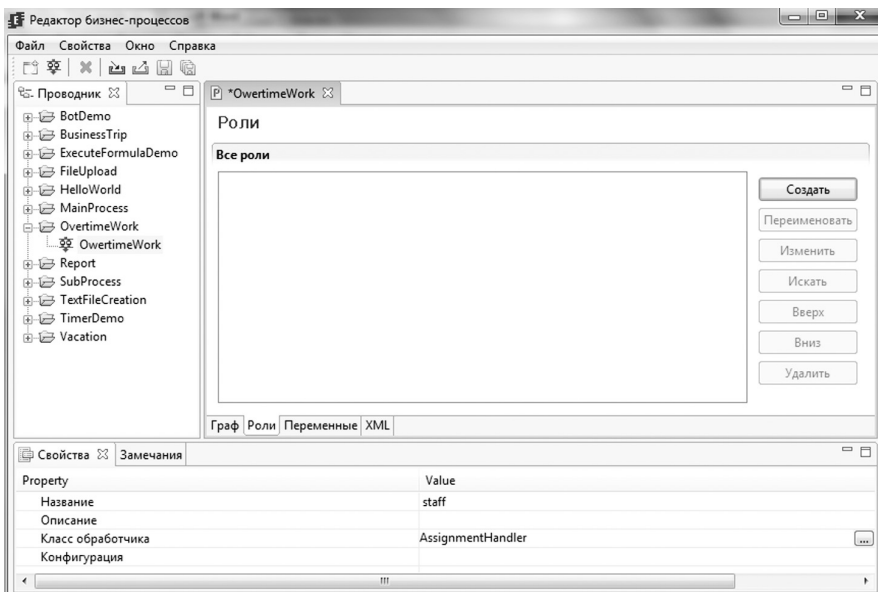


Рисунок 3.5.26

На появившейся странице кликните на кнопку «Создать».

В появившейся форме введите название роли «manager» и кликните на «ОК», т.к. данная роль-дорожка будет использована в стартовом состоянии — т.е. инициализирована пользователем, запустившим процесс. Поэтому роли-дорожке manager не требуется инициализатор.

Аналогично добавьте роль «staff».

Откройте опять вкладку «Граф», выберите стартовый узел-действие «Offer an overtime work». В свойствах кликните на правую часть поля Роль/Значение. В появившемся списке выберите «manager».

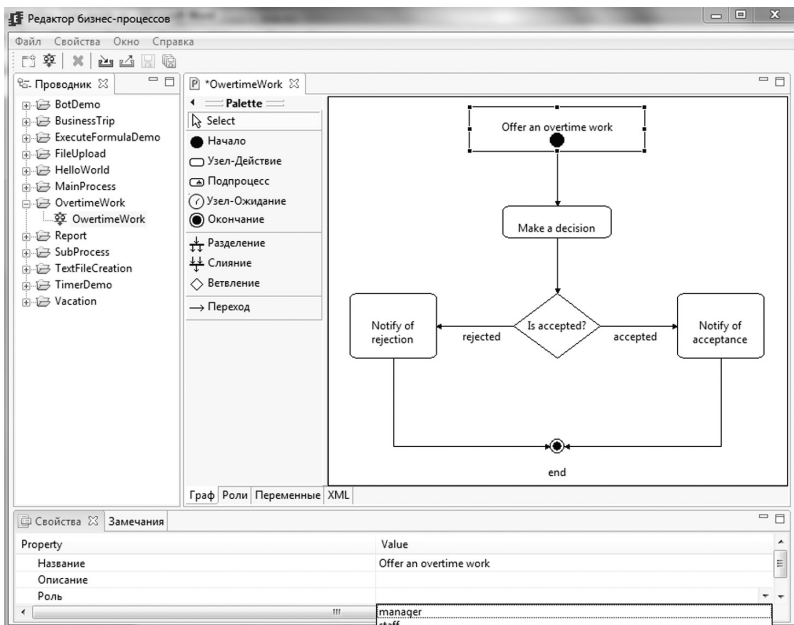


Рисунок 3.5.27

Аналогично задайте роли во всех остальных узлах-действиях.

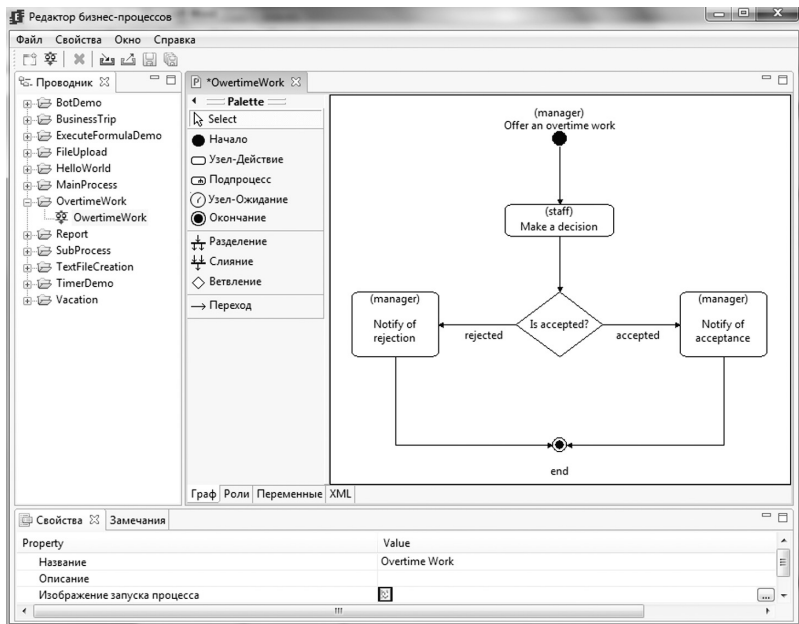


Рисунок 3.5.28

3.5.7 Создание переменных

Описание и инициализация переменных

В бизнес-процессе используются следующие переменные:

Таблица 3.5.3

Переменная	Тип	Описание
Since	Дата-время	Дата-время начала сверхурочных
Till	Дата-время	Дата-время окончания сверхурочных
Reason	Строка	Причина
Comment	Многострочный текст	Комментарий
staffPersonDecision	Логический	Решение сотрудника
staffPersonComment	Многострочный текст	Комментарий сотрудника

Переменные

- since
- till
- reason
- comment
- staff⁷

должны быть проинициализированы значениями в стартовом узле-действии «Offer an overtime work».

Переменные

- staffPersonDecision
- staffPersonComment

должны быть проинициализированы в узле «Make a decision»

⁷ Несмотря на то, что Staff является ролью-дорожкой, ее можно инициализировать как обычную переменную.

Создание переменных

Кликните на вкладке «Переменные». Кликните на кнопке «Создать»

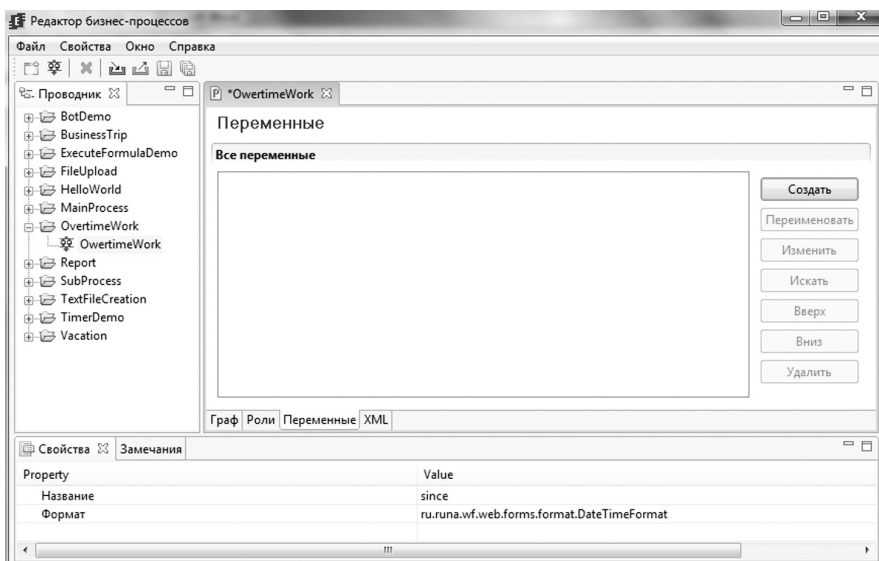


Рисунок 3.5.29

Введите название переменной — since. В качестве формата выберите DateTimeFormat.

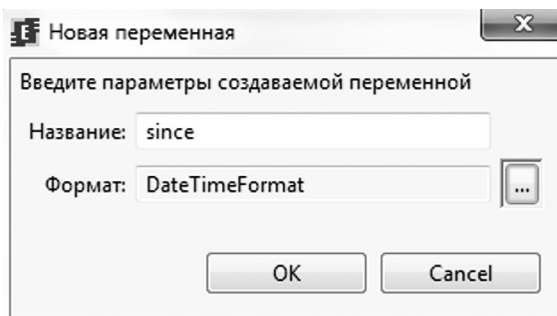


Рисунок 3.5.30

Аналогично заведите все остальные переменные бизнес-процесса.

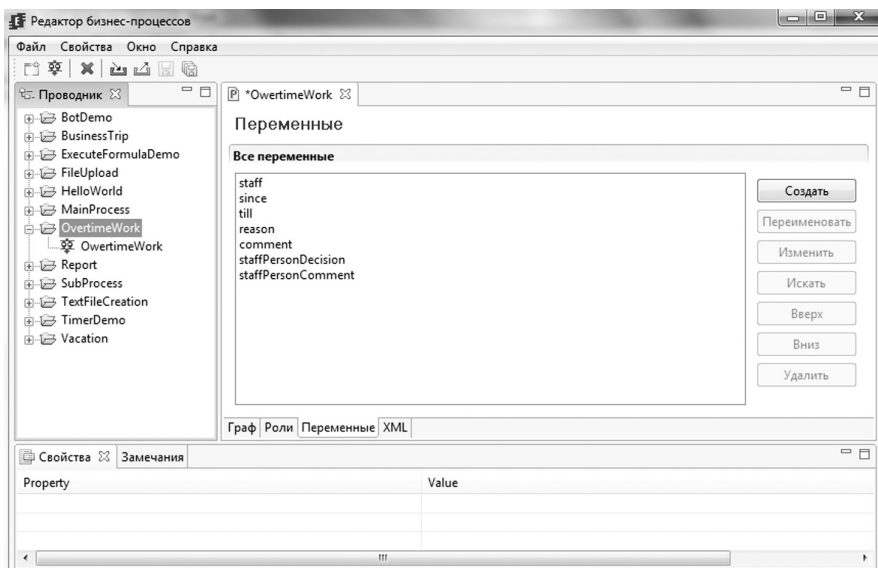


Рисунок 3.5.31

Задание формулы для элемента «Исключающий выбор»

Описание формулы

В случае, если значение переменной `staffPersonDecision` (решение сотрудника) — «истина», точка управления должна перейти в узел-действие «Notify for acceptance» (ознакомиться с сообщением о согласии). В противном случае точка управления должна перейти в узел-действие «Notify for declining» (ознакомиться с сообщением об отклонении).

Создание формулы

Выберите закладку «Граф», кликните на узел исключающего выбора «Is accepted?». В свойствах узла выберите поле, находящееся на пересечении Класс-Value. Кликните на командную кнопку, появившуюся в правой части этого поля. Откроется форма выбора обработчика для решения:



Рисунок 3.5.32

Выберите BSFDecisionHandler и нажмите ОК. Название обработчика появится в свойствах узла.

