

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

И. И. Пилецкий, В. Н. Козуб

**ТЕХНОЛОГИЯ ИНТЕГРАЦИИ ДАННЫХ
КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ.
ОСНОВЫ ТЕХНОЛОГИИ РЕАЛИЗАЦИИ
БИЗНЕС-ПРОЦЕССОВ ИНФОРМАЦИОННЫХ
СИСТЕМ, ЯЗЫКИ BPMN И VPEL**

*Рекомендовано УМО по образованию в области информатики
и радиоэлектроники в качестве пособия для специальности
1-40 04 01 «Информатика и технологии программирования»*

Минск БГУИР 2019

УДК 004.415.26(076)

ББК 32.973.2я73

ПЗ2

Рецензенты:

кафедра менеджмента технологий государственного учреждения образования «Институт бизнеса и менеджмента технологий» Белорусского государственного университета (протокол №2 от 29.09.2017);

заведующий лабораторией информационного обеспечения научных исследований государственного научного учреждения «Объединенный институт проблем информатики Национальной академии наук Беларуси»
кандидат технических наук, доцент Р. Б. Григянец

Пилецкий, И. И.

ПЗ2 Технология интеграции данных корпоративных информационных систем. Основы технологии реализации бизнес-процессов информационных систем, языки BPMN и BPEL : пособие / И. И. Пилецкий, В. Н. Козуб. – Минск : БГУИР, 2019. – 75 с. : ил.

ISBN 978-985-543-518-2.

Содержит описание технологии, инструментов и средств разработки бизнес-процессов, а также пример реализации бизнес-процесса для приложения на языке BPEL.

Предназначено для изучения учебной дисциплины «Современные средства проектирования информационных систем».

УДК 004.415.26(076)

ББК 32.973.2я73

ISBN 978-985-543-518-2

© Пилецкий И. И., Козуб В. Н., 2019
© УО «Белорусский государственный университет информатики и радиоэлектроники», 2019

Содержание

1. Технология разработки бизнес-процессов.....	4
1.1. Язык моделирования бизнес-процессов BPMN	4
1.2. Отображение BPMN на другие языки моделирования.....	6
1.3. Основные элементы языка описания бизнес-процессов	6
1.4. Бизнес-интеграция.....	21
1.5. Жизненный цикл разработки приложений, интегрированных с бизнесом.....	23
2. Инструменты IBM WebSphere для реализации бизнес-интеграции.....	25
2.1. IBM WebSphere Business Modeler	26
2.2. IBM WebSphere Integration Developer.....	27
2.3. IBM WebSphere Process Server	29
2.4. IBM WebSphere Business Monitor.....	29
2.5. Процесс реализации бизнес-интеграции.....	31
3. Бизнес-интеграция. Пример разработки большого приложения.....	33
3.1. Реализация бизнес-процессов для приложения «Дорога»	34
3.2. Диаграмма сборки для модуля «Формирование месячного плана».....	40
3.3. Объекты-данные в модели SDO.....	44
3.4. Граф данных в модели SDO	45
3.5. Реализация бизнес-процесса на языке BPEL	48
3.6. Реализация бизнес-процессов в виде конечного автомата на языке BPEL	57
3.6.1. Описание машины состояний	59
3.6.2. Инициализация бизнес-процесса по расписанию. Пример взаимодействия SCA и EJB	64
Приложение. Пример WSDL-интерфейса для SCA-компонента GenerateMonthlyPlanGU12	67
Перечень принятых терминов	72
Список использованных источников.....	74

1. ТЕХНОЛОГИЯ РАЗРАБОТКИ БИЗНЕС-ПРОЦЕССОВ

В данном пособии рассматривается применение адаптивной технологии построения сложной корпоративной системы на базе примера разработки конкретной модели, спроектированной с использованием BPM/BPMN (Business Process Management Notation) [1, 2], в среде IBM WebSphere Business Modeler, реализованной на языке выполнения BPEL [3] в среде IBM WebSphere Integration Developer [4].

1.1. Язык моделирования бизнес-процессов BPMN

В качестве стандарта для моделирования бизнес-процессов получила признание нотация BPMN (Business Process Modeling Notation – нотация моделирования бизнес-процессов), разработанная организацией Business Process Modeling Initiative (BPMI.org) [2].

Основное назначение BPMN заключается в предоставлении нотации, легкой в использовании и понимании для бизнес-пользователей, включая бизнес-аналитиков, моделирующих бизнес-процессы, технических разработчиков, которые создают системы для выполнения этих процессов, и менеджеров различных уровней, которые должны быстро читать и понимать процессные диаграммы, чтобы принимать деловые решения.

BPMN напрямую отображается на языки исполнения бизнес-процессов, такие как BPEL и BPMN. BPMN предоставляет нотацию для моделирования, а BPEL является языком описания исполнения процессов.

К ключевым особенностям BPMN относятся:

1. Пулы и дорожки (Pools and Lanes). Они используются для демаркации процессов и систем. Пул используется, чтобы разделить различные бизнес-сущности или участников. Действия (Activities) внутри пулов – это модульные (Self-contained) процессы. Дорожки (Lanes) используются для описания и разделения действий в пулах (как водные дорожки в бассейне). Они, как правило, используются для группировки действий по функциям или ролям.

2. События и действия (Events and Activities). События используются для представления того, что происходит в процессе бизнес-деятельности. У событий обычно есть причины/эффекты (следствия), и они могут влиять на сам поток. Действия ссылаются на работу, которая выполняется либо как единая задача, либо как набор задач (подпроцесс).

3. Соединенные объекты (Connected Objects). Различные сущности BPMN могут быть соединены через последовательность операций, или поток операций (Sequence Flow), чтобы отметить порядок, в котором действия выполняются,

или через поток сообщений, чтобы отметить сообщения между сущностями, или через ассоциацию, используемую для ассоциирования текста и других артефактов с объектами потока (Flow Objects).

4. Проектирование сложного процесса (Complex Process Design). BPMN может использоваться как для высокоуровневого описания процессов, так и для описания сложных процессов со многими уровнями детализации. Процессы могут включать детализирующие представления (Drill-down Views) для описания деталей более низкого уровня (Lower Levels of Detail) внутри отдельных диаграмм.

5. Моделирование и контроль сообщений (Model and Control the Message). В дополнение к спецификации порядка операций в потоке BPMN обеспечивает представление об объектах данных для использования и изменения документов, данных и других объектов во время процессного потока.

BPMN предоставляет нотацию моделирования, которая обеспечивает переход от бизнес-определений к карте исполнения процесса (Process Execution Map). Объекты BPMN обладают богатым набором атрибутов, которые позволяют легко отображать их в описаниях BPEL.

Нотация BPMN расширяема и позволяет использовать ассоциации и аннотации для установления взаимоотношений с другими артефактами внутри или вне текущей системы. Например, можно соотнести бизнес-процессы с функциями, которые они выполняют, с данными, которые они используют, с системами, на которых они развернуты, и т. д.

Основное назначение стандарта BPMN – создание нотации, понятной всем участникам бизнес-сферы – от бизнес-аналитиков, создающих первоначальные эскизы процессов, технических разработчиков, ответственных за внедрение технологии, в которой будут представлены данные процессы, до бизнесменов, которые будут управлять этими процессами, а также осуществлять их мониторинг. Таким образом, BPMN является стандартизованным связующим звеном между разработкой бизнес-процессов и их реализацией.

Другой не менее важной целью является визуализация посредством бизнес-ориентированной нотации языков XML, например BPEL, BPEL4WS (Business Process Execution Language for Web Services – язык реализации бизнес-процессов для web-служб), разработанных для выполнения бизнес-процессов.

Цель BPMN – стандартизировать нотацию моделирования бизнес-процессов при наличии множества различных нотаций и точек зрения на моделирование. Использование BPMN обеспечит легкую передачу информации

по процессам другим участникам бизнес-сферы, специалистам по внедрению процессов, клиентам и поставщикам.

Рабочая группа BPMI (Business Process Management Initiative) проанализировала опыт многих существующих нотаций и попробовала объединить лучшие концепции разных нотаций в одну стандартную нотацию. Были рассмотрены следующие нотации и методологии: UML Activity Diagram, UML EDOC Business Processes, IDEF, ebXML BPSS, Activity-Decision Flow (ADF) Diagram, RosettaNet, LOVeM, and Event-Process Chains (EPCs). Результатом работы стала нотация BPMN.

1.2. Отображение BPMN на другие языки моделирования

Так как BPMN охватывает широкий спектр применения, то существует возможность *отображения BPMN на нескольких языках спецификаций*.

BPEL – основной язык отображения BPMN, но он охватывает только один выполняемый частный бизнес-процесс. Если на схеме отображения BPMN показано более одного внутреннего бизнес-процесса, то для каждого из них будет отдельное отображение.

Абстрактные участки схемы отображения BPMN переводятся в описания интерфейса web-сервиса.

Участки модели глобальной работы BPMN могут быть отображены при помощи моделей совместной работы, например, ebXML BPSS, Rosetta-Net и спецификации W3C Choreography Working Group Specification.

В данном разделе будет рассматриваться только отображение на язык BPEL.

1.3. Основные элементы языка описания бизнес-процессов

Одним из факторов развития BPMN является создание простого механизма для разработки моделей бизнес-процессов, который способен управлять сложными бизнес-процессами. Способ решения проблемы этих двух противоречивых требований состоял в создании графических элементов нотации по конкретным категориям. При этом совокупность категорий нотации получается небольшая, таким образом, читатель схемы отображения BPMN может легко узнать основные типы элементов и понять схему. В рамках основных категорий элементов могут быть внесены дополнительные изменения и информация для обеспечения соответствия требованиям сложности без значительных модификаций основной схемы отображения. Выделяют пять основных категорий, элементами которых являются:

1. Поток объектов (Objects Flow). Содержит основные графические элементы для определения бизнес-процесса: Event (событие), Activity (деятельность), Gateway (узел, шлюз).

2. Данные (Data). Представлены четырьмя элементами: Data Objects (объекты данных), Data Inputs (входные данные), Data Outputs (выходные данные), Data Stores (хранилища данных).

3. Соединители объектов (Objects Connectors). Соединяют один к одному объекты или информацию. Выделяют четыре типа соединений: Sequence Flow (последовательный поток), Message Flow (поток сообщений), Association (ассоциация), Data Association (ассоциация данных).

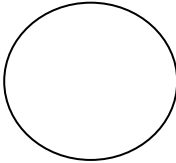
4. Области, дорожки (Swimlanes). Выделяют два способа группировки первичных данных моделирования: Pools (пулы), Lanes (дорожки).


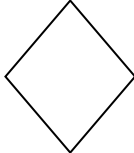
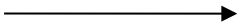
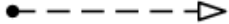

5.Arteфакты (Artifacts). Они используются для указания дополнительной информации о процессе и включают: Group (группу), Text Annotation (текстовую аннотацию). Разработчик или инструмент моделирования могут добавить столько артефактов, сколько требуется. BPMN может предпринять дополнительную попытку привести большую совокупность артефактов к единому стандарту для общего использования или для вертикальных рынков.




В табл. 1 приведены основные элементы моделирования BPMN.

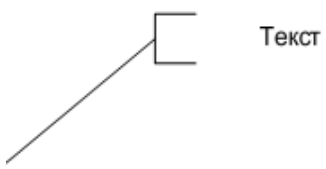
Таблица 1

Основные элементы моделирования

Элемент	Описание	Нотация
1	2	3
Event (событие)	Нечто, что «происходит» в ходе бизнес-процесса. События влияют на ход бизнес-процесса и обычно имеют причину (триггер) или воздействие (результат). На схеме события обозначают кругами с открытым центром для обеспечения возможности внутренним маркерам различать разные триггеры или результаты. Существует три типа событий, классифицированных по времени воздействия на ход процесса: начало, промежуточные события и конец	

1	2	3
Activity (деятельность)	Термин, характерный для обозначения работы. Деятельность может быть элементарной и неэлементарной (составной). Типы деятельности, являющиеся частью модели процесса: процесс, подпроцесс и задача. Задачи и подпроцессы – закругленные прямоугольники. Процессы либо безграничны, либо содержатся в пределах области	
Gateway (узел, шлюз)	Используются для контроля расхождения и схождения последовательного потока. Таким образом, узел будет обозначать ветвление, раздвоение, слияние и соединение маршрутов. Внутренние маркеры будут указывать на тип контроля развития процесса	
Sequence Flow (последовательный поток)	Показывает порядок, согласно которому будут выполняться действия процесса	
Message Flow (поток сообщений)	Символизирует поток сообщений между двумя участниками, готовыми к их отправке и получению. В BPMN две отдельные области на диаграмме будут символизировать двух участников (например, бизнес-объекты или бизнес-роли)	
Association (ассоциация)	Используется для связывания информации и артефактов с объектами. Текстовые и графические объекты, не относящиеся к схеме, могут быть связаны с объектами схемы	

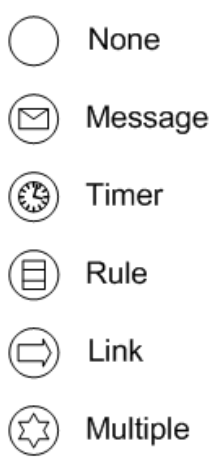

1	2	3			
Pool (пул)	Область представляет собой участника процесса. Она также играет роль «дорожки» и графического контейнера для разделения совокупности действий из других областей, обычно в контексте ситуаций «бизнес для бизнеса»	<div style="border: 1px solid black; width: 100%; height: 100%; display: flex; align-items: center; justify-content: center;"> <p style="writing-mode: vertical-rl; transform: rotate(180deg);">Наименование</p> </div>			
Lane (дорожка)	Подраздел, в пределах области которого протяженность равна длине области как по вертикали, так и по горизонтали. Дорожки организуют и классифицируют действия	<table border="1" style="width: 100%; height: 100%; border-collapse: collapse;"> <tr> <td rowspan="2" style="width: 30%; text-align: center; vertical-align: middle;">Наименование</td> <td style="width: 70%; text-align: center;">Наименование</td> </tr> <tr> <td style="text-align: center;">Наименование</td> </tr> </table>	Наименование	Наименование	Наименование
Наименование	Наименование				
	Наименование				
Message (сообщение)	Используется для передачи контента изображения между двумя участниками бизнес-процесса				
Data Objects (объекты данных)	Рассматриваются как артефакты, так как они не влияют непосредственно на последовательный поток или поток сообщений процесса, но обеспечивают ввод информации о том, какие действия требуют выполнения и/или что они производят				
Group (группа, прямоугольник вокруг группы объектов для документирования)	Группировка действий, не оказывающих влияние на последовательный поток. Группы могут использоваться для распознавания действий операции, распределенной по ширине области				

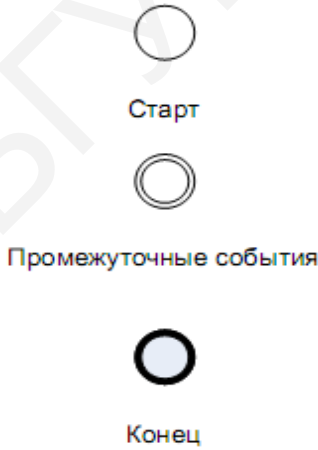

1	2	3
Text Annotation (текстовая аннотация, связанная с ассоциацией)	Способ предоставления дополнительной информации для изучающего схему BPMN	

В табл. 2 приведен список дополнительных элементов схемы моделирования бизнес-процессов, которые могут быть описаны посредством нотации моделирования бизнес-процессов (BPMN).