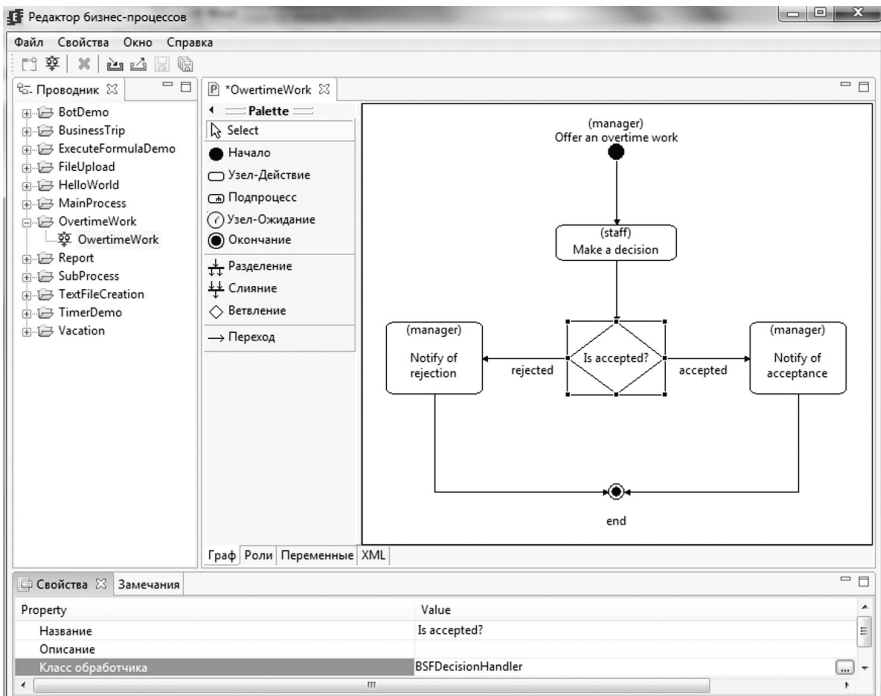


Рисунок 3.5.33

Правой кнопкой мыши кликните на элементе и выберите пункт «Изменить».

В появившейся форме для перехода accepted выберите переменную staffPersonDecision, установите операцию сравнения «равно» и значение «true». Для перехода rejected выберите переменную staffPersonDecision, установите операцию сравнения «не равно» и значение «true». Для «если ни одно из условий не выполнено» установите переход «rejected»:

Редактор Bean Shell

Конструктор код BSH

accepted	staffPersonDecision	равно	true
rejected	staffPersonDecision	не равно	true

Если ни одно из условий не выполнено: accepted

OK Cancel

Рисунок 3.5.34

Кликните «OK». Формула для узла исключающего выбора задана.

Создание графических форм

Узлам-действиям, исполнителями заданий в которых являются сотрудники (не боты) должны быть поставлены в соответствие формы. Каждая форма записывается в отдельном файле. Формы задаются в графическом конструкторе форм.

Кликните на начальный узел «Offer an overtime work» графа бизнес-процесса правой кнопкой мыши и выберите команду «Форма» → «Создать форму» (левой кнопкой мыши):

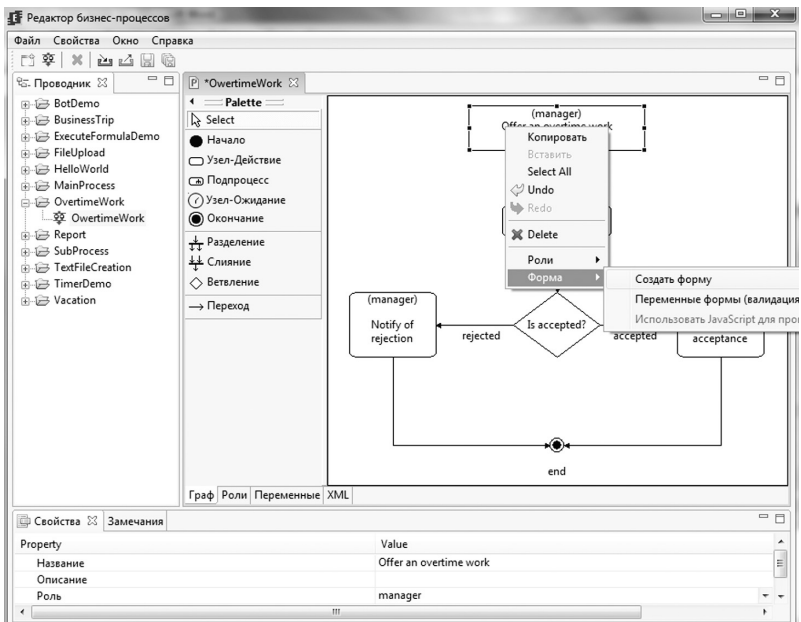


Рисунок 3.5.35

Выберите тип формы «html». Появится следующий интерфейс:

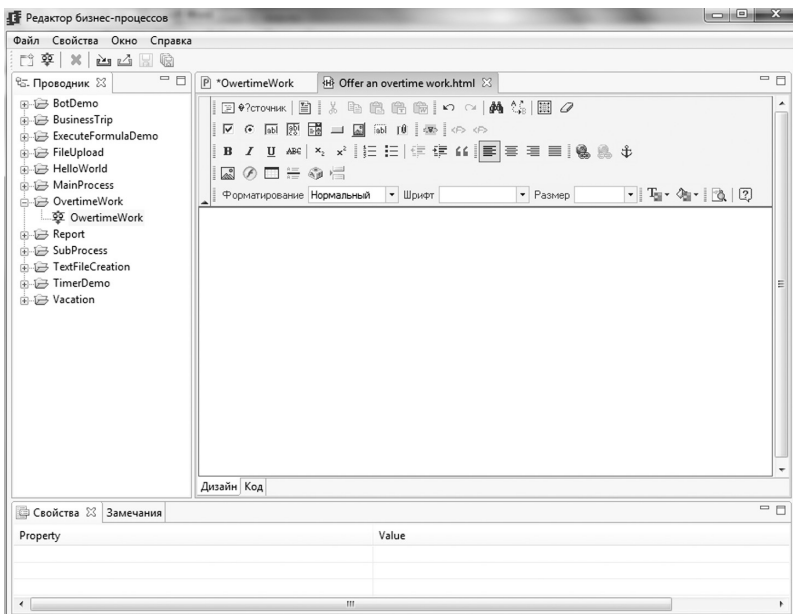


Рисунок 3.5.36

Введите текст «Offer an overtime work», на следующей строке выберите «Employee» и кликните на элемент . В появившейся форме в качестве названия элемента формы введите «staff», в качестве элемента формы выберите «Члены группы»:

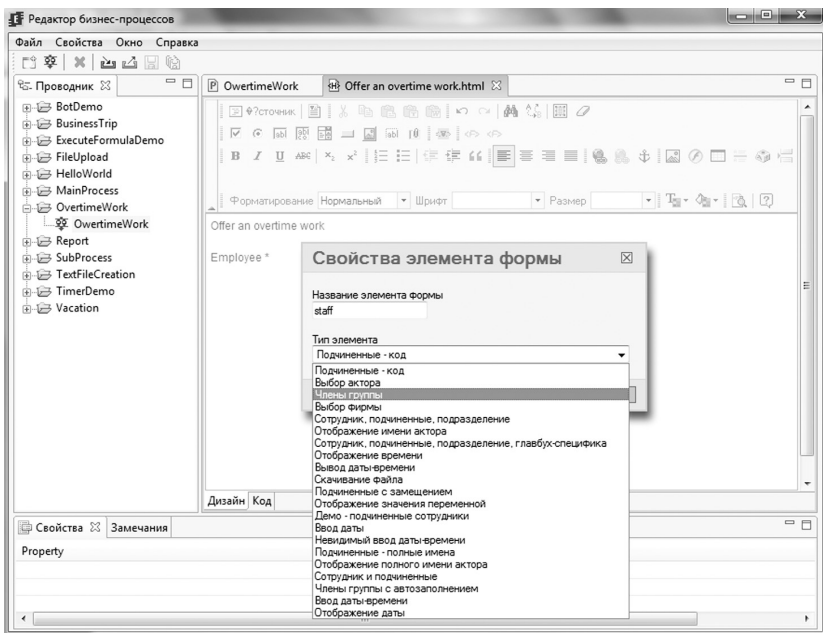


Рисунок 3.5.37

Кликните на «ОК». На форме появится элемент «ФИО СОТРУДНИКА»:

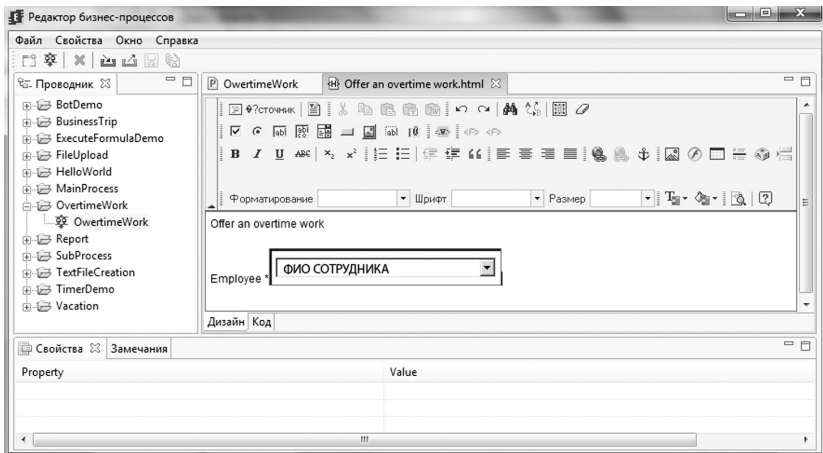


Рисунок 3.5.38

Аналогичным образом создайте элементы для ввода других переменных бизнес-процесса:

- since
- till
- reason
- comment

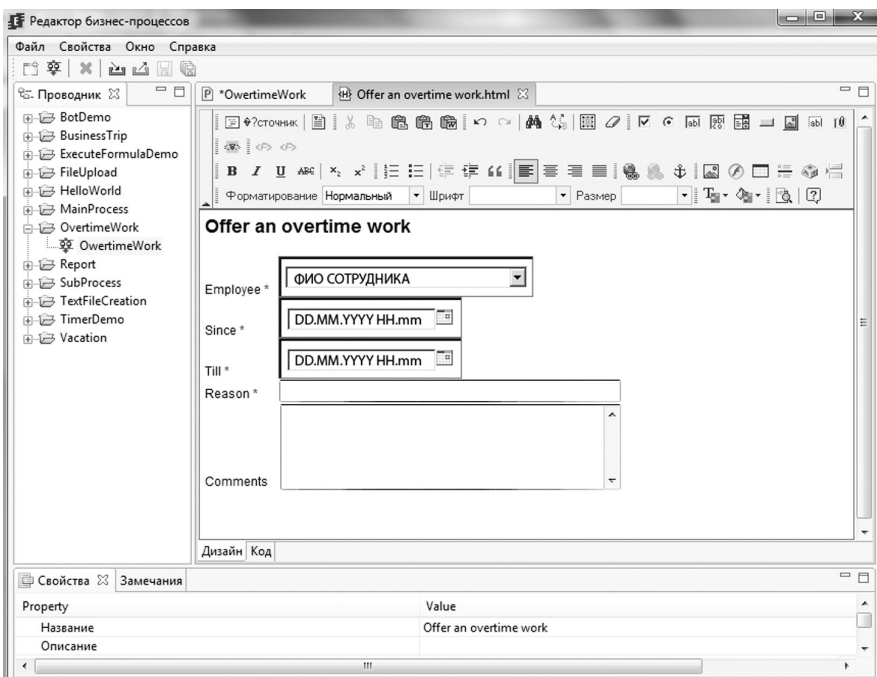


Рисунок 3.5.39

Аналогично создаются остальные формы для остальных узлов-действий

- Make a decision
- Notify for declining
- Notify for acceptance

Форма для узла «Make a decision»:

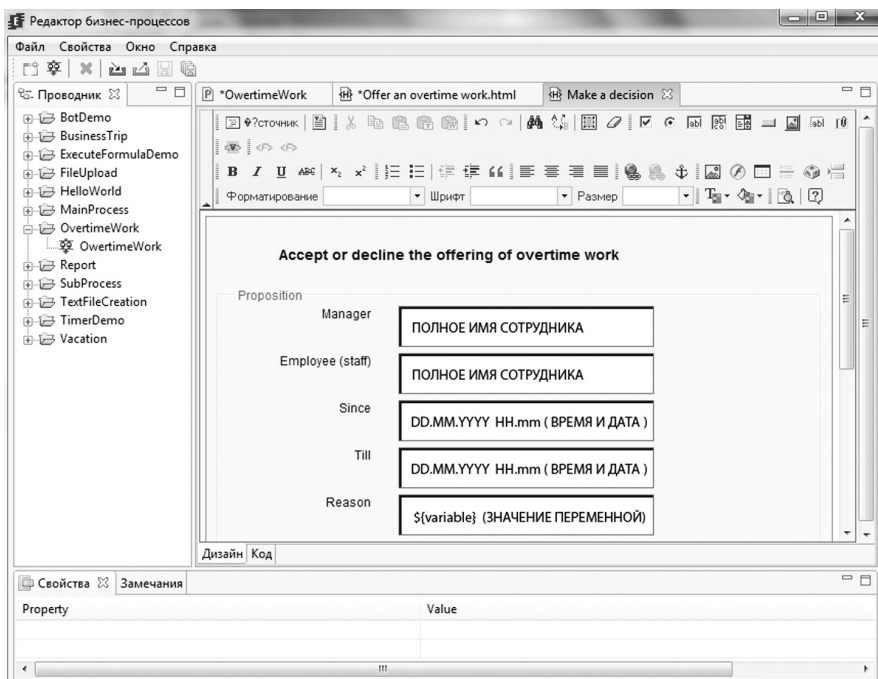


Рисунок 3.5.40

Задание проверок значений, введенных в элементы форм

Элементы форм могут быть обязательными или не обязательными для ввода. Также могут быть правила, относящиеся к нескольким элементам форм. Например, «дата начала» не может быть позже, чем «дата окончания». Кроме того, могут быть специфические для типов ограничения: для чисел — меньше или больше какого-то числа, для строк — ограничение на длину строки и т.д.