Таблица 2



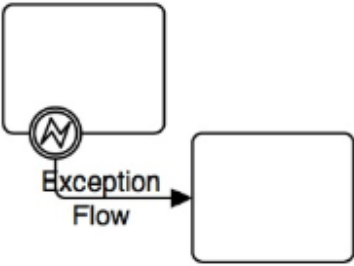
Перечень дополнительных элементов моделирования

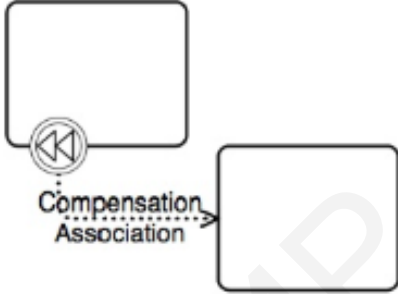

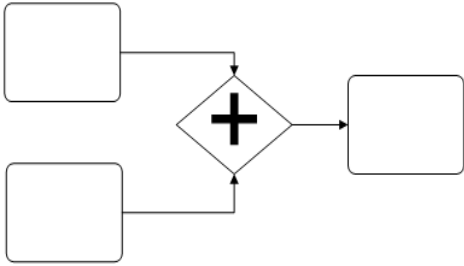
Элемент	Описание	Нотация
1	2	3
Типы триггеров событий (Сообщение, Таймер, Правило, Составное, Ошибка, Коррекция, Ссылка, Завершение и др.)	«Старт» и большинство промежуточных процессов обладают «триггерами», определяющими причину события. Есть много способов инициировать процесс. Событие «Конец» может определять «Результат», следствие завершения последовательного потока BPMN	
Задача (элементарное действие)	Элементарное действие в пределах процесса. Задача используется, когда работа в процессе не может быть разбита на составляющие	

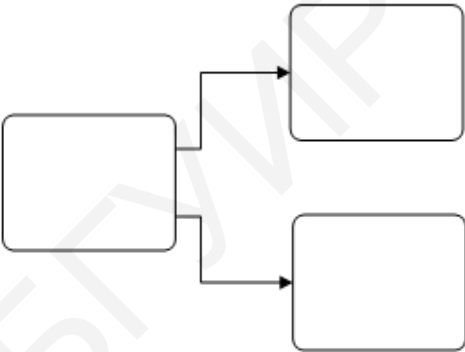
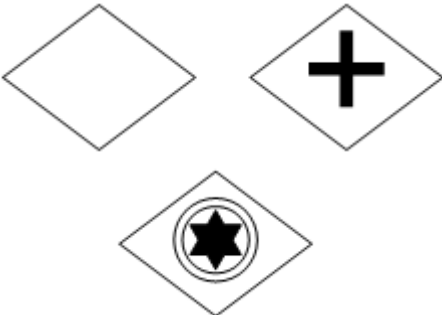
1	2	3
<p>Flow Dimensions (виды или категории потока событий)</p>	<p>Событие «Старт» указывает место, где начинается данный процесс.</p> <p>Событие «Старт» может представлять собой: пустое событие начала, сообщение, таймер, правило, связь, сложные элементы.</p> <p>Промежуточные события находятся между событиями «Старт» и «Конец». Они влияют на ход процесса, но не являются непосредственно началом или концом процесса.</p> <p>Промежуточные события могут представлять собой: пустое промежуточное событие, сообщение, таймер, ошибку, отмену, компенсацию, правило, связь, сложные элементы.</p> <p>Событие «Конец» указывает, где заканчивается процесс.</p> <p>Событие «Конец» может представлять собой: пустое конечное событие, сообщение, ошибку, отмену, компенсацию, связь, завершение, сложные элементы</p>	 <p>Start</p> <p>Промежуточные события</p> <p>Конец</p>
<p>Деятельность</p>	<p>Может быть элементарной и неэлементарной (составной)</p>	

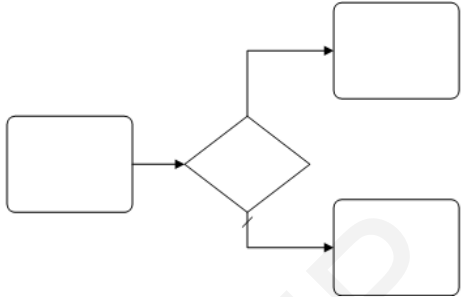
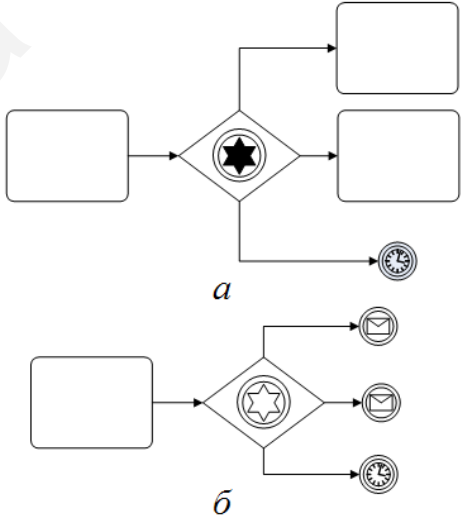
1	2	3
<p>Сжатый подпроцесс</p>	<p>Подпроцесс – это сложное действие в пределах процесса. Он может быть разбит на составляющие при помощи совокупности поддействий. Процессы либо безграничны, либо содержатся в пределах области. Элементы подпроцесса не видны на схеме. Знак «плюс» в середине нижней части блока означает, что действие – подпроцесс, и его элементы находятся уровнем ниже</p>	
<p>Расширенный подпроцесс</p>	<p>Границы подпроцесса раздвигаются и элементы (процесса) видны в пределах его границ. Следует отметить, что последовательный поток не может пересекать границы подпроцесса</p>	
<p>Collapsed Sub-Choreography (сжатый подпроцесс хореографии)</p>	<p>Знак «плюс» в нижней центральной части фигуры указывает, что «Активность» является подпроцессом и имеет более низкий уровень детализации</p>	

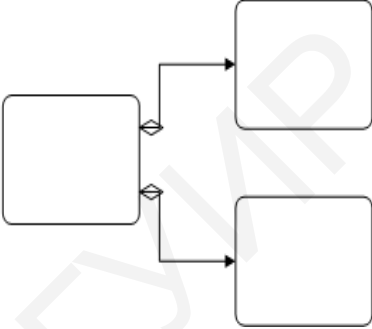
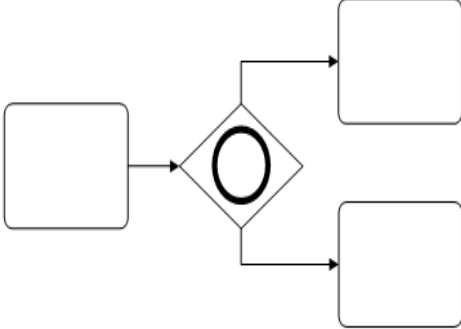
1	2	3
<p>Типы контроля узлов</p>	<p>Значки внутри ромба обозначают тип контроля потока:</p> <p><i>Исключающее ИЛИ</i> – исключаяющее решение и слияние. Может быть основанным как на данных, так и на событиях. Основанный на данных может изображаться с маркером X или без него.</p> <p><i>ИЛИ</i> – включающее решение и слияние.</p> <p><i>Сложный</i> – сложные условия и ситуации (например, 3 из 5 потоков).</p> <p><i>И</i> – раздвоение и соединение. Каждый тип контроля оказывает влияние как на входящие, так и на исходящие потоки</p>	<p>Exclusive  or </p> <p>Exclusive Event-Based  </p> <p>Parallel Event-Based </p> <p>Inclusive </p> <p>Complex </p> <p>Parallel </p>
<p>Стандартный поток</p>	<p>Поток, берущий начало от события «Старт» и идущий по действиям через альтернативные и параллельные маршруты до своего завершения в событии «Конец»</p>	
<p>Неконтролируемый поток</p>	<p>Поток, не подверженный влиянию каких-либо условий или не проходящий через объединение. Простейший пример – единичный последовательный поток, соединяющий два действия</p>	

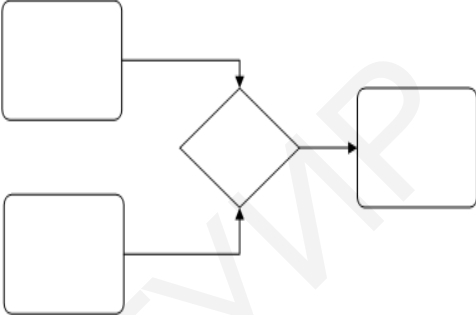
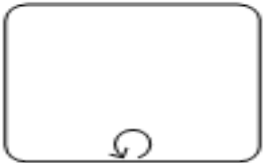
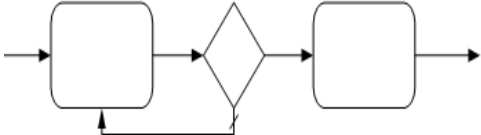
1	2	3
<p>Условный поток</p>	<p>Последовательный поток может иметь условные выражения, которые измеряются по времени выполнения с целью определить, будет ли использоваться поток. Если условный поток исходит от действия, то в начале линии последовательного потока будет ромбик (рисунок в колонке 3). Если условный поток выходит из соединения, то на линии не будет ромбика (рисунок в колонке 3 предыдущей строки)</p>	
<p>Условный поток по умолчанию</p>	<p>Как для исключающих, так и для включающих ветвлений характерен один тип потока – условный поток по умолчанию. Данный поток будет использоваться в случае, если все другие условные потоки неверны при выполнении. В начале линии у таких последовательных потоков будет диагональная черта (рисунок в колонке 3)</p>	
<p>Исключающий поток</p>	<p>Находится вне стандартного потока процесса и основывается на промежуточном событии, имеющем место при выполнении процесса</p>	



1	2	3
Компенсирующая ассоциация	Находится вне стандартного потока процесса и основывается на событии (промежуточное событие «Отмена»), которое инициируется через сбой в групповой операции, или событии «Компенсация». Цель ассоциации – действие «Компенсация»	
Объекты данных	Рассматриваются как артефакты, так как они не влияют непосредственно на последовательный поток или поток сообщений процесса, но обеспечивают информацию о том, какие действия требуют выполнения и/или что они производят	
И-соединение	BPMN использует термин «соединение» относительно соединения двух или более параллельных маршрутов в один маршрут (также известно как «И-соединение», или синхронизация). Параллельное объединение показывает соединение множественных потоков	


1	2	3
<p>Раздвоение (И-разбиение)</p>	<p>ВPMN использует термин «раздвоение» относительно деления маршрута на два или более параллельных маршрута (также известно как «И-разбиение»). Это участок процесса, где действия могут выполняться одновременно, а не последовательно. Существует два варианта «раздвоения»: множественный исходящий последовательный поток и параллельное объединение. Первый способ является предпочтительным для большинства ситуаций. Второй способ используется редко, обычно в сочетании с другими объединениями</p>	
<p>Исключающие объединения</p>	<p>Исключающее объединение (XOR) ограничивает поток таким образом, что во время выполнения можно выбрать только одну группу альтернативных маршрутов. Существует два типа исключающих объединений: основанные на данных и основанные на событиях</p>	

1	2	3
Объединения, основанные на данных	Ветвление представляет собой точку, где альтернативные маршруты основаны на условных выражениях, содержащихся в исходящем последовательном потоке, и может быть выбран только один из альтернативных маршрутов	
Объединения, основанные на событиях	Ветвление представляет собой точку, где альтернативные маршруты основываются на событии, имеющем место в данной точке процесса. Событие – обычно это получение сообщения – определяет маршрут. Могут использоваться другие типы событий, например, таймер. Можно выбрать только один из альтернативных маршрутов. Существует два способа получения сообщений: «Задачи» с типом «Получение» (рисунок <i>а</i> в колонке 3), промежуточные события с типом «Сообщение» (рисунок <i>б</i> в колонке 3)	

1	2	3
<p>Включающие объединения</p>	<p>Ветвление представляет собой точку, где альтернативные маршруты основываются на условных выражениях, содержащихся в пределах исходящего последовательного потока. В некотором смысле это группировка связанных независимых двойных (Да/Нет) ветвлений. Так как каждый маршрут независим, могут быть задействованы все комбинации маршрутов – от нуля до бесконечности. Однако ветвление должно быть построено так, чтобы был задействован хотя бы один маршрут. С целью проверки наличия хотя бы одного маршрута может использоваться условие по умолчанию. Существует два варианта данного типа ветвления: первый тип использует совокупность условных последовательных потоков, отмеченных ромбиками (рисунок <i>а</i> в колонке 3). Второй тип использует объединение OR (ИЛИ), обычно в сочетании с другими объединениями (рисунок <i>б</i> в колонке 3)</p>	 <p style="text-align: center;"><i>а</i></p>  <p style="text-align: center;"><i>б</i></p>

1	2	3
<p>Слияние (ИЛИ-соединение)</p>	<p>ВPMN использует термин «слияние» в отношении исключającego сочетания двух или более маршрутов в один (также известный как ИЛИ-соединение). Объединение «слияния» (XOR) показывает соединение множественных потоков. Если все входящие потоки альтернативные, то в объединении нет необходимости. Таким образом, неконтролируемый поток обеспечивает аналогичное развитие</p>	
<p>Цикличность действия</p>	<p>Атрибуты задач и подпроцессов определяют, носят ли они повторяющийся характер или выполнены один раз. Существует два типа циклов: стандартный и многовариантный. В середине нижней части действия – небольшой значок цикличности</p>	
<p>Цикличность последовательного потока</p>	<p>Циклы могут создаваться путем соединения последовательного потока с каким-либо «противоположным» объектом</p>	

1	2	3
<p>Перерыв в процессе</p>	<p>Участок процесса, где произойдет предполагаемая задержка. Промежуточное событие показывает фактическое развитие процесса. Кроме того, артефакт перерыва в процессе, исходя из проектирования разработчиком или инструментом моделирования, может быть связан с событием для указания места задержки в пределах потока</p>	
<p>Вложенный (встроенный) подпроцесс</p>	<p>Действие, обладающее тем же набором данных, что и его родительский процесс. Ему противопоставляется процесс независимый, повторно используемый, идущий от родительского процесса. Данные должны быть переданы основному, а не вложенному подпроцессу</p>	<p>Для вложенных подпроцессов нет специального обозначения</p>
<p>Группа (блок вокруг группы объектов в целях документирования)</p>	<p>Группировка действий, не оказывающая влияние на последовательный поток. Может использоваться для документирования или анализа, а также для обозначения действий и распределенных групповых операций, расположенных по ширине областей</p>	

1	2	3
Соединитель страниц	<p>Данный объект обычно используется при печати и указывает место, где заканчивается последовательный поток на одной странице и начинается на следующей странице. В качестве соединителя страниц может использоваться промежуточное событие «Связь»</p>	

1.4. Бизнес-интеграция

В данном подразделе описывается процесс разработки бизнес-процессов с помощью инструмента IBM WebSphere Business Integration [4]. Это не значит, что другие компании в своем арсенале не имеют аналогичных инструментов. Так, Oracle BPEL Designer [3] может выполнять аналогичные функции.

Если высокоуровневый процесс смоделирован на BPMN, он прежде всего экспортируется к скелетному (Skeletal) BPEL-процессу, который, как правило, состоит из областей действия процесса (Process Scopes), действий по вызову/получению (Invoke/Receive Activities) и партнерских связей с соответствующими сервисами (Partner Links to the Appropriate Services).

Подход к разработке информационных систем с максимальным акцентом на бизнес (или интеграцией с бизнесом) может быть внедрен с целью повышения эффективности создания и эксплуатации информационных систем, автоматизирующих различные направления деятельности организации, а также повышения эффективности самого бизнеса, который базируется на данных системах.

В корне данного подхода лежат *современные принципы построения сервис-ориентированной архитектуры* (Service Oriented Architecture, далее – SOA) [5] с использованием инструментов IBM WebSphere Business Integration [4].

Основные проблемы, с которыми сталкиваются организации и подразделения, занимающиеся автоматизацией достаточно сложного и распределенного бизнеса:

- увеличение числа разносторонних эксплуатируемых систем в организации, что создает беспрецедентную сложность;
- широкое число технологий и поставщиков решений, создающих хаотическое нагромождение систем, каждая из которых действует по своим собственным законам и правилам в заранее predeterminedных замкнутых пространствах;
- огромные затраты на сопровождение большого количества систем, что приводит к необходимости содержать разноплановых специалистов;
- длительный возврат инвестиций при внедрении большинства систем;
- дублирование реализаций бизнес-требований или отсутствие их реализаций;
- запаздывание с автоматизацией необходимых направлений.

Эти и многие другие проблемы могут быть в значительной мере преодолены благодаря внедрению современных унифицированных подходов к разработке информационных систем, базирующихся на принципах SOA и позволяющих максимально полно учитывать потребности бизнеса при создании средств автоматизации.

Кроме того, после создания и развертывания этих средств указанная технология позволит получать оперативную и точную информацию о текущем состоянии бизнеса для своевременного внесения необходимых корректив в развернутые средства автоматизации.

Эта технология представляет собой подход к разработке информационных систем, называемый бизнес-интеграцией на базе SOA (далее – бизнес-интеграция).

Бизнес-интеграция дает возможность реализовать следующие ключевые преимущества:

- *повысить способность бизнеса реагировать на изменение бизнес-условий* благодаря автоматизации процессов отслеживания событий, которые происходят в бизнесе, и отображения наглядных форм предоставления необходимой информации для ответственных лиц;
- *позволить бизнесу расти и динамично развиваться* благодаря максимальной синхронизации бизнес-процессов и средств автоматизации;

- *снизить затраты бизнеса* благодаря возможности интегрировать существующие решения в единое информационное пространство в одной или нескольких организациях, а также максимально эффективно использовать существующие наработки, реализованные, возможно, на самых разных технологических платформах;
- *снизить бизнес-риски* благодаря получению оперативной информации о бизнесе и возможности быстро предпринять необходимые действия для заблаговременного устранения потенциальных проблем и своевременного принятия грамотных решений.

1.5. Жизненный цикл разработки приложений, интегрированных с бизнесом

Ведение бизнеса в современных условиях с высоким уровнем конкуренции вынуждает организации адаптироваться к этим условиям и стремиться к повышению собственной конкурентноспособности.

Реализовав модель деятельности «бизнес по требованию», в которой в единую систему объединены все внутренние бизнес-процессы (и при этом они интегрированы с бизнес-процессами ключевых партнеров и клиентов), организация может получить значительные преимущества перед конкурентами. Стратегическое преимущество модели «бизнес по требованию» состоит в оптимизации и эффективной автоматизации бизнес-процессов.

Для формирования модели «бизнес по требованию» необходимо сосредоточиться на *двух аспектах: построении бизнес-архитектуры (моделировании бизнес-процессов) и базовой ИТ-инфраструктуры (реализации этих бизнес-процессов).*

Бизнес-архитектура должна отражать важнейшие цели и приоритеты компании. Подобную задачу нетрудно поставить, но непросто решить. *Необходимы инструментальные средства, с помощью которых сложные процессы можно представить в достаточно наглядной форме, чтобы «узкие места» и дублирование функций сразу стали очевидными.*

Но одной оптимизации бизнес-процессов недостаточно. Необходимо проанализировать возможности ИТ-инфраструктуры: способна ли она обеспечить автоматизацию бизнес-процессов, а также ответить на вопросы, насколько гибкой она окажется в будущем, когда бизнес-процессы потребуются оперативно корректировать. Такой динамический подход к моделированию деятельности организации и управлению ею невозможен без соответствующей адаптации ИТ-инфраструктуры.

Гарантом такой адаптации являются современные технологии SOA, которые могут обеспечить необходимую степень гибкости. SOA и web-сервисы становятся стандартами [6], на основе которых создаются большие интеграционные приложения. Применение SOA удешевляет развитие и эксплуатацию информационных систем, однако главное преимущество – возможность строить гибкие решения.

Смысл SOA состоит в преобразовании монолитной IT-инфраструктуры во множество унифицированных, многократно используемых сервисных компонентов, из которых можно быстро собирать необходимые для бизнеса решения.

Благодаря технологиям IBM WebSphere, зрелости стандарта описания бизнес-процессов BPEL и гибкости спецификаций web-сервисов WSDL интеграция процессов, основанная на SOA, становится реальностью, сервисные компоненты становятся строительными блоками, из которых собираются приложения, обеспечивающие информационное взаимодействие как внутри компании, так и с внешними партнерами и контрагентами.

Управление бизнес-процессами с использованием инструментальной поддержки IBM WebSphere позволяет компаниям внедрять непрерывный сквозной жизненный цикл ПО для создания и оптимизации бизнес-процессов.

Данные инструменты поддерживают следующие наиболее важные *виды деятельности* в этой области:

1. Моделирование – захват, симуляция (имитация), анализ и оптимизация бизнес-моделей с целью снижения рисков и увеличения гибкости.

2. Разработка – построение новых сервисов и формирование адаптеров к существующим приложениям, сборка и тестирование интегрированных решений.

3. Выполнение – развертывание реализованных бизнес-процессов и политик (бизнес-правил) для реализации намерений бизнеса, запуск интегрированных решений в промышленных условиях.

4. Управление – оперативное управление размещенной моделью и принятие решений на основе мониторинга развернутых бизнес-процессов с помощью метрик и предупреждений, получаемых в реальном времени от внутренних и внешних источников, а также анализ показателей эффективности бизнеса.

2. ИНСТРУМЕНТЫ IBM WEBSHERE ДЛЯ РЕАЛИЗАЦИИ БИЗНЕС-ИНТЕГРАЦИИ

Жизненный цикл разработки приложений, интегрированных с бизнесом, реализуется набором средств IBM WebSphere [7] и этапов разработки, ключевыми из которых являются следующие (рис. 1):

- IBM WebSphere Business Modeler, предназначенный для этапа моделирования;
- IBM WebSphere Integration Developer, предназначенный для этапа разработки;
- IBM WebSphere Process Server (Manager), предназначенный для этапа выполнения;
- IBM WebSphere Business Monitor, предназначенный для этапа управления.

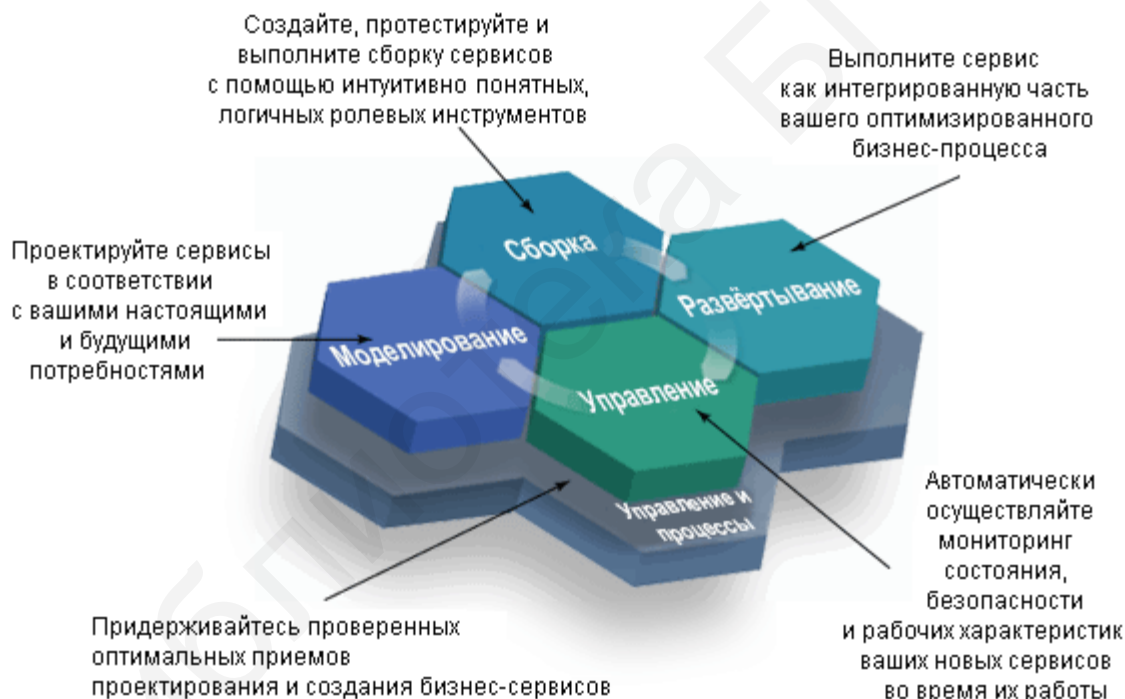


Рис. 1. Жизненный цикл разработки бизнес-приложений на базе IBM WebSphere

Кроме того, в тесной интеграции с перечисленными средствами могут быть использованы другие средства управления разработкой, такие как IBM Rational и даже средства других вендоров.

2.1. IBM WebSphere Business Modeler

Тщательное определение и моделирование бизнес-процессов является критическим фактором повышения эффективности бизнеса.

Бизнес-процессы определяются через последовательности взаимодействий, которые происходят между организационными компонентами бизнеса и окружающей средой.

Бизнес-процессы часто оказываются достаточно сложными из-за многочисленных дополнений, которые были сделаны в ответ на изменение бизнес-условий. Без формального описания процессов и системы управления процессами эта сложность накапливается и может перегрузить организацию лишними помехами и узкими местами, снижающими эффективность бизнеса. Тщательно сформированная модель бизнес-процессов может помочь локализовать и устранить эту скрытую неэффективность, снизить затраты и задержки. На рис. 2 приведен пример моделирования на BPMN.

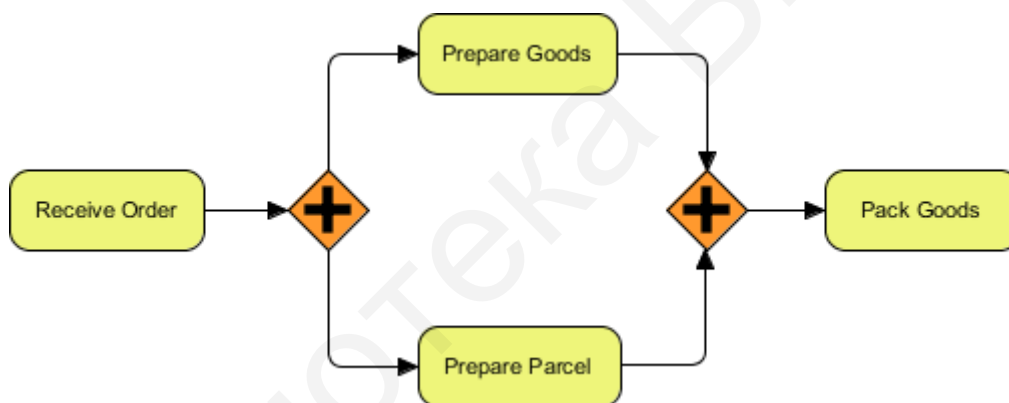


Рис. 2. Пример моделирования на BPMN

IBM WebSphere Business Modeler позволяет:

- преобразовывать описания бизнес-процессов в форму, удобную для автоматизации;
- отделять модель бизнес-процессов от лежащей в их основе реализации;
- создавать бизнес-процессы, основанные на отраслевых стандартах;
- повышать квалификацию бизнес- и IT-специалистов;
- обеспечивать взаимодействие между участниками команды;
- имитировать и анализировать бизнес-процессы с целью их улучшения.

2.2. IBM WebSphere Integration Developer

IBM WebSphere Integration Developer – это инструмент, который можно использовать для решения задач непосредственной интеграции приложений. Он позволяет создавать единое пространство из взаимодействующих информационных систем с помощью создания новых web-сервисов и подключения существующих. При этом большое число наглядных помощников (Wizards) позволяют вести разработку и подключение сервисов при минимальных навыках в разработке сервис-ориентированной архитектуры, что, в свою очередь, способствует быстрому вхождению разработчиков в работу и их обучению новейшим технологиям.

С помощью IBM WebSphere Integration Developer можно создавать сервисы для SOA и выстраивать из них практическую реализацию бизнес-процессов, описанных с помощью IBM WebSphere Business Modeler и запускаемых на IBM WebSphere Process Server. IBM WebSphere Integration Developer использует технологии, базирующиеся на стандартах, таких, как язык описания web-сервисов WSDL (Web Service Definition Language), язык описания XML-структур XSD (XML Schema Definition), язык описания бизнес-процессов BPEL (Business Process Execution Language) и т. д. Данные технологии позволяют моделировать, разрабатывать и размещать составные приложения без глубоких знаний особенностей их реализации.

IBM WebSphere Integration Developer «бесшовно» интегрируется с IBM Rational Software Architect для разработки архитектурных решений, IBM Rational Application Developer – для создания web-сервисов и IBM Rational Functional Tester for SOA – для организации функционального тестирования разработанных сервисных приложений. Собранные в единую интегрированную среду разработки (IDE), эти продукты обеспечивают полный набор инструментов для полноценной разработки сложных бизнес-приложений в области бизнес-интеграции. На рис. 3 приведен пример моделирования на BPEL.

IBM WebSphere Integration Developer предоставляет широкий набор возможностей для сборки, размещения и управления бизнес-процессами:

- импорт бизнес-моделей из WebSphere Business Modeler и их автоматическое преобразование в BPEL-стандарт для дальнейшей автоматизации бизнес-процессов;
- возможность реализации бизнес-процессов в различных вариантах и формах в составе единого интеграционного решения;

- возможности разработки web-сервисов и интеграции с существующими web-сервисами;
- возможность определения задач, выполняемых человеком (Human Task);
- возможность внедрения механизмов отслеживания бизнес-правил в информационную систему (Business Rules);
- возможность определения карт преобразований бизнес-объектов и интерфейсов при интеграции разнородных приложений, которые должны обмениваться напрямую несовместимыми данными;
- возможность реализации бизнес-процессов в виде цепочек как деятельности, так и состояний (Business State Machines);
- сервисы для импорта из ERP- и EIS-систем, таких как PeopleSoft, SAP, CICS®, and IMS™.

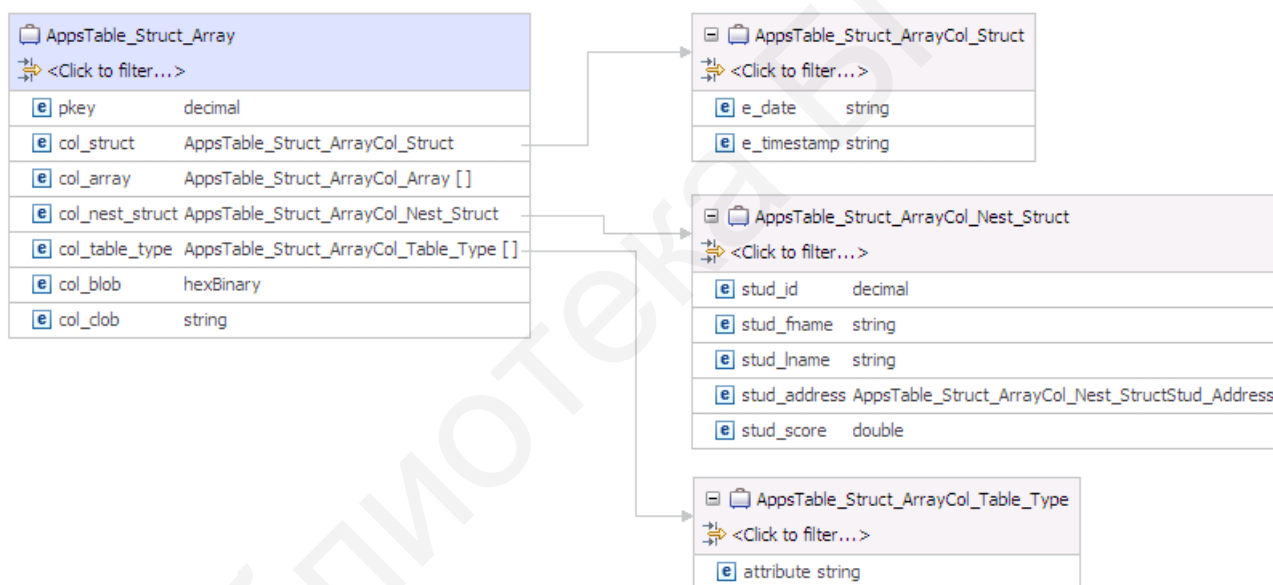


Рис. 3. Пример моделирования на BPEL

Кроме того, IBM WebSphere Integration Developer позволяет преобразовывать существующие приложения в соответствующие стандартам сервис-ориентированные приложения.