Для задания проверки значения полей формы «Offer an overtime work» кликните на узле правой кнопкой мыши, в появившейся форме выберите «Проверка переменных формы»:

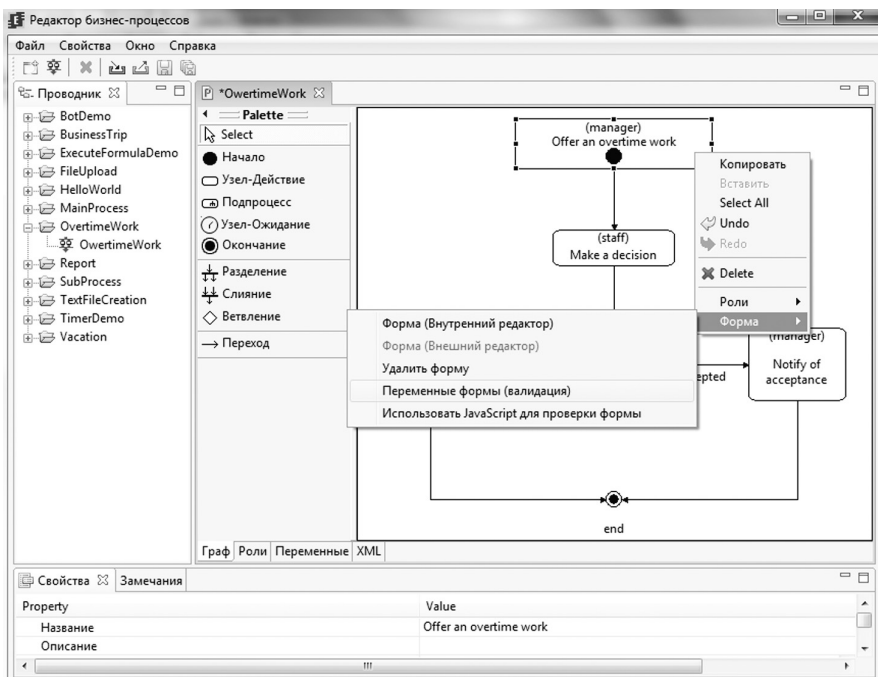


Рисунок 3.5.41

В появившейся форме будут отображены все использующиеся в бизнес-процессе переменные. Все использующиеся в текущей форме переменные будут помечены галочкой.

Для выбранного поля в окне «Проверка переменных формы» будут находиться доступные типы проверок поля. При клике на конкретную проверку в нижнем окне можно будет посмотреть и отредактировать параметры этой проверки:

Валидация формы

Валидаторы переменных формы

Задайте валидацию для переменных формы

Переменные | Роли

☒	staff
☒	since
☒	till
☒	reason
☒	comment
☐	staffPersonDecision
☐	staffPersonComment

Валидаторы

☐	Регулярное выражение
☒	Ограничитель длины строки
☐	url
☐	Валидатор email
☒	Обязательное поле

Строка (тип java.lang.String), удовлетворяющая условиям для длины

Сообщение об ошибке

Length cannot be more than 100 symbols

Удалять пробелы (trim)

Максимальная длина

Минимальная длина

Рисунок 3.5.42

Например, в данном случае для проверки «ограничитель длины строки» установлена максимальная длина 100 символов, в качестве сообщения об ошибке использовано «Длина не может превышать 100 символов».

Если требуется установить проверки на совместные значения нескольких полей, после установки всех проверок на одно поле кликните «Далее».

В появившейся форме при помощи «мастера» можно установить совместные проверки значений полей. Например, что «Дата по» не должна быть позже «Дата с»:

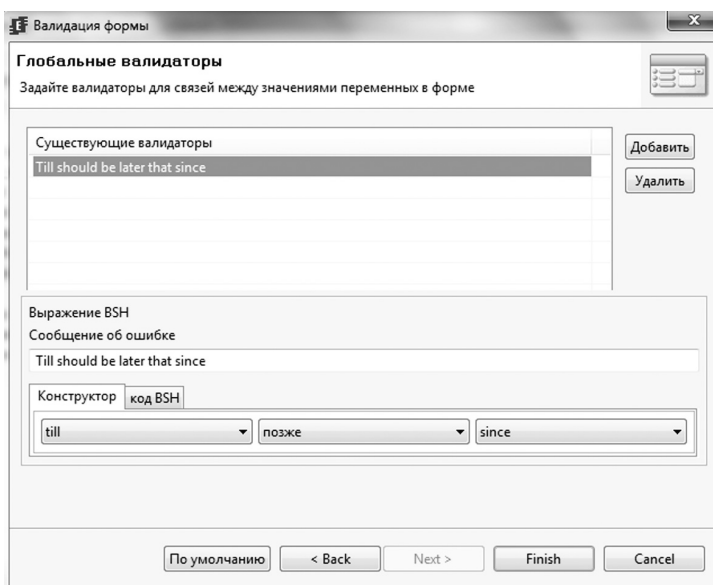


Рисунок 3.5.43

Создание файла-архива бизнес-процесса и загрузка его в систему

Последовательность действий полностью повторяет последовательность, описанную для процесса разработки бизнес-процесса «HelloWorldProcess».

Замечание. Для выполнения процесса «Overtime Work» необходимо создать в Workflow системе группы пользователей:

- manager
- staff
- all

а также завести конкретных пользователей этих групп и раздать соответствующие права.

Задание инициализаторов для ролей-дорожек.

Первый пример

В качестве примера задания инициализатора при создании новой роли рассмотрим создание роли «human resource inspector» для демо-процесса «Vacation».

На закладке «Роли» кликните «Создать». В появившейся форме введите «human resource inspector» в качестве названия роли-дорожки. Кликните «ОК».

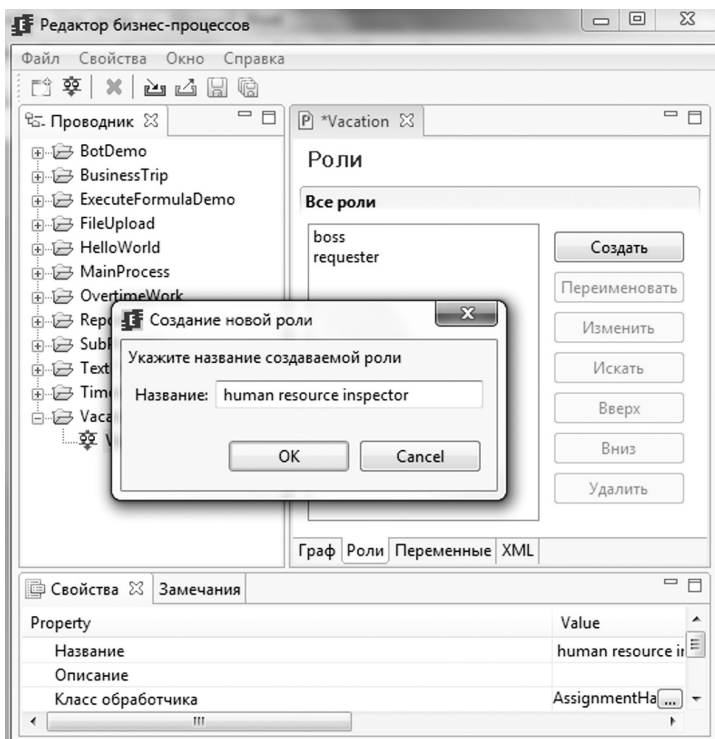


Рисунок 3.5.44

Кликните по роли, нажмите кнопку «Изменить». В появившемся окне на вкладке «Код» выберите инициализатор «функция ExecutorByName» и в качестве параметра введите имя группы, содержащей сотрудников — инспекторов кадровой службы.

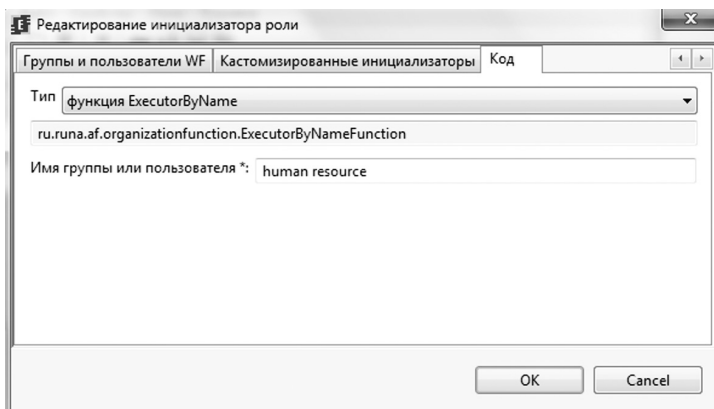


Рисунок 3.5.45

Кликните «OK». Инициализатор новой роли при этом будет определен.

Второй пример

Приведем пример задания инициализатора роли-дорожки, параметром которого является значение переменной бизнес-процесса.

Рассмотрим создание роли «boss» (руководитель) для демо-процесса «Vacation».

На закладке «Роли» кликните «Создать». В появившейся форме введите «boss» в качестве названия роли-дорожки. Кликните «ОК».

Кликните по роли, нажмите кнопку «Изменить». В появившемся окне на вкладке «Код» выберите инициализатор DemoChiefFunction (руководитель сотрудника в случае демо-процессов). В качестве параметра введите строку «\${requester}» — значение переменной requester.

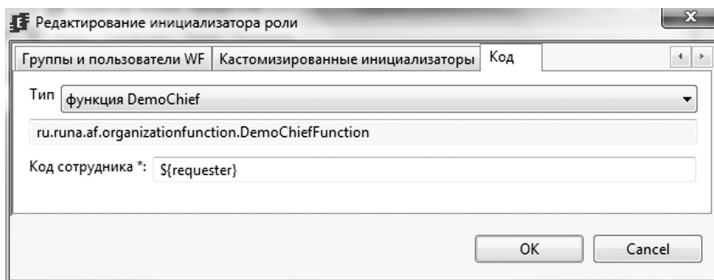


Рисунок 3.5.46

Кликните «ОК». Инициализатор новой роли при этом будет определен.

Замечание. Для задания в качестве параметра значения переменной надо ввести в окно ввода параметра выражение: \${имя_переменной}.

3.6 Клиент-оповещатель о поступивших заданиях.