Сервисы могут быть достаточно легко собраны в единое приложение со значительным снижением обычно связанных с этими процедурами технических сложностей.

2.3. IBM WebSphere Process Server

IBM WebSphere Process Server – это интеграционный сервер [8]. Он создан для развертывания приложений бизнес-интеграции, созданных на основе сервис-ориентированной архитектуры.

IBM WebSphere Process Server основан на IBM WebSphere Application Server и на дополнительных библиотеках, реализующих технологии из семейства продуктов IBM WebSphere Business Integration.

IBM WebSphere Process Server построен на базе IBM WebSphere Application Server Network Deployment, он может использовать все преимущества последнего. Сюда входят кластеризация, высокий коэффициент надежности, встроенные возможности управления обменом сообщениями и др.

IBM WebSphere Process Server содержит три слоя:

- ядро SOA;
- вспомогательные сервисы;
- сервисные компоненты.

Каждый из этих слоев отвечает за реализацию соответствующей части бизнес-логики, реализованной по правилам компонентного подхода к разработке ПО.

2.4. IBM WebSphere Business Monitor

IBM WebSphere Business Monitor – это web-приложение, созданное на принципах клиент-серверной архитектуры, позволяющее проводить динамические измерения эффективности реального бизнеса, мониторинг развернутых бизнес-процессов, а также формировать необходимые для бизнеса отчеты [9].

Полученная информация поможет выявить проблемы бизнеса, исправить ошибки и оперативно скорректировать бизнес-процессы, чтобы достичь еще более высокой эффективности.

IBM WebSphere Business Monitor позволяет проводить мониторинг бизнес-процессов во время выполнения приложений с помощью IBM WebSphere Process Server на основе событий, правила возникновения которых определены еще на предыдущих стадиях бизнес-интеграции.

Таким образом, основываясь на данных событиях, *вычисляются ключевые показатели эффективности (KPI) и метрики бизнеса.*

Вычисленные KPI и метрики отображаются в наглядных представлениях, вид которых определен исходя из потребностей бизнеса.

IBM WebSphere Business Monitor уведомляет пользователей об инцидентах, требующих внимания, и может также выполнять некоторые корректирующие действия для устранения отказов. Он *поддерживает различные методы уведомления (предупреждение, электронная почта, мобильный телефон, пейджер и вызов сервисов) о возникающих ситуациях.*

IBM WebSphere Business Monitor выполняет мониторинг в соответствии с моделью бизнес-метрик. Такие модели создаются в редакторе бизнес-метрик, в котором могут быть указаны точки измерения и выбираемые события, определены метрики, их взаимосвязь и источник бизнес-данных. Когда модель бизнес-измерений разработана, она может быть экспортирована в IBM WebSphere Business Modeler для организации обратной связи и учета проблем, обнаруженных в бизнесе, на новом витке моделирования бизнеса в рамках нового жизненного цикла бизнес-интеграции.

Пример применения IBM WebSphere Business Monitor приведен на рис. 4.

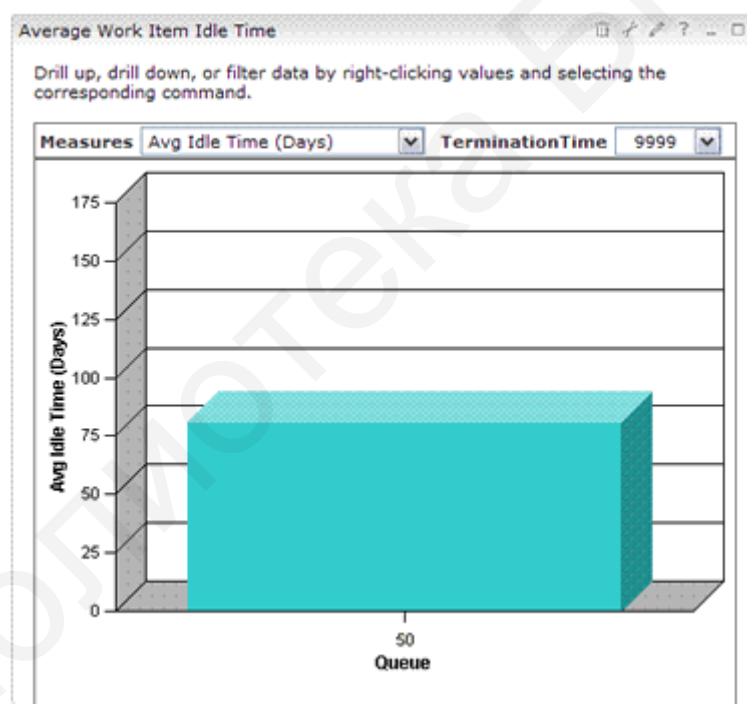


Рис. 4. Пример применения IBM WebSphere Business Monitor

Основные возможности IBM WebSphere Business Monitor:

- захват больших объемов оперативных данных на основе событий о выполненных действиях и превращение их в метрики и значения KPI;
- извлечение переменных, связанных с измерениями и отражающих реальные данные бизнеса;
- отображение измеренных значений в наглядных представлениях;

- обеспечение методов анализа данных и формирования отчетов;
- выполнение действий, связанных с оперативной коррекцией бизнеса;
- уведомление пользователей о необходимых действиях для предотвращения рисков.

2.5. Процесс реализации бизнес-интеграции

С помощью указанных в подразд. 2.4 инструментов выполняется организация полностью замкнутого жизненного цикла процесса разработки бизнес-интегрированных решений (рис. 5) – от моделирования до управления развернутыми бизнес-процессами.

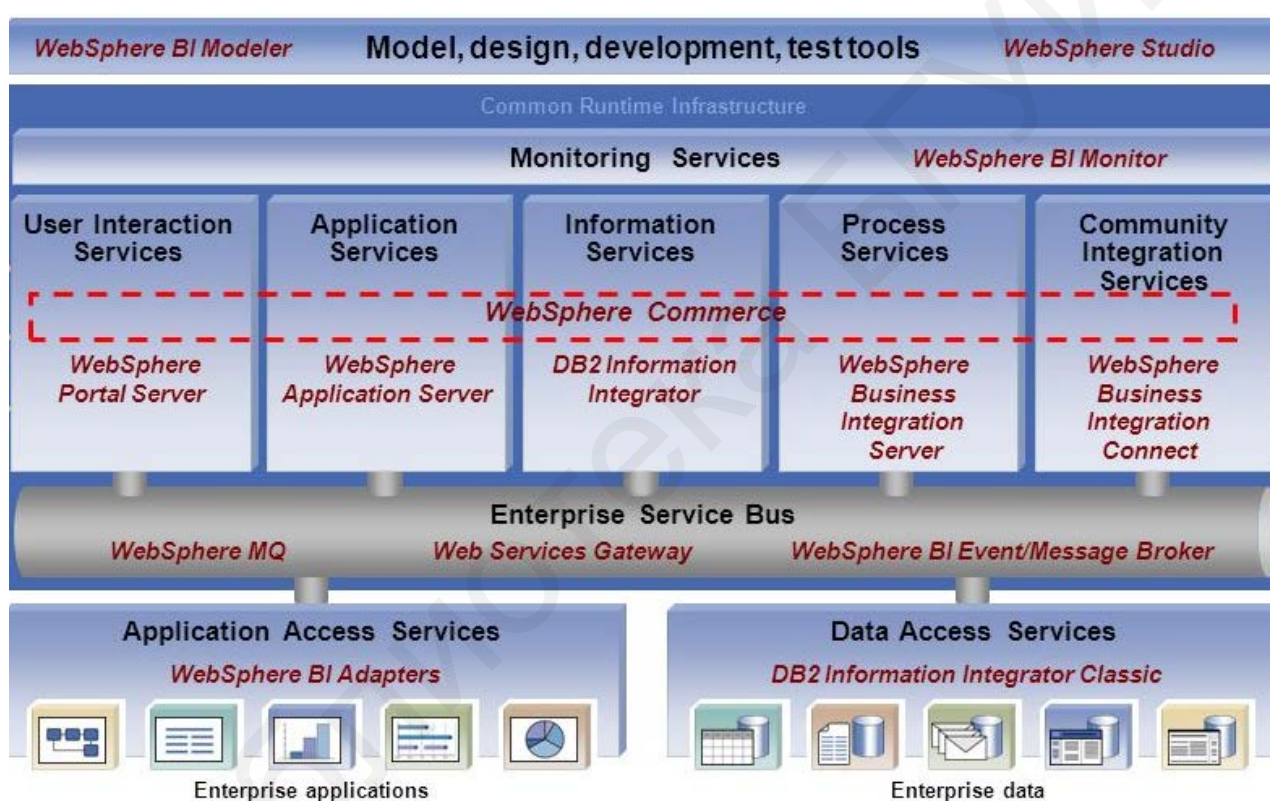


Рис. 5. Жизненный цикл процесса разработки бизнес-интегрированных решений

На шагах 1 и 2 разрабатывается модель процесса в IBM WebSphere Business Modeler. При этом проводится имитационное моделирование и оценивается длительность выполнения задач.

Шаг 3а посвящен экспорту модели процесса в IBM WebSphere Integration Developer, в котором завершается разработка приложения и процесс развертывается на IBM WebSphere Process Server.

Именно здесь для разработки новых web-сервисов используется полный жизненный цикл IBM Rational Unified Process и в полной мере интегрируются дополнительные инструменты IBM Rational, такие, как средство управления запросами на изменения IBM Rational ClearQuest, средство управления требованиями IBM Rational RequisitePro, инструмент для детальной разработки архитектурных решений IBM Rational Software Architect, средства разработчика IBM Rational Application Developer и IBM Rational Web Developer и разнообразные средства тестирования, среди которых стоит отметить инструменты для общего управления тестированием IBM Rational Quality Manager, создания и запуска функционального тестирования IBM Rational Functional Tester, а также инструмент для тестирования нагрузки и производительности IBM Rational Performance Tester.

На шаге 3б модель бизнес-измерений экспортируется из IBM WebSphere Business Modeler в формате объектов IBM WebSphere Business Monitor для размещения их на сервере монитора.

Шаг 4 использует Dashboard в составе IBM WebSphere Business Monitor для просмотра, анализа и генерации отчетов о выполнении бизнес-процессов, аналитической работы с данными бизнес-процессов, а также бизнес-метриками и KPI.

На шаге 5 цикл замыкается с помощью передачи данных обратно из IBM WebSphere Business Monitor в IBM WebSphere Business Modeler для старта нового витка жизненного цикла. Этот шаг осуществляется для непрерывного улучшения существующих бизнес-процессов.

Эта методология позволяет бизнесу согласовывать бизнес-процессы «как есть» с бизнес-процессами «как будет» на основе реальных данных о выполнении бизнес-процессов.

Эти данные можно сравнивать с результатами имитации текущих бизнес-процессов, описанных в IBM WebSphere Business Modeler, и использовать их для оптимизации существующей модели бизнес-процессов «как есть» и получения более эффективной модели «как будет».

3. БИЗНЕС-ИНТЕГРАЦИЯ. ПРИМЕР РАЗРАБОТКИ БОЛЬШОГО ПРИЛОЖЕНИЯ

Бизнес-интеграция означает интеграцию приложений, данных и процессов в рамках предприятия или среди множества предприятий.

Пример общей архитектуры предприятия, доступа клиентов к приложению «Дорога», которое взаимодействует с удаленными приложениями, приведен на рис. 6. Архитектурное решение позволяет не только реализовать конкретное приложение, но и развивать корпоративную систему предприятия.

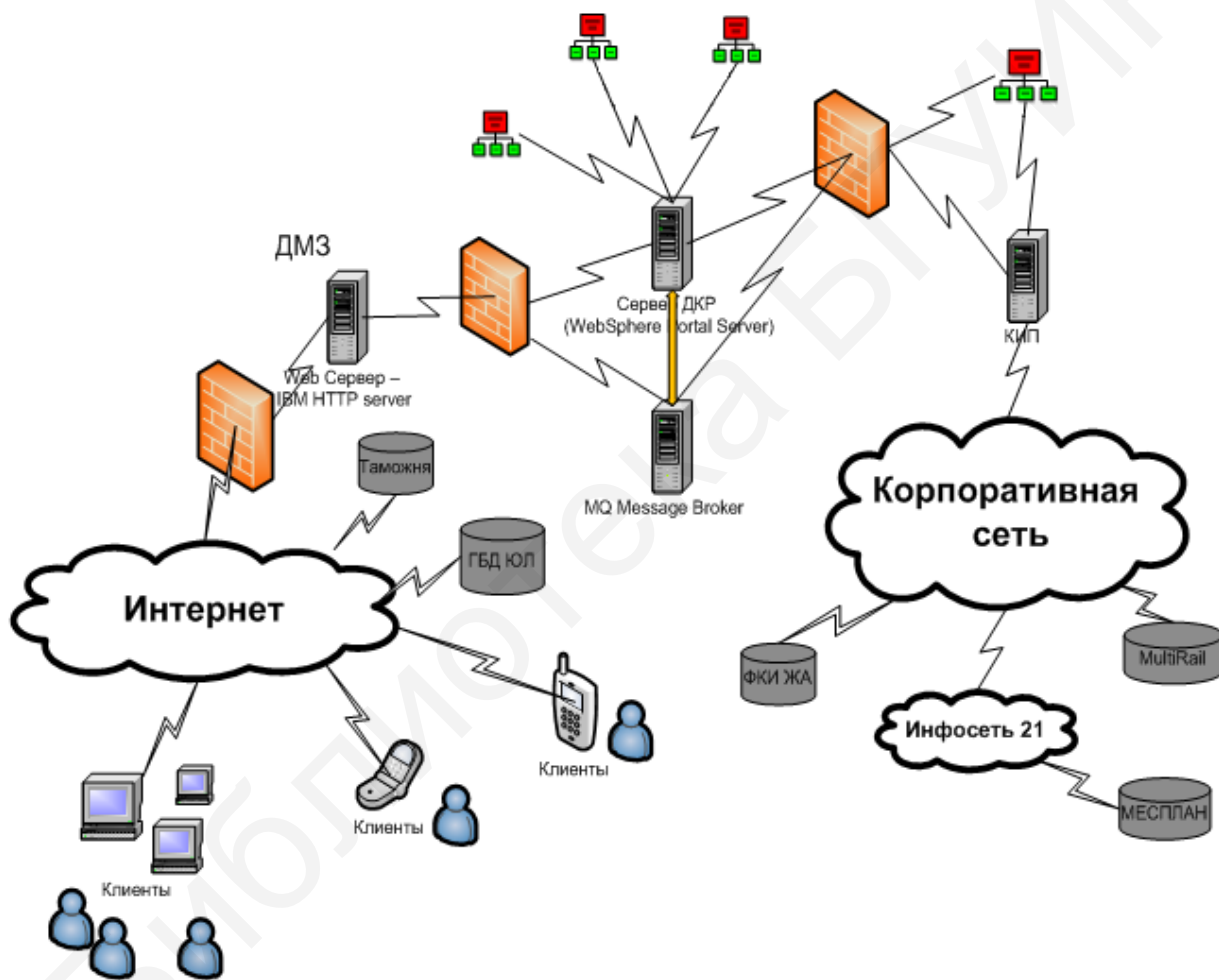


Рис. 6. Общая архитектура приложения «Дорога»

Язык реализации бизнес-процессов BPEL полностью основан на web-сервисах и XML [10], обеспечивающих открытый и гибкий способ общения в распределенной сети, задачей которого является объединение сервисов, предоставляющих доступ к среде Интернет.