Клиент-оповещатель реализует оповещение пользователя о поступивших задачах. Кроме того, клиент-оповещатель так же как Web-интерфейс, доступный через браузер, представляет собой среду доступа пользователей к функциональности RunaWFE-сервера. Клиент-оповещатель является приложением, которое устанавливается на компьютер пользователю.

3.6.1 Иконка клиента-оповещателя о поступивших заданиях.

После запуска клиента-оповещателя о поступивших заданиях, если у пользователя в Workflow системе отсутствуют задания, то в правой нижней части экрана появляется значок . Если у пользователя в Workflow системе есть задания, то в правой нижней части экрана появляется значок :



Рисунок 3.6.1

Замечание. В случае операционной системы Windows удобно вызов клиента-оповещателя поместить в автозагрузку системы. Тогда клиент-оповещатель будет вызываться автоматически при запуске системы.

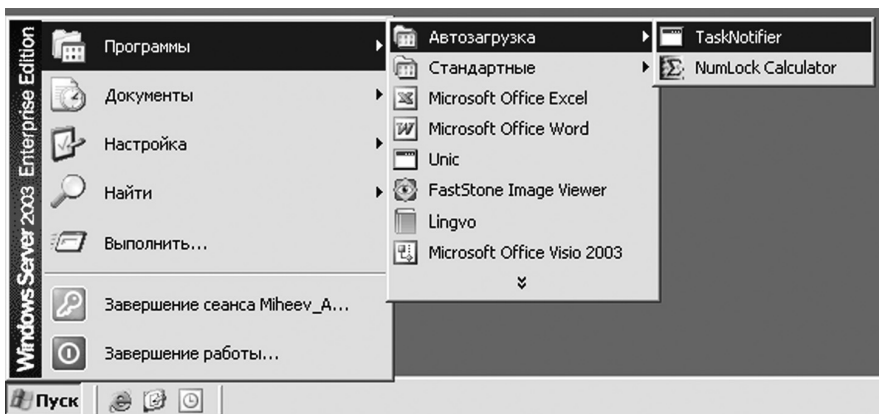


Рисунок 3.6.2

Замечание. Если клиент-оповещатель настроен на аутентификацию по логину и паролю, то при запуске клиента-оповещателя появится диалоговое окно ввода логина и пароля:

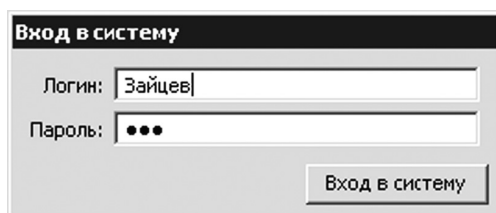


Рисунок 3.6.3

Если логин и пароль введены правильно, то после выполнения «Вход в систему» в правом нижнем углу экрана появится иконка  или . Если логин или пароль введен неправильно, то в правом нижнем углу экрана появится иконка .

3.6.2 Оповещение о поступивших заданиях.

При появлении новых задач у пользователя в правом нижнем углу экрана возникает сообщение о поступивших задачах:

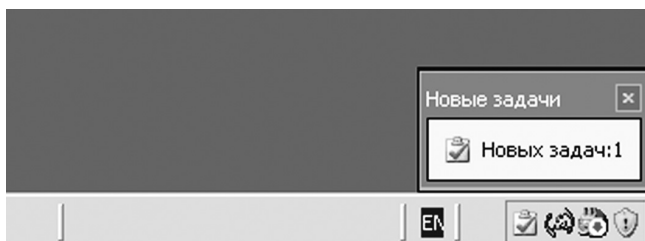


Рисунок 3.6.4

3.6.3 Интерфейс клиента-оповещателя о поступивших заданиях.

При двойном клике на иконку или сообщение о поступивших задачах на экране появляется приложение, содержащее графический интерфейс клиента системы RunaWFE.



Рисунок 3.6.5

3.7 Примеры выполнения простейших бизнес-процессов

В данном разделе содержатся описания выполнения некоторых процессов, содержащихся в дистрибутивах системы RunaWFE, а также установленных в демонстрационную конфигурацию, которая доступна по ссылке: <http://wfdemo.runa.ru/wfe>

В демонстрационной конфигурации сотрудники условного предприятия могут участвовать в специально разработанных демонстрационных бизнес-процессах. Пользователи могут входить под разными именами (логинами) и таким образом выполнять задачи разных сотрудников предприятия.

Список логинов сотрудников условного предприятия:

Таблица 3.7.1.

Логин	ФИО
julius	Gaius Julius Caesar
nero	Nero Claudius Caesar
cleopatra	Cleopatra VII The Daughter of the Pharaoh
octavia	Octavia Minor
caligula	Gaius Iulius Germanicus
tiberius	Tiberius Claudius Drusus
marcus	Marcus Aurelius Antoninus
gaiua	Gaiua Flavius Valerius Constantinus
attila	Attila the King of Huns

Пароль для всех демо-логинов — 123

Список групп пользователей и членов групп:

Таблица 3.7.2.

Группа	Члены группы
manager (Руководители)	• julius • nero
human resource (Отдел кадров)	• cleopatra • octavia
bookkeeper (Бухгалтерия)	• caligula • tiberius
staff (Работники)	• marcus • gaiua • attila
all (Все демо-пользователи)	Все пользователи и группы пользователей

Непосредственный руководитель для любого сотрудника задается специальной функцией над организационной структурой: если логин пользователя начинается на символ «g», то эта функция в качестве непосредственного руководителя возвращает — julius, во всех остальных случаях функция возвращает nero.

Запускать бизнес-процессы Сверхурочные и Командировка могут только руководители (julius, nero), запускать бизнес-процесс Отпуск могут все пользователи.

В настоящем разделе представлены описания выполнения следующих бизнес-процессов:

- Сверхурочные
- Отпуск
- Командировка
- Загрузить-скачать файл
- Отчет
- Создание текстового файла
- Таймер
- Процесс с подпроцессом
- Процесс с мульти-действием

Пример экрана входа в систему в демонстрационном режиме:

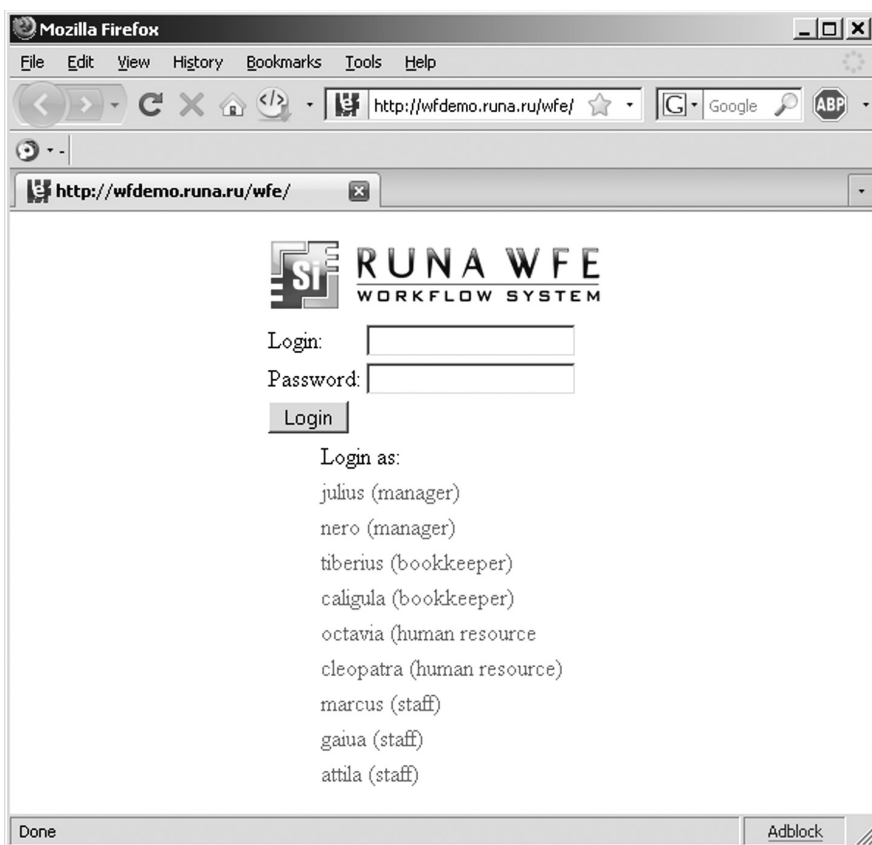


Рисунок 3.7.1

Для ускорения входа можно не вводить логин и пароль, а просто кликнуть на имени пользователя, под которым надо войти в систему.

3.7.1 Описание выполнение бизнес-процесса «Сверхурочные»

Краткое описание процесса:

Руководитель предлагает сотруднику выйти на работу сверхурочно. Сотрудник соглашается или отказывается. Руководитель получает уведомление соответственно о согласии или об отказе.

Графическое изображение процесса:

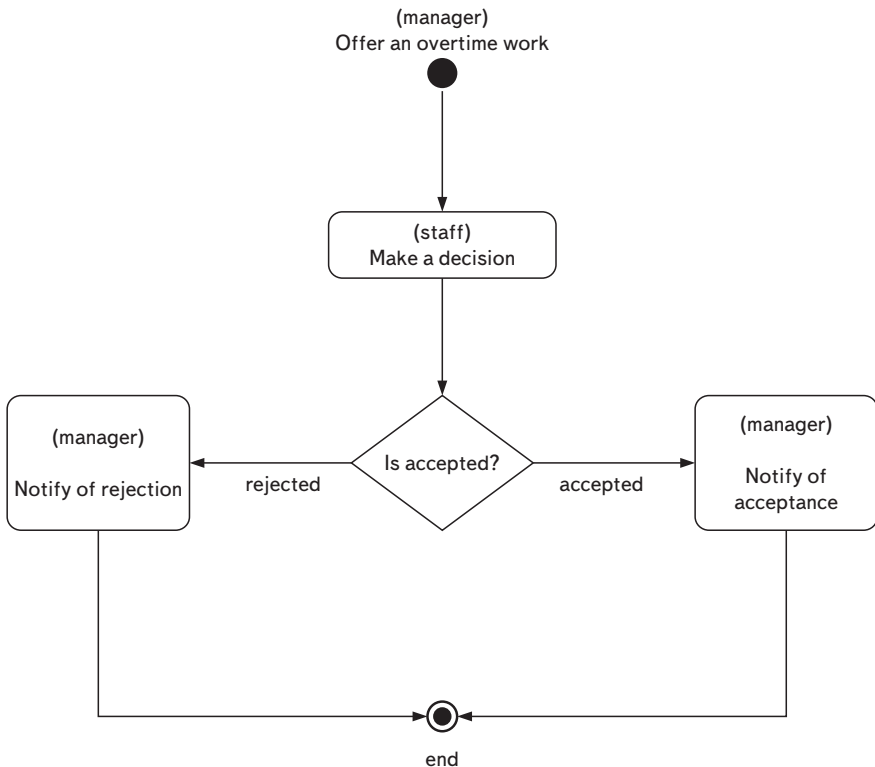


Рисунок 3.7.2

Как исполнить демо-процесс:

1. Войдите в систему под пользователем, входящим в группу manager (например, julius)
2. Кликните на пункте меню «Определения процессов»
3. В появившейся таблице найдите строку, содержащую бизнес-процесс «over time work demo» и кликните на иконку процесса