Методология использования языка выполнения обеспечивает гибкость и адаптивность разработанного приложения.

Компания IBM предлагает использовать продукт IBM WebSphere Integration Developer для реализации стандартов BPEL.

3.1. Реализация бизнес-процессов для приложения «Дорога»

Разработка бизнес-процессов для компонентов «Формирование месячного портфеля заявок» и «Работа с клиентами».

Шаг 1

Для компонента «Формирование месячного портфеля заявок» на перевозку грузов разработаны следующие бизнес-процессы:

- «Формирование месячного плана»;
- «Согласование заявок ГУ – 1х межобластные перевозки»;
- «Согласование заявок ГУ – 1х экспорт К-тх»;
- «Согласование заявок ГУ – 1х экспорт через порт А-тх»;
- «Согласование заявок ГУ – 1х международные перевозки»;
- «Согласование заявок ГУ – 1х на транзит, импорт через порт А-тх»;
- «Согласование заявок ГУ – 1х на транзит, импорт из АС М-Пх»;
- «Согласование заявок ГУ – 1х через станцию М-х-мбет».

Шаг 2

Моделирование бизнес-процессов на языке BPMN и реализация сервисных компонентов. Для этого необходимо экспортировать проект из Business Modeler в Integration Developer и сгенерировать BPEL проекта.

Шаг 3

На этапе разработки проекта в Business Modeler определено, каким образом должна быть реализована задача. В Integration Developer выполняется реализация с использованием бизнес-правил, кода Java, вызова web-сервиса и менеджера задач, выполняемых человеком, и т. д.

Шаг 4

После тестирования приложения в Integration Developer выполняется имплементация на сервере Process Server.

Шаг 1 и Шаг 2

IBM WebSphere Business Modeler позволяет спроектировать бизнес-процессы на языке BPMN (рис. 7–12).

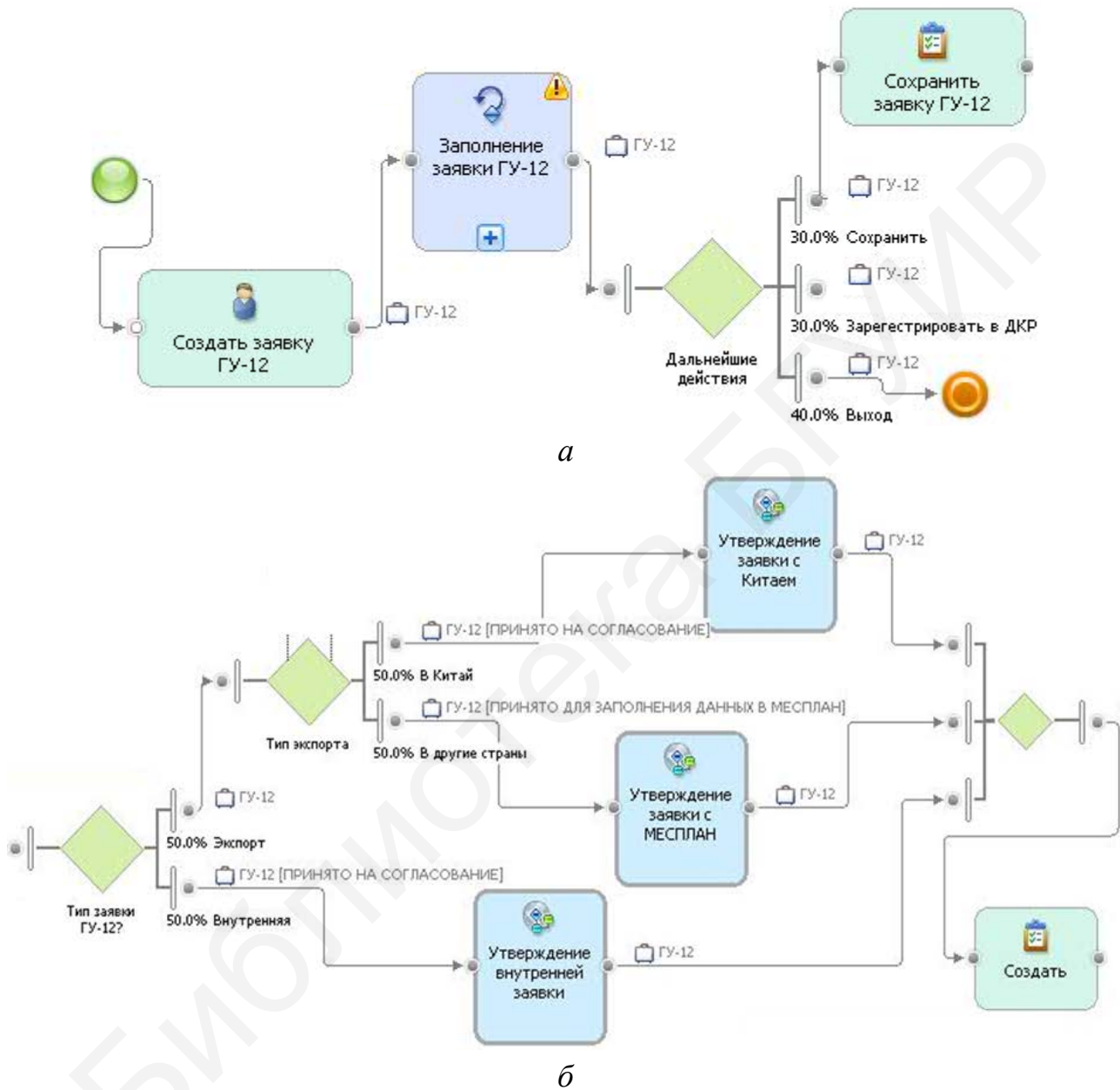


Рис. 7. Формирование месячного портфеля заявок: общий план

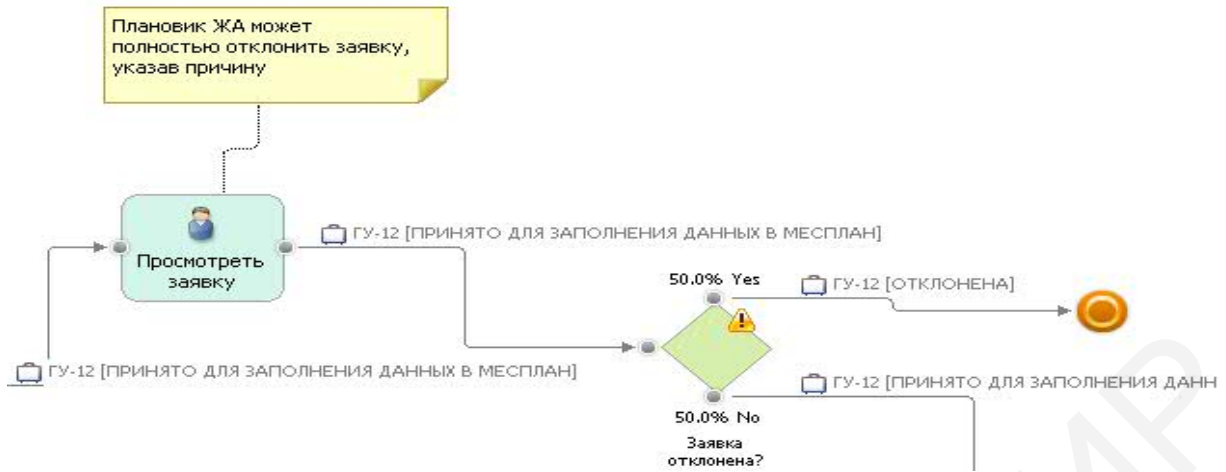


а



б

Рис. 8. Согласование внутренних перевозок



а

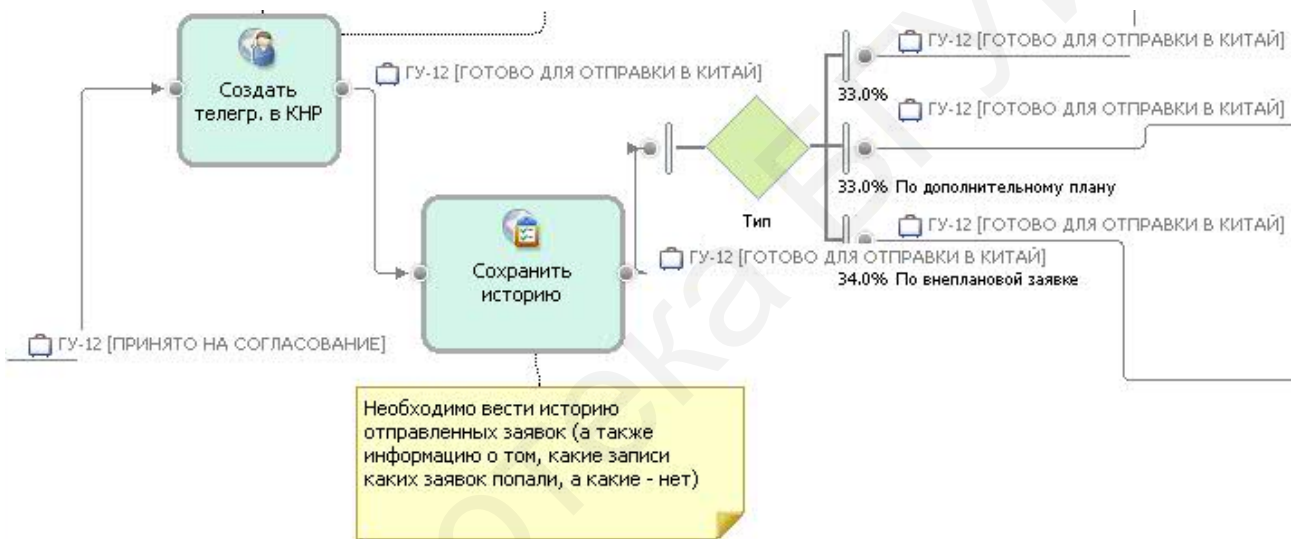


б

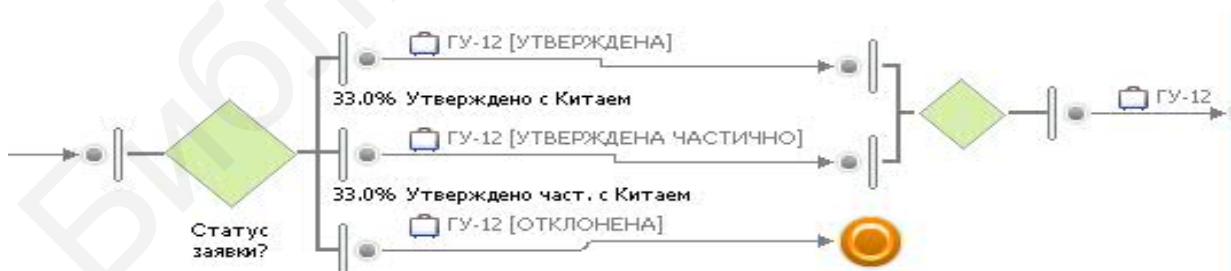
Рис. 9. Согласование экспортных перевозок. СНГ



а



б



в

Рис. 10. Согласование экспортных заявок. Китай

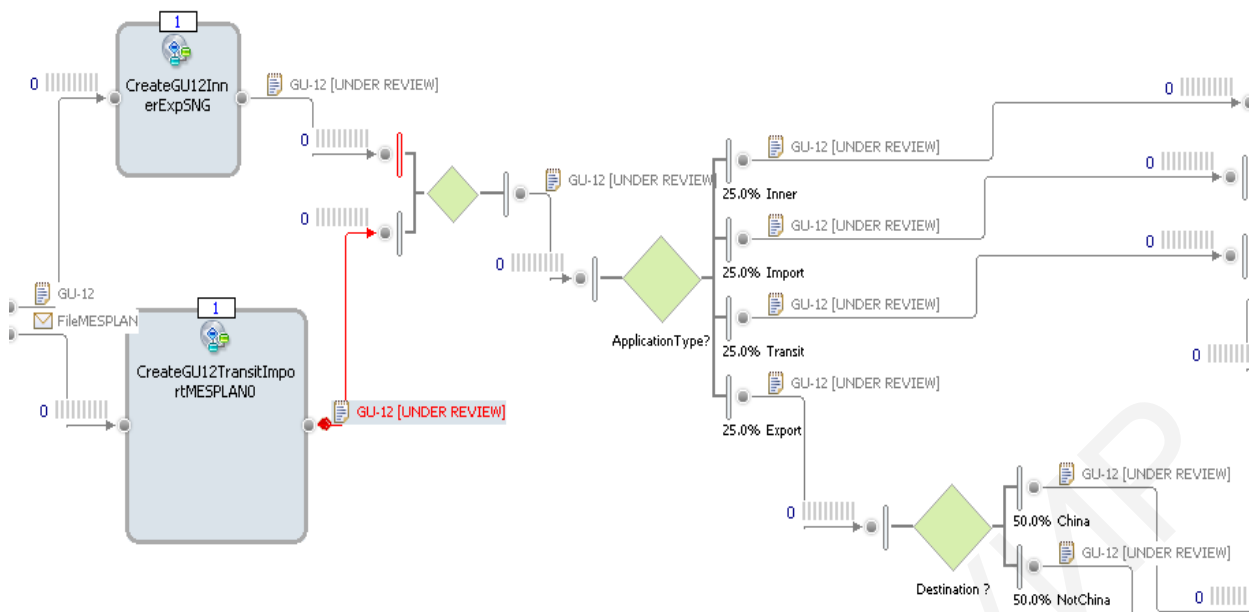


Рис. 11. Имитационное моделирование бизнес-процесса компонента «Формирование месячного портфеля заявок»

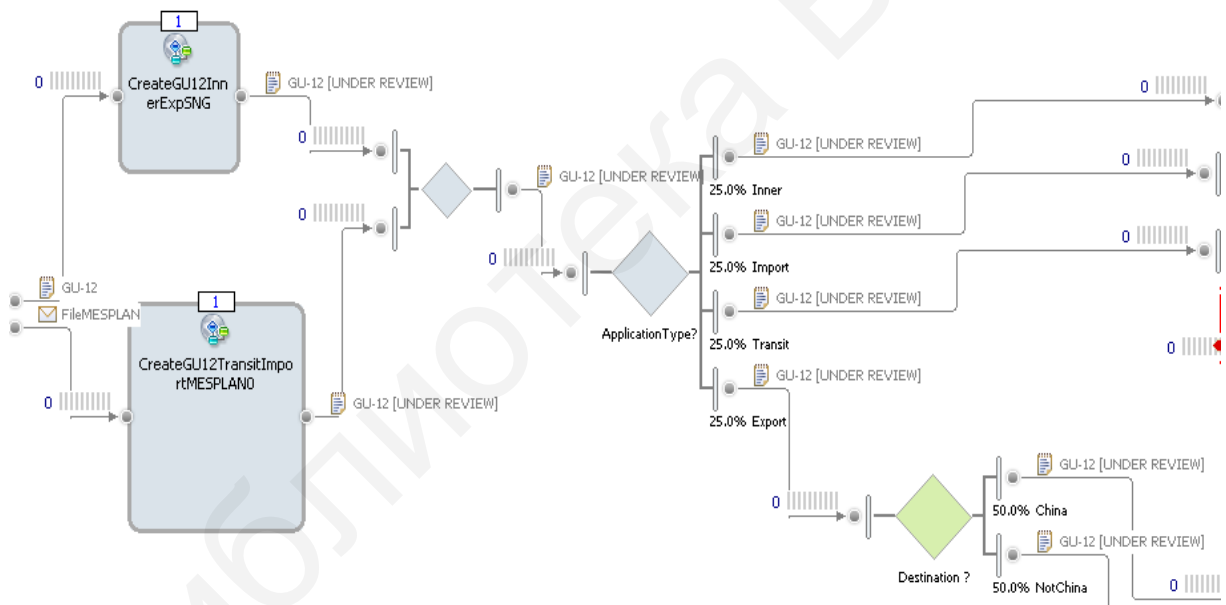


Рис. 12. Имитация процесса «Согласование ГУ-12 на транзит, импорт»

Шаг 2 и Шаг 1

Для перевода приложения на язык BPEL применяется IBM WebSphere Integration Developer (рис. 13).

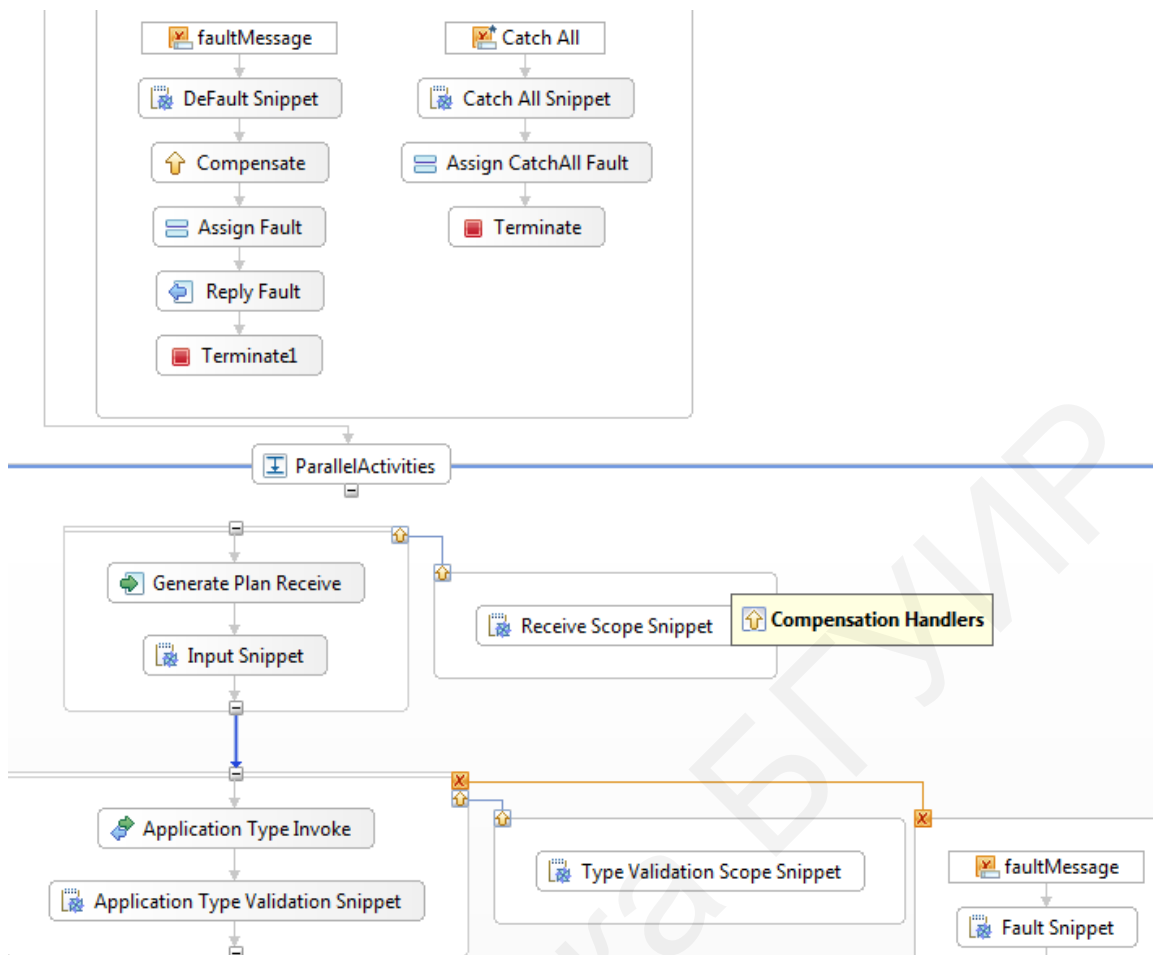


Рис. 13. Бизнес-процесс на языке BPEL

3.2. Диаграмма сборки для модуля «Формирование месячного плана»

На рис. 14 представлена диаграмма модуля «Формирование месячного плана». Можно видеть, что этот модуль является первоисточником дальнейшего следования других бизнес-процессов. Однако *диаграмма сборки не отражает последовательности действий; она показывает, что действие вызывается процессом, а также показывает, каким образом стартует процесс.*

На диаграмме модуля «Формирование месячного плана» показаны связи между бизнес-процессами и то, как бизнес-процессы переходят друг в друга. Такое представление является наглядным и позволяет при беглом взгляде на диаграмму увидеть зависимости между бизнес-процессами.

Как видно на диаграмме, один бизнес-процесс порождает множество бизнес-процессов, которые могут выполняться параллельно и являются независимыми друг от друга.

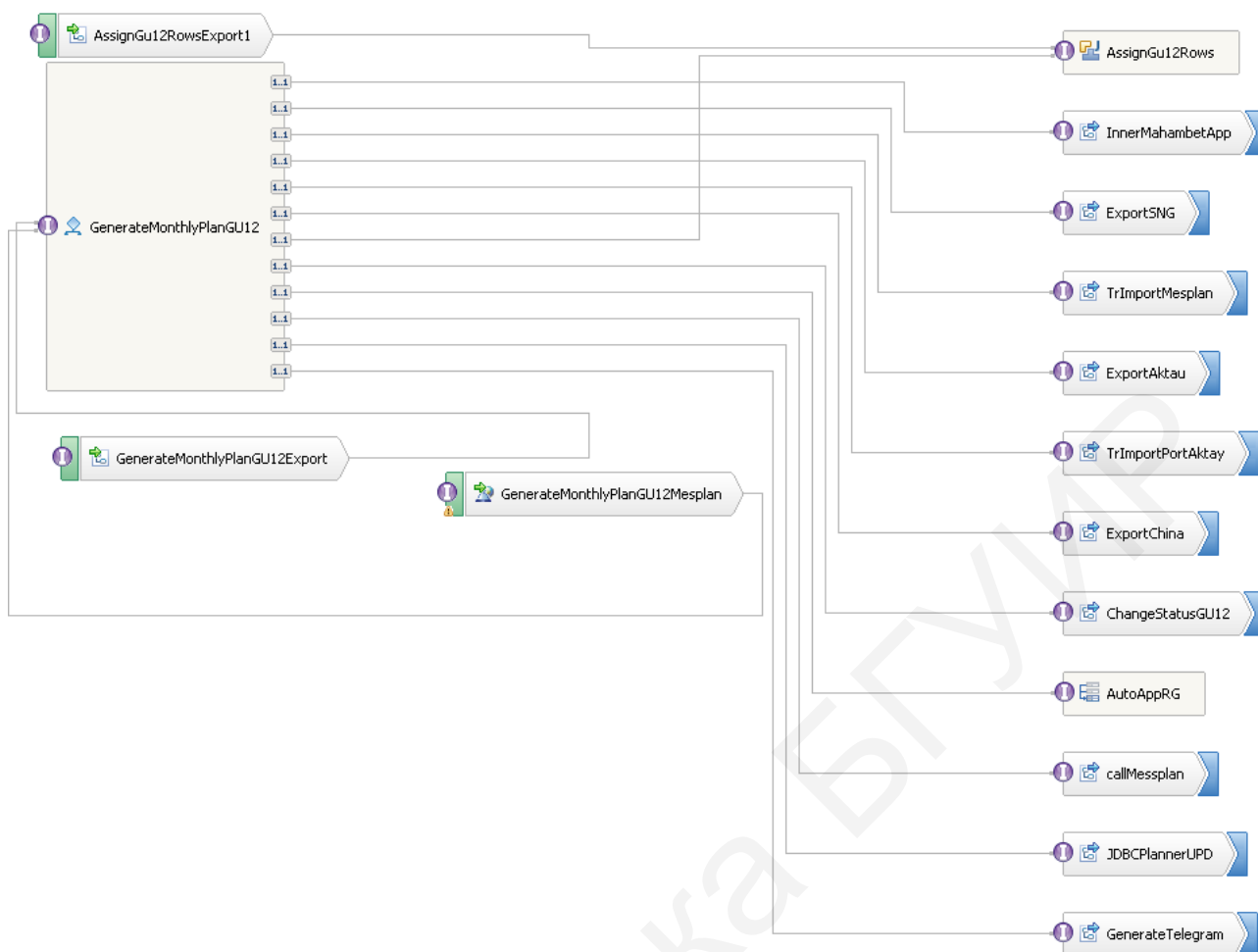


Рис. 14. Диаграмма сборки модуля «Генерация месячного плана»

Диаграмма сборки является инновационным средством создания интеграционного приложения. Она может содержать различные сервисные и SCA -компоненты:

1. Адаптеры. Предназначены для интеграции с КИС (корпоративной информационной системой), базами данных, FTP, почтовыми серверами и клиентами.

2. Поток-посредники. Являются маршрутизаторами сообщений между сервисами, конверторами, транспортными протоколами запроса и сервисом, преобразуют форматы сообщения между запросом и сервисом, работают с бизнес-событиями различных источников, являются посредниками в интеграции существующих сервисов.

3. Бизнес-правила. Их использование позволяет динамически в процессе выполнения изменять правила выполнения процесса, при этом бизнес-процесс не требуется останавливать и деинсталлировать, что обеспечивает высокую гибкость и адаптивность приложения.

4. Селекторы. Предназначены для динамического вызова различных сервисных компонентов. Селекторы играют роль маршрутизаторов, которые в ходе выполнения изменяют путь протекания бизнес-процесса.

На рис. 15 приведена общая схема SCA-компонента.

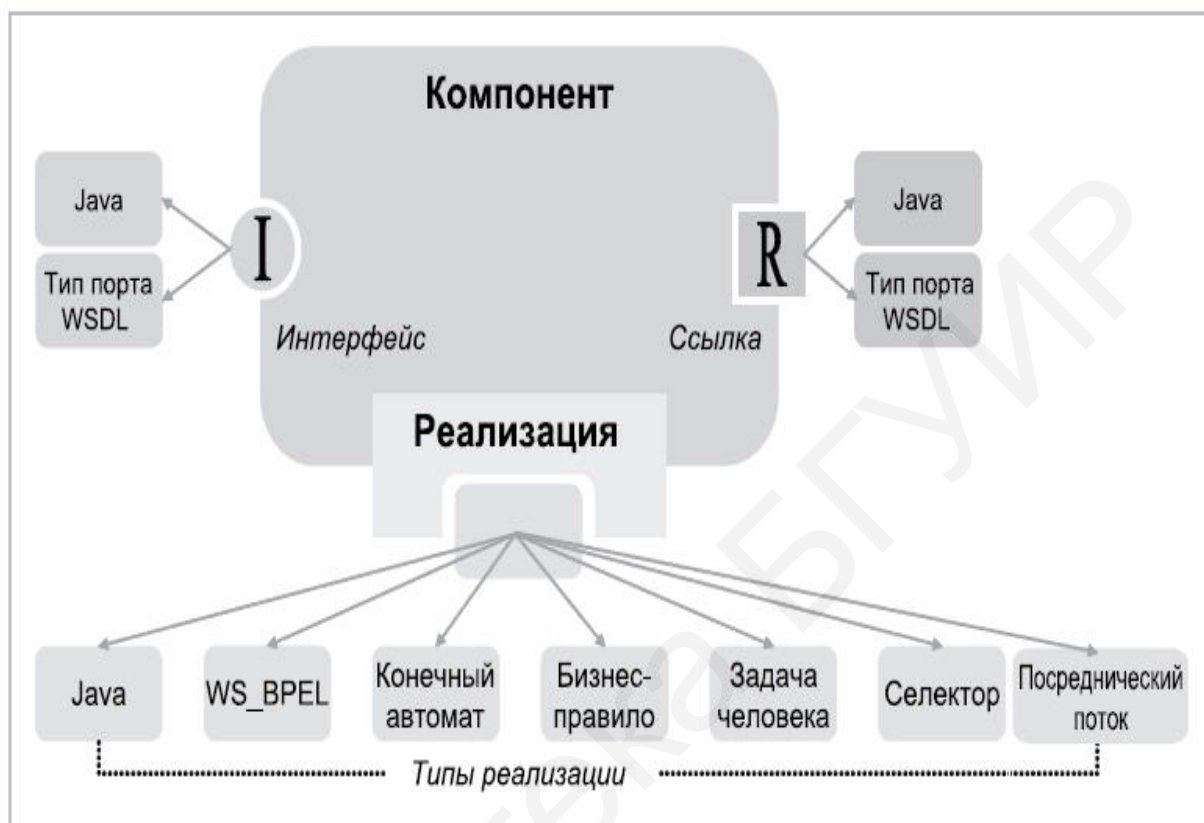


Рис. 15. Общая схема SCA-компонента

На диаграмме сборки *определяется тип реализации SCA-компонента:*

- Java;
- WS-BPEL;
- Конечный автомат;
- Селектор;
- Бизнес-правило;
- Задача человека;
- Посреднический поток.

На рис. 16 можно видеть типы реализации SCA-компонентов:

- GenerateMonthlyPlanGU12 – в виде WS-BPEL;
- AssignGu12Rows – Java;
- AutoAppRG – Бизнес-правило.

На диаграмме сборки интерфейсы SCA-компонента определяются Java-интерфейсом или типом порта WSDL.

Для SCA-компонента `GenerateMonthlyPlanGU12` определен WSDL-интерфейс, для которого на XML описаны аргументы, исключения и операции. Ниже приведен рис. 16 графического отображения *SDO GU12*.

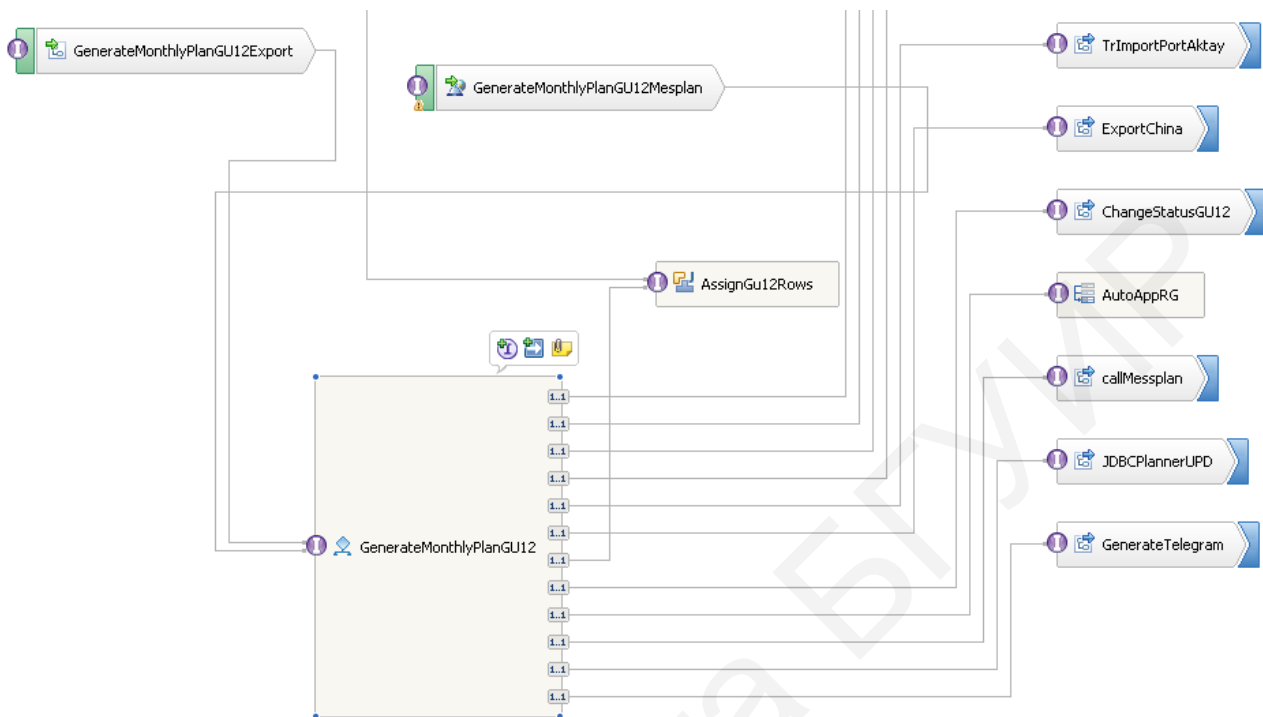


Рис. 16. Реализация SCA-компонента. Определение интерфейса

Для интерфейса `GenerateMonthlyPlanGU12` определены бизнес-объекты SDO: `gu12RequestsCreatedResponse`, `gu12RequestsCreated`, `sendPlanRequests`, `GU12` и т. д.