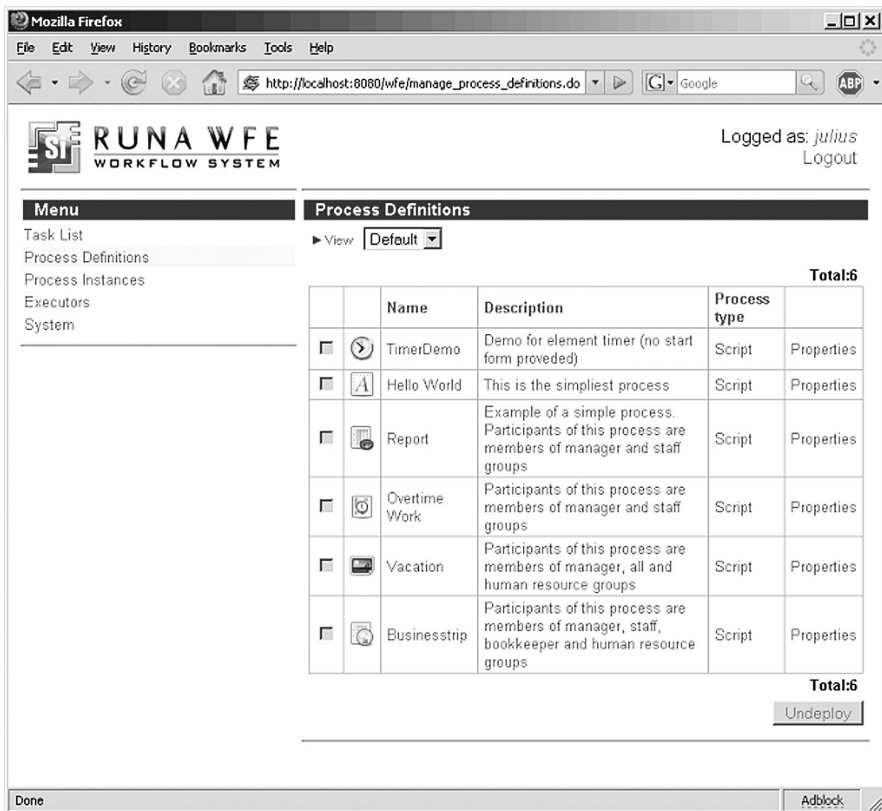


Рисунок 3.7.3

4. В появившейся форме:

- выберите из списка сотрудника, которому будет предложена сверхурочная работа (например, gaiua)
- выберите интервал сверхурочной работы (например, 22.08.2009 11:30 — 30.08.2009 11:30)
- Заполните поля «Причина» и «Комментарий»

Пример заполнения формы:

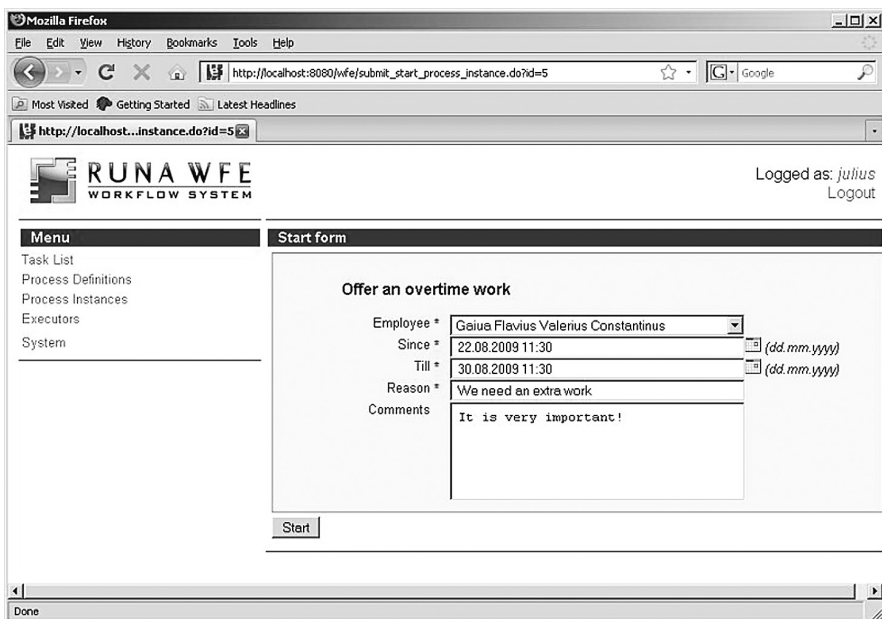


Рисунок 3.7.4

- Нажмите кнопку «запустить» в нижней части формы

После этого будет создан экземпляр бизнес-процесса «Сверхурочные», в рамках этого процесса выбранный в списке сотрудник (в предлагаемом примере — gaiua) получит соответствующее задание.

5. Войдите в систему под другим пользователем, который был выбран из списка в предыдущем пункте (в предлагаемом примере — gaiua). Для этого можно кликнуть на командной кнопке «выход» в правой верхней части экрана, далее ввести новый логин и соответствующий пароль в появившийся форме после чего нажать кнопку «войти».

6. После входа в систему вы окажетесь на странице, соответствующей пункту меню «Задания». В таблице заданий должна быть строка, соответствующая заданию принять или отклонить предложение сверхурочной работы. Кликните на эту строку.

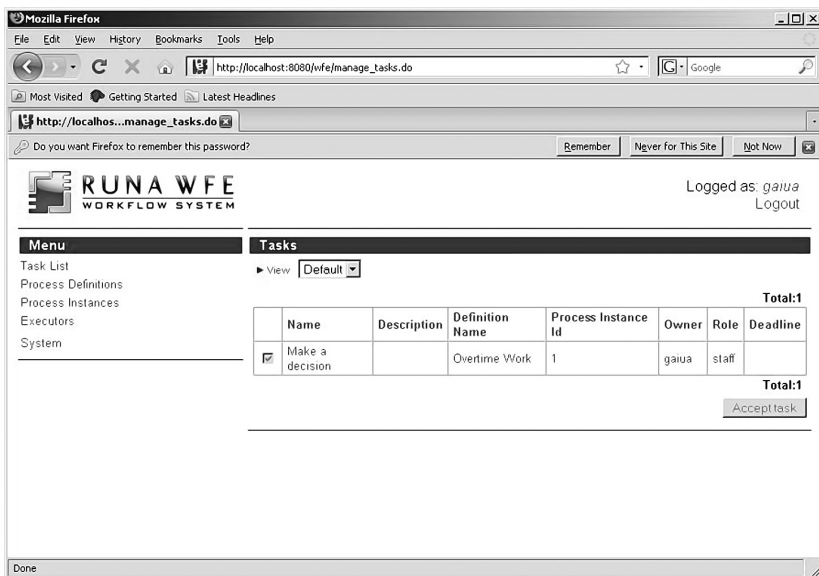


Рисунок 3.7.5

7. В появившейся форме из группы «radio button» выберите «принять» или «отклонить» (например, «принять»), а также заполните поле «комментарий» и нажмите кнопку «выполнить» в нижней части формы.

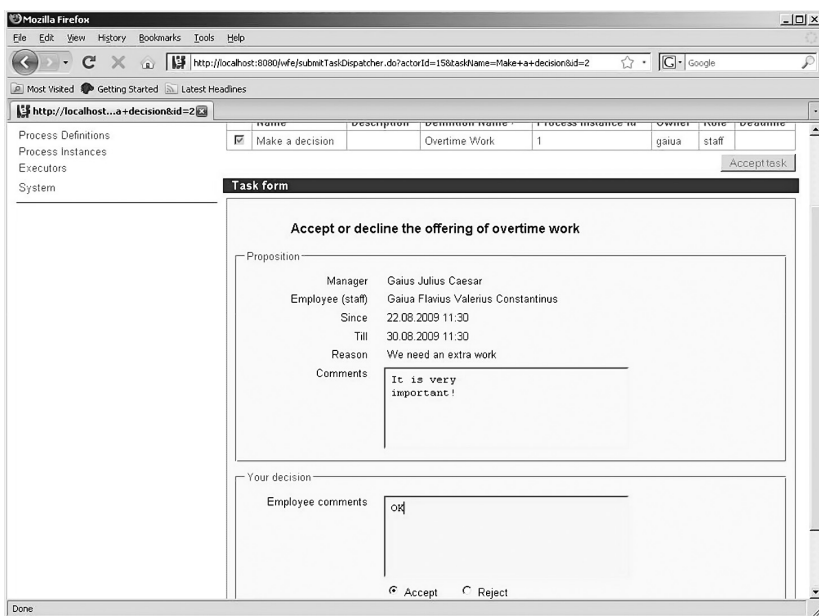


Рисунок 3.7.6

8. Далее опять войдите в систему под пользователем, запустившим процесс (в данном примере — julius)

9. Вы должны увидеть задание — «ознакомиться с сообщением о согласии» или «ознакомиться с сообщением об отказе» (в данном примере — «ознакомиться с сообщением о согласии»). Кликните на это задание.

10. В появившейся форме нажмите кнопку «выполнить» в нижней части формы. При этом бизнес-процесс будет завершен.

Замечание. В процессе выполнения всегда можно посмотреть состояние процесса. На странице, соответствующей меню «экземпляры процессов», надо кликнуть на соответствующий экземпляр бизнес-процесса.

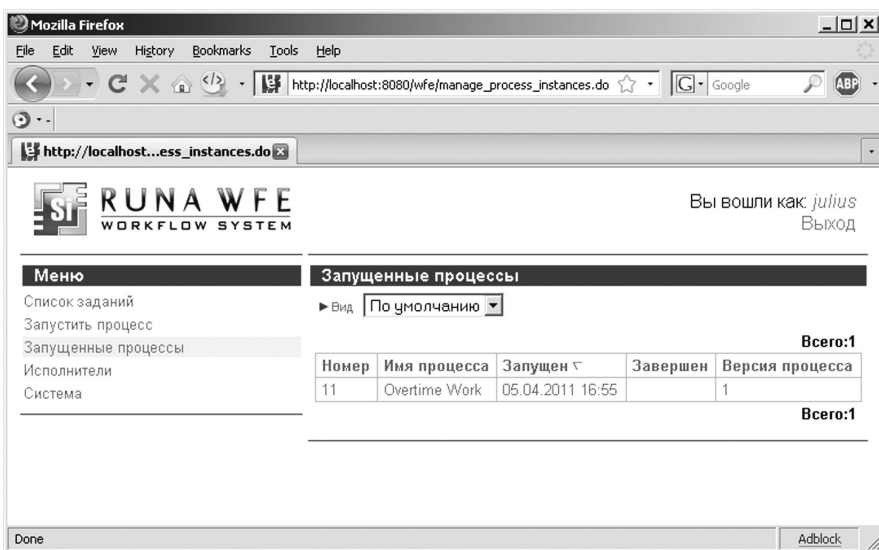


Рисунок 3.7.7

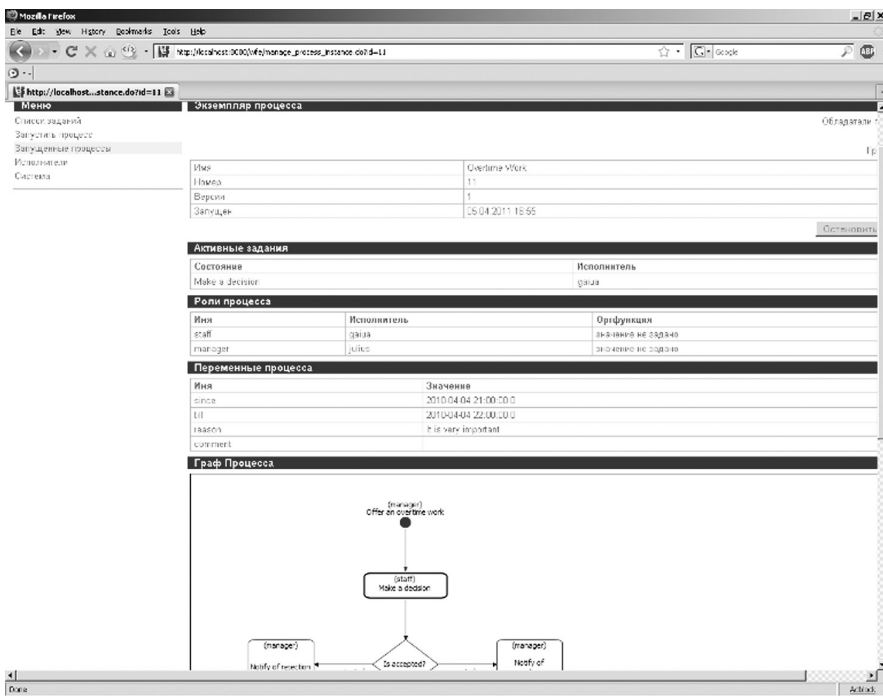


Рисунок 3.7.8

3.7.2 Описание выполнения бизнес-процесса «Отпуск»

Краткое описание процесса:

Сотрудник подает заявку на отпуск. Непосредственный руководитель сотрудника подтверждает или отклоняет заявку. Если заявка отклонена, подавший заявку сотрудник знакомится с соответствующим уведомлением, после чего процесс завершается. Если заявка подтверждена непосредственным руководителем, то далее бизнес-процесс направляет задание «проверить правила и технологии» в кадровую службу. Сотрудник кадровой службы сообщает бизнес-процессу результаты проверки. Если правила и технологии не выполнены, заявка опять направляется на рассмотрение руководителю (этот участок бизнес-процесса является примером цикла), если выполнены, бизнес-процесс переходит к оформлению официального заявления и подписанию приказа: Сотруднику направляется задание «представить в отдел кадров заявление на отпуск». Одновременно работнику отдела кадров направляется задание «получить подписанное заявление на отпуск». После завершения этого задания работник отдела кадров получает задание «подготовить и подписать приказ». После выполнения всех этих заданий бизнес-процесс завершается.

Графическое изображение процесса:

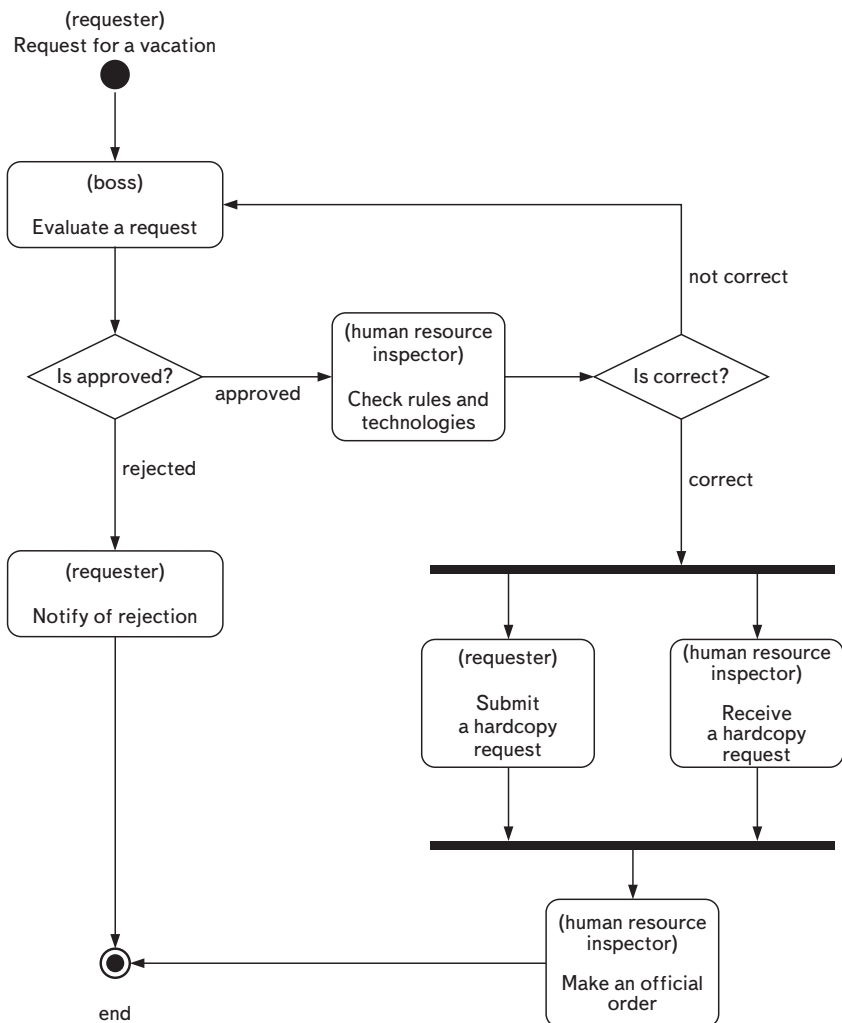


Рисунок 3.7.9

Пример исполнения бизнес-процесса:

1. Войдите в систему под пользователем, входящим в группу staff: (marcus)
2. Кликните на пункте меню «Определения процессов»
3. В появившейся таблице найдите строку, содержащую бизнес-процесс «vacation demo» и кликните на команду «Запустить» в этой строке таблицы.

4. В появившейся форме:
 - выберите интервал дат, соответствующий предполагаемому отпуску
 - Заполните поля «Причина» и «Комментарий»
 - Нажмите кнопку «выполнить» в нижней части формы

После этого будет создан экземпляр бизнес-процесса «Отпуск», в рамках этого процесса следующее задание получит непосредственный руководитель сотрудника, запустившего процесс (т.к. непосредственным руководителем galia является julius, этим пользователем будет julius)
5. Войдите в систему под другим пользователем (julius). Для этого можно кликнуть на командной кнопке «выход» в правой верхней части экрана, далее ввести новый логин и соответствующий пароль в появившийся форме и нажать кнопку «войти»
6. После входа в систему вы окажетесь на странице, соответствующей пункту меню «Задания». В таблице заданий должна быть строка, соответствующая заданию «рассмотреть заявку на отпуск». Кликните на эту строку.
7. В появившейся форме из группы «radio button» выберите «одобрить», а также заполните поле «комментарий руководителя» и нажмите кнопку «выполнить» в нижней части формы.
8. Далее войдите в систему под пользователем, входящим в группу human resource — Отдел кадров (cleopatra)
9. Вы должны увидеть задание — «проверить правила и технологии». Кликните на это задание.
10. В появившейся форме выберите «выполнены» и нажмите кнопку «выполнить» в нижней части формы.
11. Далее в списке заданий вы должны увидеть новое задание «получить заявление на отпуск» — войдите в это задание и нажмите кнопку «выполнить» в нижней части формы.
12. Войдите в систему под другим пользователем (marcus). В списке заданий вы должны увидеть новое задание «отдать в отдел кадров заявление на отпуск». Войдите в это задание и нажмите кнопку «выполнить» в нижней части формы.
13. Войдите в систему под другим пользователем (marcus). В списке заданий вы должны увидеть новое задание «подготовить и подписать приказ». Войдите в это задание, введите дату и номер приказа и нажмите кнопку «выполнить» в нижней части формы. При этом бизнес-процесс будет завершен.

Замечание. В процессе выполнения всегда можно посмотреть состояние процесса на странице, соответствующей меню «экземпляры процессов» надо кликнуть на соответствующий экземпляр бизнес-процесса.

3.7.3 Описание исполнения бизнес-процесса «Командировка»

Замечание. Так как бизнес-процесс достаточно сложный в данном документе для него представлено только краткое описание

Краткое описание процесса:

Руководитель направляет сотрудника в командировку. Сотрудник знакомится с соответствующим уведомлением. Если командировка местная, то на этом бизнес-процесс завершается. Если это командировка в другой регион, то сотрудник кадровой службы получает задание подготовить официальный приказ. Далее сотрудник получает задание подписать приказ, а сотрудник кадровой службы — получить подпись на приказе. После получения подписи на приказе сотрудник получает задание «получить командировочное удостоверение», а сотрудник отдела кадров — выдать командировочное удостоверение. Далее бухгалтер выдает, а сотрудник соответственно получает командировочные. По завершении командировки сотрудник представляет, а бухгалтер получает финансовый отчет о командировке. На этом бизнес-процесс завершается.

Графическое изображение процесса:

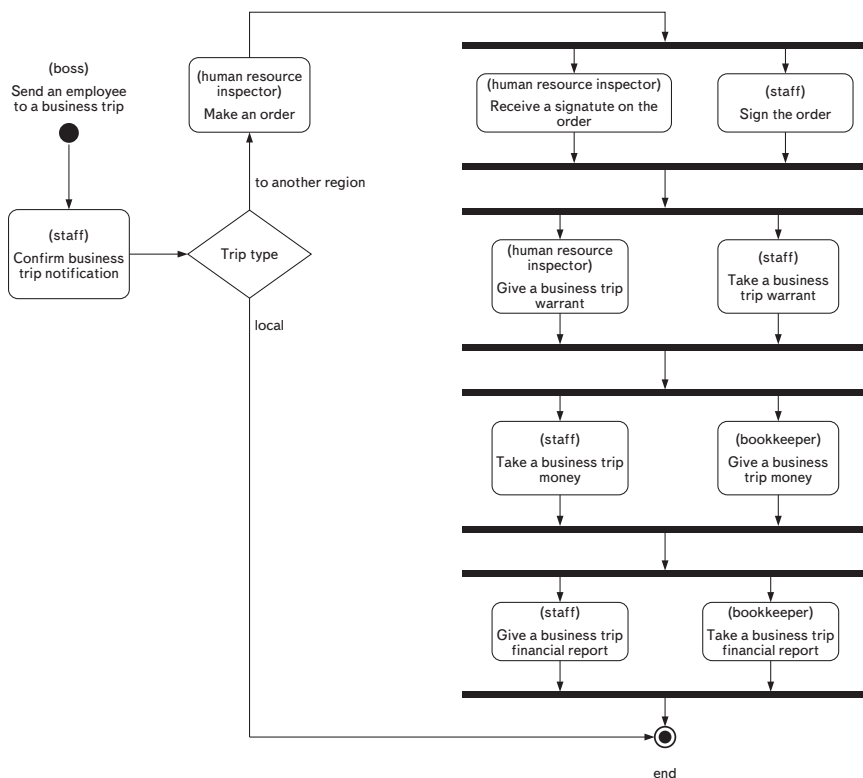


Рисунок 3.7.10

Замечание. Если на странице, соответствующей пункту меню «определения процессов» кликнуть на названии процесса, откроются свойства процесса, и, в частности, будет показан граф бизнес-процесса (в UML-нотации).

3.7.4 Описание демо-процесса «Загрузить, Скачать файл»

Краткое описание процесса:

В стартовой форме процесса пользователь загружает файл. Потом в первом задании этого бизнес-процесса пользователь скачивает файл.

Графическое изображение процесса:

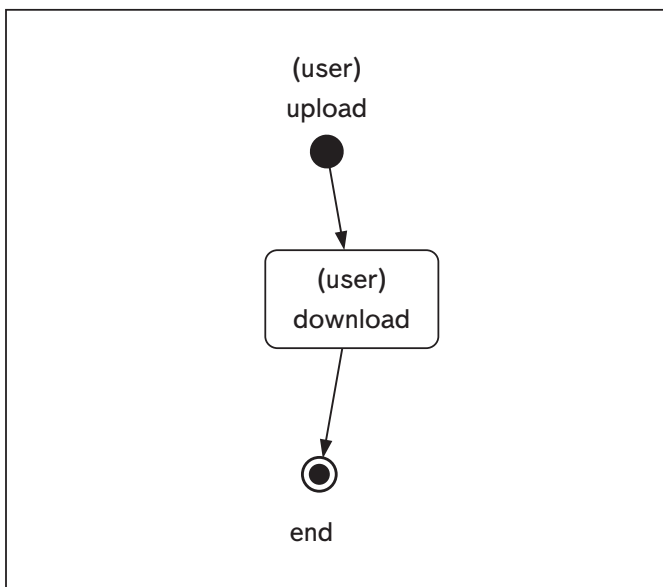


Рисунок 3.7.11

Исполнение бизнес-процесса:

1. Войдите в систему под пользователем, входящим в группу manager (например, julius)
2. Кликните на пункте меню «Определения процессов»
3. В появившейся таблице найдите строку, содержащую бизнес-процесс «FileUpload» и кликните на иконку процесса.
4. В появившейся форме укажите путь к файлу, который затем будет необходимо скачать и кликните на «запустить процесс».