На рис. 17 показан интерфейс приложения IBM WebSphere Business Modeler. Он позволяет описывать WSDL-интерфейс, используя для этого графический пользовательский интерфейс. Такой подход удобен для пользователя, так как не требует знаний основ программирования для внесения изменений в модель. При этом он поддерживает все возможности IBM WebSphere Business Modeler.

После того как пользователь закончит описание модели в графическом пользовательском интерфейсе, IBM WebSphere Business Modeler создаст XML-код, который представляет собой данное описание на языке XML.

▼Operations

Operations and their parameters

	Name	Type
▼ transitImportRequestsCreated		
Inputs	trImpCreateReq	trImpReqCreated
▼ exportRequestsChanged		
Inputs	exportReqChanged	sendResponseToBP
▼ sendPlanSumInfo		
Inputs		
▼ sendPlanRequests		
Inputs		
▼ transitImportRequestsChanged		
Inputs	trImpReqChanged	sendResponseToBP
▼ sendExportPlanRequests		
Inputs		
▼ gu12RequestsCreated		
Inputs	gu12Un	long
Outputs	response	Message
Fault	faultResponse	Message

Рис. 17. Описание WSDL-интерфейса для SCA-компонента

В приложении представлено описание WSDL-интерфейса для SCA-компонента GenerateMonthlyPlanGU12 в виде XML.

3.3. Объекты-данные в модели SDO

В программной модели SDO объекты-данные представлены путем определения *Java-интерфейса* `commonj.sdo.DataObject`. Этот интерфейс включает в себя определение методов, которые позволяют клиентам читать и устанавливать свойства, связанные с `DataObject`.

Например, рассмотрим моделирование данных о заявке на перевозку ГУ-1х из компонента «Формирование портфеля заявок» при помощи объекта данных SDO.

Далее приведен пример использования *API DataObject* для получения свойств объекта-данных ГУ-1х.

```
private com.ibm.WebSphere.bo.BOFactory boFactory = null;  
private com.ibm.WebSphere.sca.ServiceManager serviceManager = null;  
serviceManager = new com.ibm.WebSphere.sca.ServiceManager();
```

```

boFactory = (com.ibm.WebSphere.bo.BOFactory) serviceManager
                .locateService("com/ibm/WebSphere/bo/BOFactory");
DataObject gU12 = boFactory.create("http://DkrKtgENG/Businessitems",
"GU12");
gU12.setInt("SPEC_COND", gu12DTO.getSpecCond());
gU12.setString("PLANNED_LOADDATE",
gu12DTO.getPlannedLoadDate());
gU12.setLong("REQ_STAUS_INT", gu12DTO.getReqStaGu12Un());
gU12.setLong("REQ_TYPE_TRAF_UN",gu12DTO.getReqTypeTraf());
gU12.setString("TRAFFIC_BY_ROUTES",gu12DTO.getTrafByRoutes());
gU12.setString("CAR_OWNER", gu12DTO.getCarOwner());
gU12.setString("REQ_STATUS", gu12DTO.getReqStatus());
gU12.setString("DEPARTMENT_NR", gu12DTO.getDepartmentNR());
gU12.setString("REQ_TYPE", gu12DTO.getReqType());
gU12.setString("VID", gu12DTO.getVid());
gU12.setBigDecimal("SUM_TONN", gu12DTO.getTonneSum());
gU12.setBigDecimal("REQ_MONTH", gu12DTO.getReqMonth());
gU12.setBigDecimal("REQ_YEAR", gu12DTO.getReqYear());
gU12.setBigDecimal("SUM_VAGON", gu12DTO.getVagonSum());
if(gu12DTO.getStartStation()!=null) {
gU12.setString("NOD", gu12DTO.getStartStation().getNOD()); }
gU12.setString("CARGO_OPER", gu12DTO.getCargoOper());
gU12.setString("BRANCH_LINE", gu12DTO.getBranchLine());
gU12.setString("BRANCH_LINE_OWNER",
gu12DTO.getBranchLineOwner());
gU12.setString("CARRIER_NAME", gu12DTO.getCarrierName());
gU12.setString("PLAN_NR", gu12DTO.getPlanNR());
gU12.setString("REG_DATE", gu12DTO.getRegDate());
gU12.setString("USER_NAME", gu12DTO.getUserName());

```

3.4. Граф данных в модели SDO

Граф данных – это структура, инкапсулирующая набор объектов-данных. От объекта-данных, находящегося в графе на верхнем уровне, можно обращаться к другим объектам-данным, переходя по ссылкам из корневого объекта-данных (рис. 18, 19).

В программной модели SDO графы данных представлены путем определения *Java-интерфейса commonj.sdo.DataGraph*.

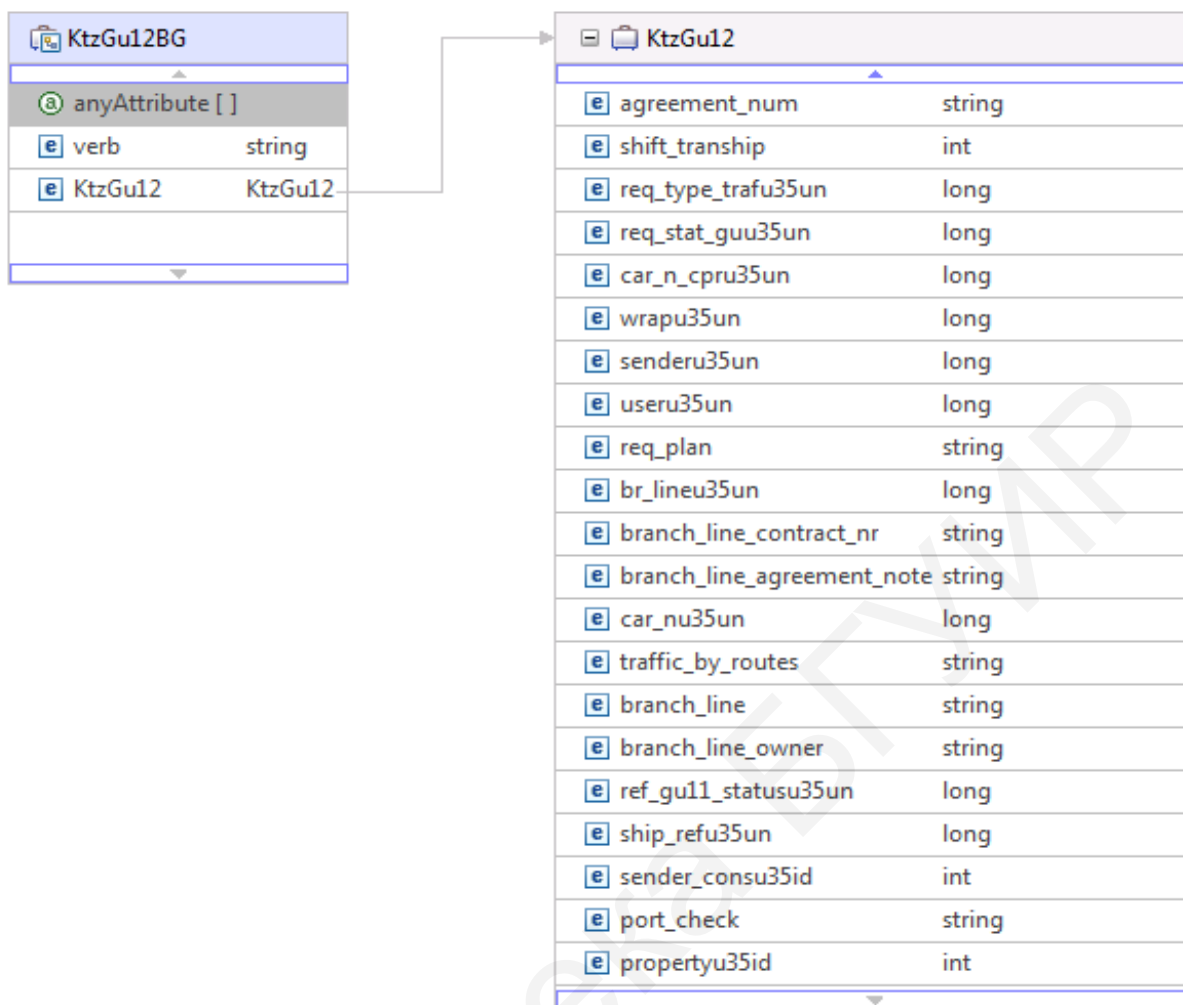


Рис. 18. Бизнес-граф «ГУ-1х заявка»

На диаграмме сборки определяется *импорт компонента*, который позволяет использовать функции, не являющиеся частью собираемого модуля.

Импорт (компонент импорта) имеет интерфейсы, аналогичные интерфейсам удаленной службы (или их части), с которой он связан и через которые можно вызывать удаленную службу.

Импорт используется в приложении точно так же, как локальные компоненты. Этим создается единая модель для всех функций независимо от их местоположения и реализации.

С помощью диаграммы сборки *публикуется интерфейс SCA-компонентов путем создания его экспорта.*

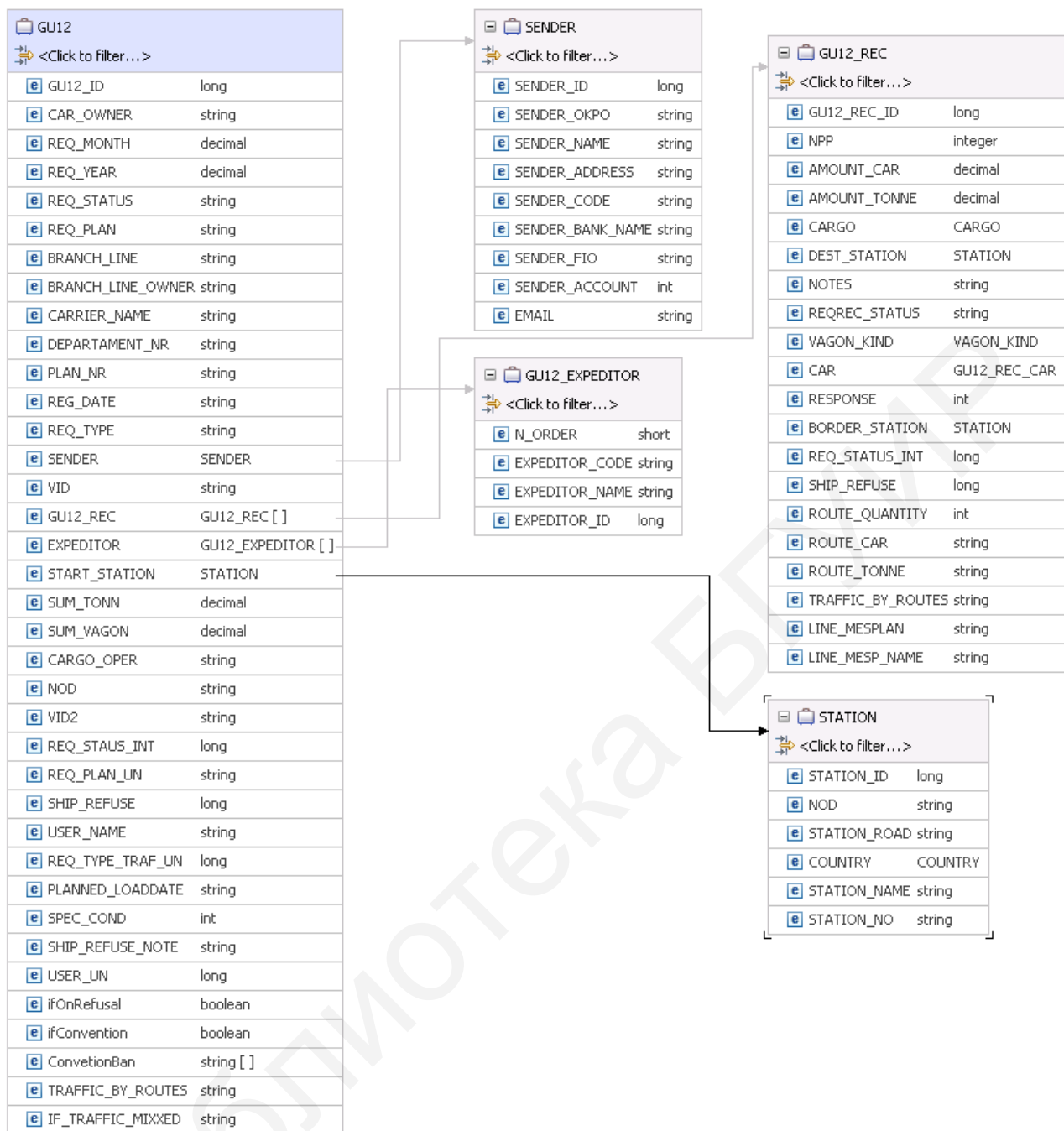


Рис. 19. Графическое отображение SDO GU12 «стока заявки»

На рис. 19 показано, как импортированы SCA-компоненты:

- InnerMahambetApp;
- ExportSNG;
- TrImportMesplan;
- ExportAktau;
- ChangeStatusGU12;
- callMessplan.

На рис. 19 также можно увидеть, как экспортирован SCA-компонент *GenerateMonthlyPlanGUI2*.

Диаграмма сборки позволяет динамически заменять сервисные компоненты и компоненты-помощники на новые, при этом прослеживается гибкость, так как отсутствует жесткая связанность модулей.

Можно интегрировать показанный модуль в другое приложение путем генерации экспорта всего бизнес-процесса и импорта на другую диаграмму сборки в качестве сервисного компонента.

Фрагмент диаграммы сборки приведен на рис. 20.



Рис. 20. BPEL процесса «Формирование месячного плана»

3.5. Реализация бизнес-процесса на языке BPEL

Язык реализации BPEL является языком описания бизнес-процессов. Процесс, реализованный в виде последовательности действий или конечного автомата, при развертывании на Process Server описываются как BPEL-процесс.

В бизнес-процессе описывается последовательность выполняемых действий, которая включает:

- вызов других сервисов;
- компоненты логирования;
- простые Java-вставки (при компиляции преобразуются в Java-класс);
- назначения;
- таймеры, описывающие логику задержки либо установки на определенное время;
- карты преобразования объектов;
- различные виды циклов;
- обработчики ошибок;
- описание задач, выполняемых человеком.

На рис. 21 показан процесс согласования экспортных перевозок плановиком центрального аппарата.

Данный процесс реализован как длительный, так как он содержит задачу, выполняемую человеком. Можно выбрать и альтернативный вариант: микропоток. Данная реализация используется при описании процесса, который не содержит вызов сервисов, таймеров либо задач человека.

Процесс «Согласование экспортных перевозок плановиком» содержит:

- вызов сервиса «Изменение статуса заявки»;
- карту преобразования бизнес-объектов;
- таймер;
- различные циклы;
- обработчики ошибок разного уровня.