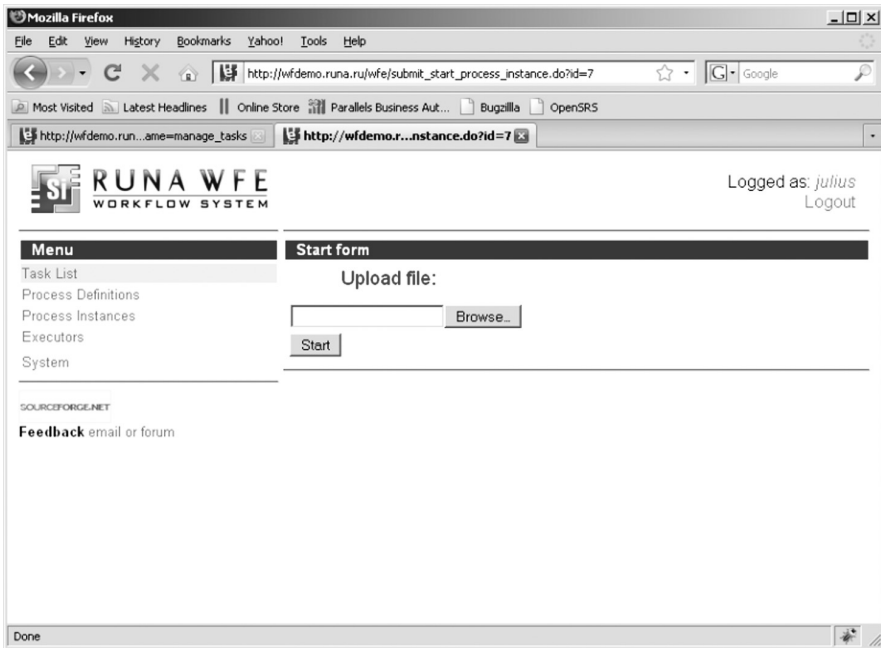


Рисунок 3.7.12

5. Будет создан экземпляр процесса, после чего пользователь Julius получит новую задачу.

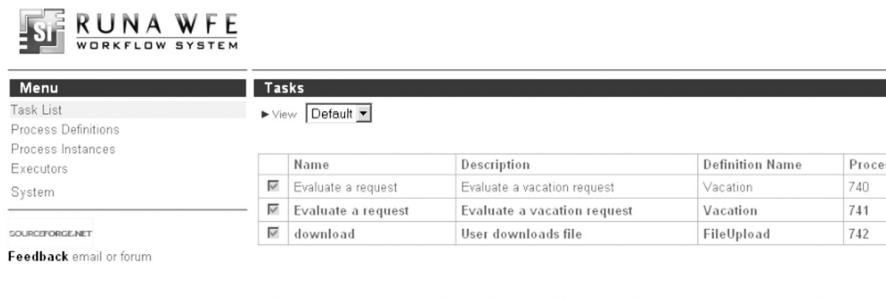


Рисунок 3.7.13

6. Кликнув на имя файла на форме задания, можно скачать файл

The screenshot shows the RUNAWFE Workflow System interface. On the left is a menu with options: Task List, Process Definitions, Process Instances, Executors, and System. The main area displays a task form for a task named 'download'. The task description is 'User downloads file'. Below the task list, there is a section titled 'Task form' with the instruction 'Download file: click on file name: patch.txt'. At the bottom of the form is an 'OK' button.

Name	Description	Definition Name	Process Instance Id	Owner
download	User downloads file	FileUpload	742	julius

Task form

Download file:
click on file name: patch.txt

OK

Рисунок 3.7.14

3.7.5 Описание исполнения бизнес-процесса «Отчет»

Краткое описание процесса:

Руководитель дает задание составить отчет, один из сотрудников составляет отчет, руководитель знакомится с отчетом.

Графическое изображение процесса:

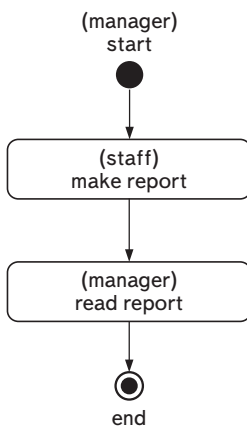


Рисунок 3.7.15

Исполнение бизнес-процесса:

1. Войдите в систему под пользователем, входящим в группу manager (например, julius)
2. Кликните на пункте меню «Определения процессов»
3. В появившейся таблице найдите строку, содержащую бизнес-процесс «Report» и кликните на иконку процесса.
4. Введите тему отчета и нажмите кнопку старт

Menu		Tasks					
Task List		View Default					
Process Definitions							
Process Instances							
Executors							
System							
SOURCEFORGE.NET							
Feedback email or forum							
Name	Description	Definition Name	Process Instance Id	Owner	Role		
☒ download	User downloads file	FileUpload	742	julius	user		
☒ read report	report is ready	Report	743	julius	manager		
☒ read report	report is ready	Report	744	julius	manager		
☒ read report	report is ready	Report	745	julius	manager		

Рисунок 3.7.19

8. Выберите любой из отчетов, который сделали сотрудники и после прочтения нажмите кнопку «ОК». После этого текущая задача будет выполнена. После выполнения всех задач список задач уже не будет содержать задачи «report is ready».

3.7.6 Описание исполнения бизнес-процесса «Создание текстового файла»

Краткое описание процесса:

Пользователь вводит текстовую информацию, бот создает текстовый файл, пользователь читает или сохраняет файл на диск.

Графическое изображение процесса:

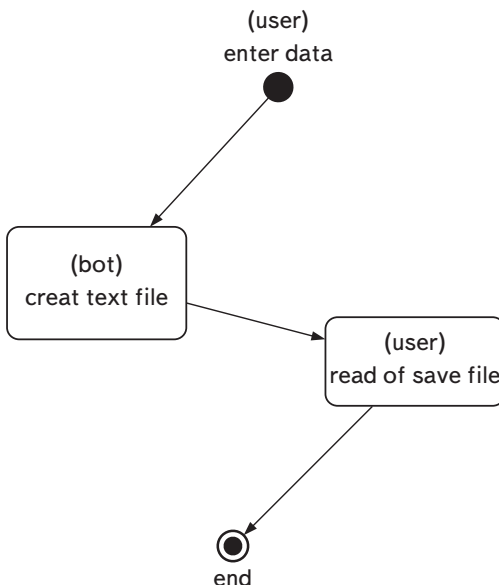


Рисунок 3.7.20

Исполнение бизнес-процесса:

1. Войдите в систему под пользователем, входящим в группу manager: (например, julius)
2. Кликните на пункте меню «Определения процессов»
3. В появившейся таблице найдите строку, содержащую бизнес-процесс «TextFileCreation» и кликните на иконку процесса.
4. Ведите текстовые данные и нажмите кнопку старт.

The screenshot shows the RUNAWFE Workflow System interface. On the left is a 'Menu' with links: Task List, Process Definitions (highlighted), Process Instances, Executors, and System. Below the menu is a 'Feedback' link. The main area is titled 'Start form' and contains the text 'Enter parameters:'. It has three input fields: 'Date' with the value '24.12.2009', 'Number' with the value '3', and 'Header' with the value 'Test'. Below these is a large text area labeled 'Main text:' containing the text 'Simple text'. At the bottom of the form is a 'Start' button.

Рисунок 3.7.21

5. Кликните в меню ссылку «Task List», в списке задач должна появиться новая задача: «Прочитать или сохранить файл». Кликните по ссылке, сохраните файл на диск, завершите задачу.

3.7.7 Описание исполнения бизнес-процесса «Таймер»

Краткое описание процесса:

Пользователь запускает процесс, управление переходит в узел «before timer triggered». Если в течение трех минут пользователь выполнит задачу в этом узле, то управление перейдет в узел «окончание» и процесс завершится. Если в течение трех минут пользователь не выполнит задачу, то таймер сработает и управление перейдет в узел «after timer triggered».

Графическое изображение процесса:

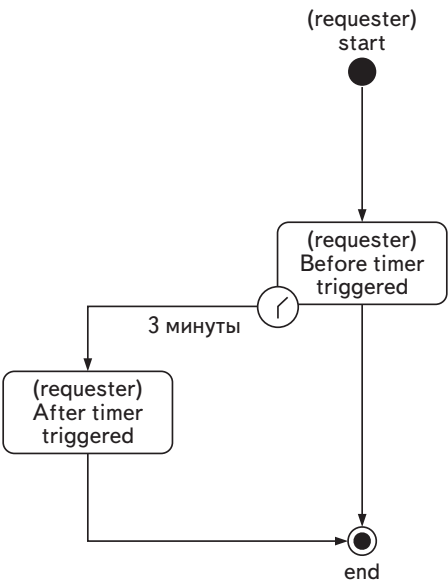


Рисунок 3.7.22

Исполнение бизнес-процесса:

- 1. Войдите в систему под пользователем, входящим в группу manager: (например, julius)
- 2. Кликните на пункте меню «Определения процессов»
- 3. В появившейся таблице найдите строку, содержащую бизнес-процесс «TimerDemo» и кликните на иконку процесса. При этом создастся и запустится экземпляр процесса.

Process instance 903 started

Menu

- Task List
- Process Definitions
- Process Instances
- Executors
- System

Name	Description	Definition Name	Process Instance Id	Owner	Role
☒ Before timer triggered	Before timer triggered	TimerDemo	903	julius	requester

Task form

Timer not triggered yet

Ok

Рисунок 3.7.23

4. Таймер в процессе установлен на 3 минуты. После истечения этого периода времени, таймер сработает и управление перейдет по переходу, присоединенному к таймеру

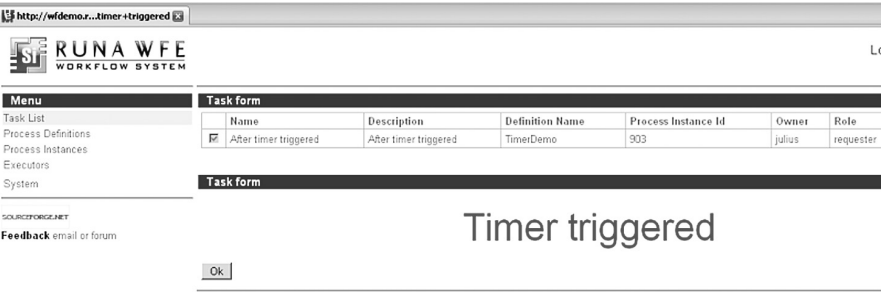


Рисунок 3.7.24

3.7.8 Описание исполнения бизнес-процесса с подпроцессом

В процесс в качестве подпроцесса можно включить другой отдельный процесс. Подпроцесс может быть как самостоятельным (т.е. его можно запускать отдельно от родительского процесса), так и полностью зависимым от процесса-родителя (не иметь смысла вне родительского контекста). Один и тот же подпроцесс может быть включен в разные процессы.

Для отображения подпроцесса в графическом редакторе процессов есть отдельный элемент. О создании и настройке связи между подпроцессом и родительским процессом см. Руководство пользователя графическим редактором процессов.

Пример демо-процесса MainProcess (слева) с подпроцессом SubProcess (справа):

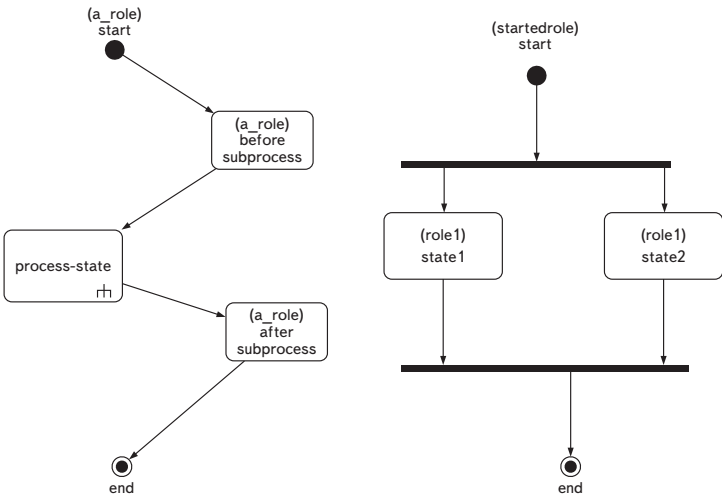


Рисунок 3.7.25

1. Для запуска демо-процесса «MainProcess» кликните в меню на «Запустить процесс». В появившемся списке процессов найдите «MainProcess» и кликните на название. Процесс запустится. На открывшейся форме «before subprocess» нажмите «Задание выполнено».

2. В списке задач найдите задачи процесса MainProcess. Появившиеся задачи принадлежат подпроцессу SubProcess, но в колонке «Имя процесса» стоит имя процесса-родителя. Выполните задачи подпроцесса. Затем выполните последнюю задачу процесса-родителя «after subprocess». Процесс завершится.

3. Для просмотра свойств процессов перейдите в «Запущенные процессы». Раскройте только что завершённый процесс MainProcess. На графе процесса кликните на узле подпроцесса, вы перейдете на страницу со свойствами экземпляра подпроцесса SubProcess, выполненного в рамках MainProcess.

3.7.9 Описание исполнения бизнес-процесса с мульти-действием

В случае, когда в бизнес-процессе необходимо выполнить одну и ту же последовательность действий для списка значений переменной, для этой последовательности создается подпроцесс-мультидействие. Для каждого экземпляра такого бизнес-процесса запускается столько подпроцессов-мультидействий, сколько значений в списке значений переменной мультидействия. Например, у заказчика несколько адресов, на каждый адрес необходимо оформить отдельный наряд в рамках одного заказа. В переменной мультидействия будет список адресов заказчика. В экземпляре родительского процесса запустится столько подпроцессов, сколько адресов в списке. В каждом подпроцессе будет переменная с одним из адресов. Тип переменной-мультидействия в родительском процессе должен быть StringArray, в подпроцессе — String.

Для добавления узла-мультидействия в процесс в графическом редакторе процессов нужно выбрать узел «Мульти-действие» из палитры и поместить на диаграмму. Затем настроить мэппинг переменных. См. Руководство пользователя графического редактора процессов.

Пример графа процесса в мультидействии (слева) и его подпроцесса (справа):

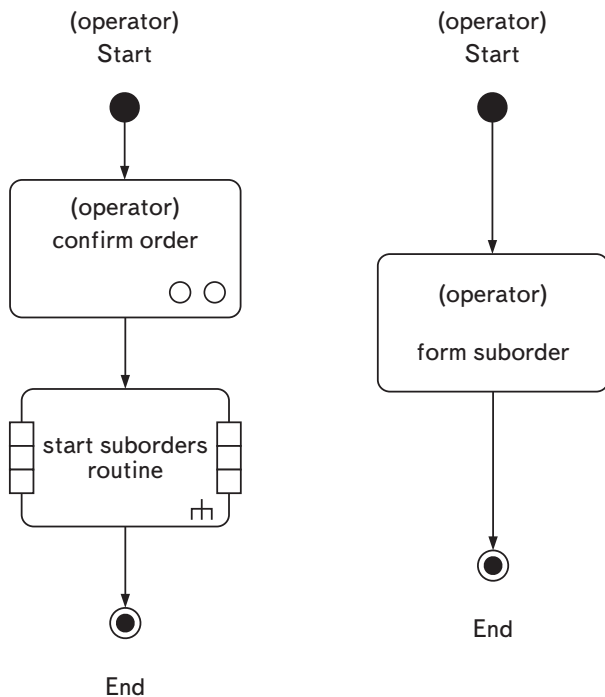


Рисунок 3.7.26

1. Для запуска демо-процесса «MultilInstanceProcess» кликните в меню на «Запустить процесс». В появившемся списке процессов найдите «MultilInstanceProcess» и кликните на название.
2. На стартовой форме процесса нужно выбрать количество адресов заказчика, что совпадает с числом запускаемых подпроцессов мультидействия.
3. Зайдите в «Список заданий» и подтвердите заказ.
4. Для наглядности зайдите в «Запущенные процессы» и проверьте сколько подпроцессов MultilInstanceSubProcess запущено. Кликните на одном из подпроцессов и посмотрите значение переменной address в свойствах экземпляра подпроцесса. Посмотрите значение этой переменной в другом подпроцессе.
5. Зайдите в «Список заданий» и сформируйте наряд для каждого подпроцесса.

На этом процесс завершается.

3.8 Проект RunaWFE

Проект RunaWFE размешен на сайте разработчиков открытого программного обеспечения sourceforge по адресу <http://sourceforge.net/projects/runawfe>. На этом сайте можно скачать систему как в исходных кодах, так и в скомпилированном виде, посмотреть и послать сообщения в форум, познакомиться с последними новостями по проекту.

RunaWFE — проект разработки свободного программного обеспечения. Он открыт для сотрудничества и кооперации различных участников. Вся инфраструктура проекта, включая текущий код и историю его изменений, находится в интернете. Поэтому участвовать в развитии проекта можно, находясь в любой точке земного шара, в которой есть Интернет.

Краткая история проекта:

Проект RunaWFE начался в сентябре 2003 года. Интересно, что источником финансирования данного проекта разработки ПО с открытым кодом была не государственная структура, а коммерческая компания. Консалтинговая группа РУНА решила внедрить у себя систему управления бизнес-процессами и административными регламентами. Зрелых российских программных продуктов тогда на рынке обнаружено не было, а стоимость иностранных программных продуктов составляла от 450 до 2000 долларов США за 1 рабочее место. Кроме затрат на покупку системы, заметных расходов потребовало бы внедрение иностранной системы, а также обучение и поддержка. В то время в компании работало более 500 человек, поэтому величины расходов на покупку и внедрение какой-либо иностранной системы оказались бы значительными. Было решено не тратить деньги на покупку одного экземпляра иностранной системы, а потратить их на сборку системы из уже существующих компонентов с открытым кодом.

Летом 2004 года началось кодирование. В ноябре 2004 года на портале sourceforge был заведен проект RunaWFE и была выложена первая версия кода системы. В июне 2005 г. появилось OnLine demo. В конце 2005 года в интернет была выложена первая версия редактора бизнес-процессов. В конце 2006 года разработка базовой конфигурации системы была закончена. В период с 1 января 2007 г. по 30 июня 2007 г. система прошла в консалтинговой группе РУНА промышленное тестирование и с 1 июля 2007 г. система была переведена в промышленную эксплуатацию.

В 2005 году проект RunaWFE стал дипломантом конкурса Java-технологий, проводившимся корпорацией Sun Microsystems при официальной поддержке Министерства информационных технологий и связи РФ, в 2006 году проект получил Honorable Mentions статус на конкурсе JBoss Innovation Award в двух категориях: «Управление бизнес-процессами» и «Хранение информации».

В 2008 г. дистрибутив системы RunaWFE был помещен репозиторий свободных программ Sisyphus (<http://sisyphus.ru>), система была включена в открытые дистрибутивы ОС Linux — ALT Linux и Mandriva.

В настоящее время на портале sourceforge проект имеет статус Production/Stable. Система RunaWFE используется OpenSource сообществом пользователей во всем мире. С портала sourceforge произведено более 45000 скачиваний системы.

4. Список литературы.

- 1 Абдикеев Н.М., Данько Т.П., Ильдеменов С.В., Киселев А.Д. Реинжиниринг бизнес-процессов. М.: Эксмо, 2005. 592 с.
- 2 Калянов Г. Н. Моделирование, анализ, реорганизация и автоматизация бизнес-процессов. — М.: Финансы и статистика, 2006. 240 с.
- 3 Куликов Г. Г., Михеев А. Г., Орлов М. В., Габбасов Р. К., Антонов Д. В. Изучение методологии BPMN на примере программного продукта RunaWFE. Лабораторный практикум по дисциплине «Автоматизированные информационные системы в производстве» и «Автоматизированные информационные системы в экономике». — Уфа. УГАТУ. 2010
- 4 Михеев А. Г., Орлов М. В. Перспективы Workflow-систем / PC Week/RE, №№ 23 2004, 28 2004, 43 2004, 36 2005
- 5 Петерсон Л. Г. Математика. Учебники для 1-4 класса. Ювента.
- 6 Тельнов Ю.Ф. Реинжиниринг бизнес-процессов: Компонентная методология. — М.: Финансы и статистика, 2004. 319 с.
- 7 Фиошин. М. Основы π -исчисления.
- 8 W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, A.P. Barros. Workflow Patterns. 2002.
- 9 W.M.P. van der Aalst. Patterns and XPDL: A Critical Evaluation of the XML Process Definition Language. Department of Technology Management. Eindhoven University of Technology, The Netherlands.
<http://www.workflowpatterns.com/documentation/documents/ce-xpdl.pdf>
- 10 W.M.P. van der Aalst, M. Dumas, A.H.M. ter Hofstede, P. Wohed. Pattern based analysis of BPML <http://xml.coverpages.org/Aalst-BPML.pdf>
- 11 W.M.P. van der Aalst, A.H.M. ter Hofstede. YAWL: Yet Another Workflow Language.
- 12 W.M.P. van der Aalst. Pi calculus versus Petri nets: Let us eat «humble pie» rather than further inflate the «Pi hype». <http://is.ieis.tue.nl/staff/wvdaalst/pi-hype.pdf>
- 13 Tom Baeyens The State of Workflow
<http://www.theserverside.com/news/1365159/The-State-of-Workflow>
- 14 Lauri Eloranta, Eero Kallio, Ilkka Terho A Notation Evaluation of BPMN and UML Activity Diagrams.
(http://www.soberit.hut.fi/T-86/T-86.5161/2006/BPMN_vs_UML_final.pdf)

- 15 Hammer M., Champy J. Reengineering the Corporation: A Manifesto for Business Revolution. — New York: HarperBusiness, 1993
- 16 High-level Petri Nets — международный стандарт ISO/IEC 15909, версия 4.7.1. www.informatik.hu-berlin.de/top/PNX/pnstd-4.7.1.pdf.
- 17 S. Jablonski and C. Bussler. Workflow Management: Modeling Concepts, Architecture, and Implementation. International Thomson Computer Press, London, UK, 1996.
- 18 jPDL Reference Manual <http://docs.jboss.org/jbpm/v3/userguide/jpdl.html>
- 19 B.Kiepuszewski Expressiveness and suitability of languages for control flow modeling in workflow
- 20 Koulopoulos T. M. The Workflow Imperative. — New York: Van Nostrand Reinhold, 1995
- 21 Коалиция OMG/BPMI, графическая нотация BPMN. <http://www.bpmn.org>
- 22 OMG. Workflow Management Facility Specification
- 23 Petri Nets. Introductory tutorial. <http://www.informatik.uni-hamburg.de/TGI/PetriNets> An introduction to Petri nets.
- 24 R Shapiro. A comparison of XPDL, BPML and BPEL4WS. <http://xml.coverpages.org/Shapiro-XPDL.pdf>
- 25 WfMC. Workflow reference model.
- 26 The Workflow Management Coalition Specification, Workflow Standard, Process Definition Interface, XML Process Definition Language, Document Number WfMC-TC-1025, October 10, 2008, Version 2.1a
- 27 Web Services Business Process Execution Language Version 2.0. OASIS Standard. <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>
- 28 WfMC. Terminology and glossary. www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf
- 29 WfMC. WAPI (Workflow Application Programming Interface)
- 30 WfMC. Workflow Standard — Interoperability Wf-XML Binding. www.wfmc.org/standards/docs/Wf-XML-11.pdf
- 31 Stephen A. White Process Modeling Notations and Workflow Patterns (<http://www.bptrends.com/publicationfiles/03-04%20WP%20Notations%20and%20Workflow%20Patterns%20-%20White.pdf>)
- 32 P. Wohed , Van der Aalst, M. Dumas, A.H.M. ter Hofstede. Pattern based analysis of BPEL4WS. <http://xml.coverpages.org/AalstBPEL4WS.pdf>

Учебное издание

Серия «Библиотека ALT Linux»

Андрей Михеев

**Системы управления бизнес-процессами и административными
регламентами на примере свободной программы RunaWFE**

Оформление обложки: А.С. Осмоловская

Подписано в печать 04.04.2011. Формат 70х100/16.
Гарнитура Computer Modern. Печать офсетная. Бумага офсетная.
Усл. печ. л. 26. Заказ

ООО «Альт Линукс»
Адрес для переписки: 119334, г. Москва, 5-й Донской проезд, д. 15, стр. 6,
(для ООО «Альт Линукс»)
Телефон: (495) 662-38-83.
E-mail: sales@altlinux.ru
<http://altlinux.ru>