Вызов сервиса (рис. 22) осуществляется путем внедрения в процесс партнерских ссылок, которые описывают WSDL внешнего сервиса. Также в качестве сервиса может выступать задача человека (рис. 23–25). Она в таком случае будет называться *Standalone-задача*. То есть данная задача будет храниться в контейнере задач, выполняемых человеком, как описание шаблона. При вызове такой задачи будет генерироваться объект шаблона.

Каждая задача, которая должна выполняться человеком, на языке BPEL описывается с помощью либо SCA-компонента, либо линейной задачи человека. Такая задача является компонентом при построении практически любого бизнес-процесса.

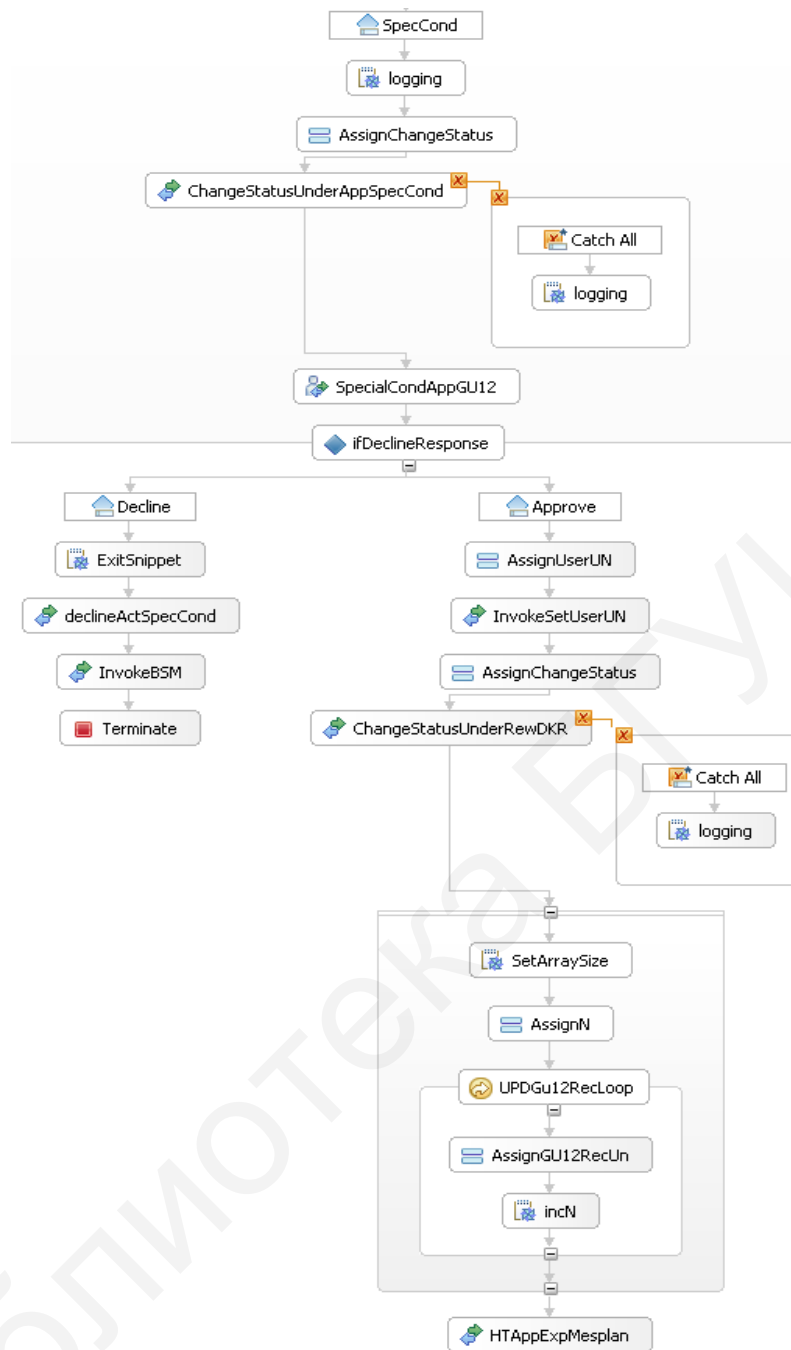


Рис. 21. BPEL «Согласование экспортных перевозок плаником»

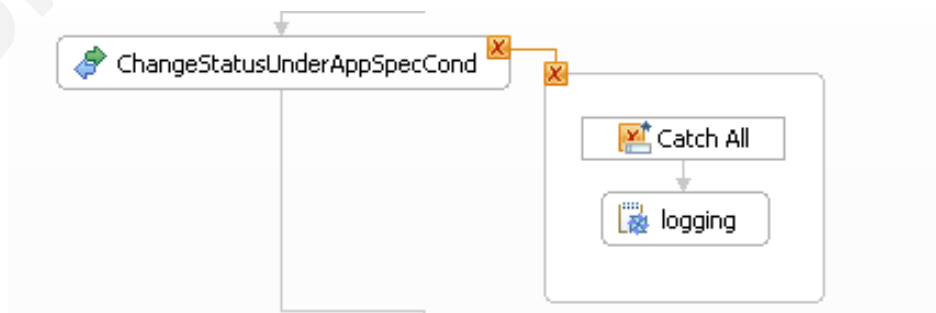


Рис. 22. Вызов внешнего сервиса

Рис. 23. Вызов задачи человека в BPEL

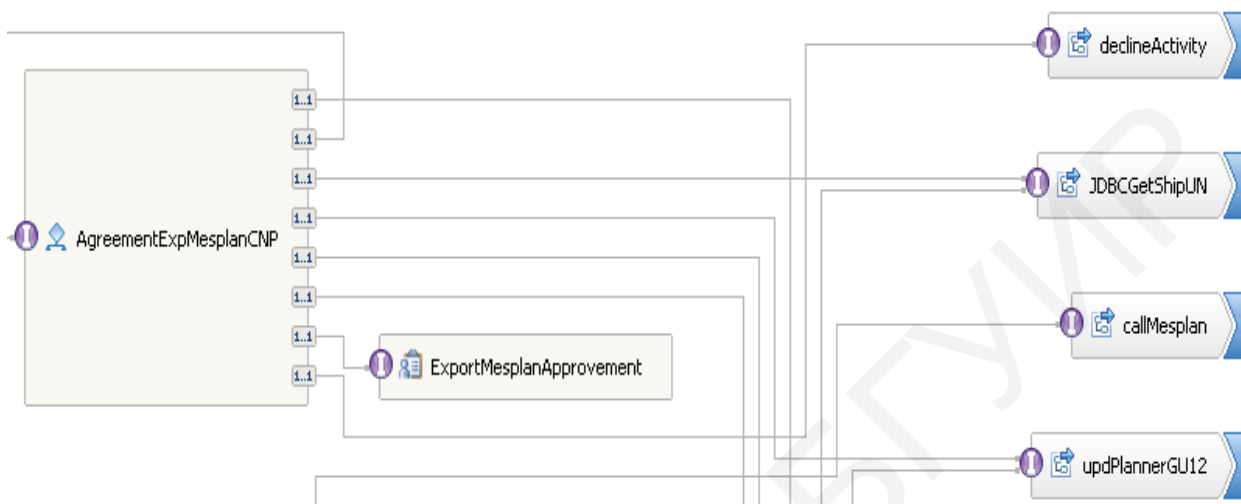


Рис. 24. Вызов задачи человека. Диаграмма сборки

Как и любой SCA-компонент, задача имеет WSDL-интерфейс, описывающий SDO, с которым она будет работать.

Также задача может содержать описание групп пользователей, которые смогут с ней работать.

Данные группы пользователей могут иметь привилегии администратора, владельца, инициатора, чтеца или корректора, а также быть определены как группа, специализированный пользователь, группа без пользователя. Возможно определить запрос выборки пользователей. Аккаунты пользователей могут храниться в LDAP, VMM, User Registry.

Таким образом, специфика реализации задачи человека позволяет силами Process Server назначить выполнение задачи определенным пользователям.

При описании задачи человека можно указать метод реализации:

- с помощью Lotus-форм (рис. 26);
- с помощью порталных средств в Busness Choreographer (рис. 27).

▼ **To-do Task**

Name	ExportMesplanApproval
------	-----------------------

► **Service Interface**

▼ **People Assignment (Receiver)** + ×

Administrators	Everybody								
Potential Owners	Group Members								
	<table border="1"> <tr> <td>GroupName *</td> <td>CNPPlanners</td> </tr> <tr> <td>IncludeSubgroups *</td> <td>✓</td> </tr> <tr> <td>AlternativeGroupName1</td> <td>NODPlanners</td> </tr> <tr> <td>AlternativeGroupName2</td> <td>wpsadmins</td> </tr> </table>	GroupName *	CNPPlanners	IncludeSubgroups *	✓	AlternativeGroupName1	NODPlanners	AlternativeGroupName2	wpsadmins
GroupName *	CNPPlanners								
IncludeSubgroups *	✓								
AlternativeGroupName1	NODPlanners								
AlternativeGroupName2	wpsadmins								

▼ **User Interface** + ×

User Interface
Portlets for Portal Server

▼ **Escalation**

Ready
|
sendEmail

Claimed Subtask started

Рис. 25. Реализация задачи ExportMesplanApproval, выполняемой человеком

Заявка ГУ-12 № 101377 Статус заявки: На согласовании АСУ ДКР

Общие сведения Грузоотправитель перевозка Строки ГУ-12 ГУ-114

Признак перевозки *
3 - экспорт в третьи страны СНГ, Латвию, ▼

Месяц * Год *
Март 2011

Дата подачи заявки Предполагаемая дата погрузки *
2011-02-15 [calendar icon]

Тип заявки
внеплановая

Перевозка маршрутами

Комментарии и дополнения

March, 2011						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

NUM

Рис. 26. Реализация задачи человека с помощью Lotus-форм



Общие сведения	Грузоотправитель	Перевозка	Строки ГУ-12	ГУ-114								
Код номенклатурной группы	Номенклатурная группа *	Принадлежность вагона *										
34	ЗЕРНО	ИНВЕНТАРНЫЙ ПАРК										
№	Наименование груза	Особые условия	Страна назначения	Отделение назначения	Станция назначения	Грузополучатель	Род/Тип Вагонов	Кол-во вагонов	Кол-во тонн	Статус строки	Причина несогласования	Маршрутная отправка
1	011005 - Пшеница	<input type="checkbox"/>	398 - КАЗАХСТАН	03 - ПАВЛОДАРСКОЕ	694709 - Екибастуз II (бывш. Экибастуз II)	1746 - ТОО "БОГАТЫРЬ КОМИР"	20/0200 - КРЫТЫЕ	60	2000	На оформлении		Нет
2	011005 - Пшеница	<input type="checkbox"/>	398 - КАЗАХСТАН	03 - ПАВЛОДАРСКОЕ	694709 - Екибастуз II (бывш. Экибастуз II)	1746 - ТОО "БОГАТЫРЬ КОМИР"	20/0200 - КРЫТЫЕ	600	20000	На оформлении		Да

Количество найденных записей - 2.

ВСЕГО	В т.ч. количество вагонов с разбивкой по родам	
ТОНН	ВАГОНОВ	КР
22000	660	660

[Назад](#)

Рис. 27. Реализация задачи человека с помощью порталных средств

Задачи человека разделяются по принципу:

- «Сделать»;
- «Передать управление»;
- «Инициализация бизнес-процесса».

Для приложения «Дорога» были созданы задачи, реализованные с помощью порталных средств, которые инициализируют процессы, а также задачи согласования задач. Задачи согласования попадают в лист задач, который специально реализовывался для приложения «Дорога», происходит интеграция портала и Process Server. Портлет «лист заданий» работает с бизнес-объектом, таким образом система повышает быстродействие, так как нет необходимости в работе с физической базой данных (рис. 28).



НОДЫ

- Все
- АКМОЛИНСКОЕ
- АКТЮБИНСКОЕ
- АЛМАТИНСКОЕ
- АТЫРАУСКОЕ
- ЖАМБЫЛСКОЕ
- ЗАЩИТИНСКОЕ
- К-ОРДИНСКОЕ
- КАРАГАНДИНСКОЕ
- КУСТАНАЙСКОЕ

Станции

- Все
- Абаил
- Абыла
- Аврора
- Адыр
- Азат
- Аиртау
- Айгыржал
- Айке
- Айна-Булак

Номен. группы

- Все
- АВТОМОБИЛИ
- БУМАГА
- ВСЕГО
- ГРАНУЛИРОВАННЫЕ ШЛАКИ
- ГРУЗЫ В КОНТЕЙНЕРАХ
- ЖИВНОСТЬ
- ЖМЫХИ
- ЗЕРНО
- ИМПОРТНЫЕ ГРУЗЫ

Найти

Вид задачи	Номер плана	Год	Месяц	Грузоотправитель	Номенклатура	Принадлежность вагона
<input type="checkbox"/> Согласовать ГУ-12	102162	2011	11	"ЭЛЬ ПАНДА" ЖАУАПКЕРШІЛІГ ШЕКТЕУЛІ СЕРІКТЕСТІГІ	ПРОДУКТЫ ПЕРЕМОЛА	инвентарный парк
<input type="checkbox"/> Согласовать ГУ-12	102163	2011	11	ТОО "МАҒЖАН И К"	ПРОДУКТЫ ПЕРЕМОЛА	инвентарный парк
<input type="checkbox"/> Согласовать ГУ-12	102166	2011	11	ТОО "МАҒЖАН И К"	ПРОДУКТЫ ПЕРЕМОЛА	инвентарный парк
<input type="checkbox"/> Согласовать ГУ-12	009044	2011	12	ТОО "МЕТАЛЛТЕРМИНАЛСЕРВИС"	РУДА ЖЕЛЕЗНАЯ И МАРГАНЦЕВАЯ	инвентарный парк
<input type="checkbox"/> Согласовать ГУ-12	009047	2011	12	ТОО "АРМАН"	РУДА ЖЕЛЕЗНАЯ И МАРГАНЦЕВАЯ	инвентарный парк
<input type="checkbox"/> Согласовать ГУ-12	009050	2011	12	ТОО "VATAN"	АВТОМОБИЛИ	инвентарный парк
<input type="checkbox"/> Согласовать ГУ-12	009051	2011	12	ТОО "ЗЕРНОВАЯ КОМПАНИЯ "СУНКАР И К"	ПРОДУКТЫ ПЕРЕМОЛА	инвентарный парк
<input type="checkbox"/> Согласовать ГУ-12	009054	2011	12	ТОО "ФИРМА ДАНА"	ЛОМ ЧЕРНЫХ МЕТАЛЛОВ	инвентарный парк
<input type="checkbox"/> Согласовать ГУ-12	009064	2011	12	ТОО "КАРТАШ"	ПРОДУКТЫ ПЕРЕМОЛА	инвентарный парк
<input type="checkbox"/> Согласовать ГУ-12	009065	2011	12	ТОО "РАУАН 2010"	ПРОДУКТЫ ПЕРЕМОЛА	инвентарный парк

Рис. 28. Лист согласования ГУ-хх

Удобным способом управления задачей, выполняемой человеком, является эскалация – механизм, позволяющий выполнять различные действия в разных статусах задачи. Например, для задачи ExportMesplanApproval определена отправка E-mail нотификации при создании экземпляра задачи. Также можно назначить перенаправление задачи другим пользователям в случае, если она не была затребована либо находится в статусе «Готова» определенное количество времени (описание календаря может быть либо на Java, либо на Modeler).

С помощью задач человека можно построить процесс, «подстроенный под человека». Можно указать любое количество задач человека и ролью «администратор» упразднить определенное количество задач либо назначить на выполнение другие группы пользователей.

При реализации бизнес-процесса «Согласование экспортных перевозок» использовался таймер, который определял количество времени на выполнение задачи (данный механизм можно также реализовать с помощью эксальции). По достижении временного интервала 3 дня до начала месяца либо 3 дня до дня отправки при несогласовании заявки бизнес-процесс автоматически отклонит заявку. Таймер может быть задан как длительность или определенный день с помощью календаря (рис. 29).

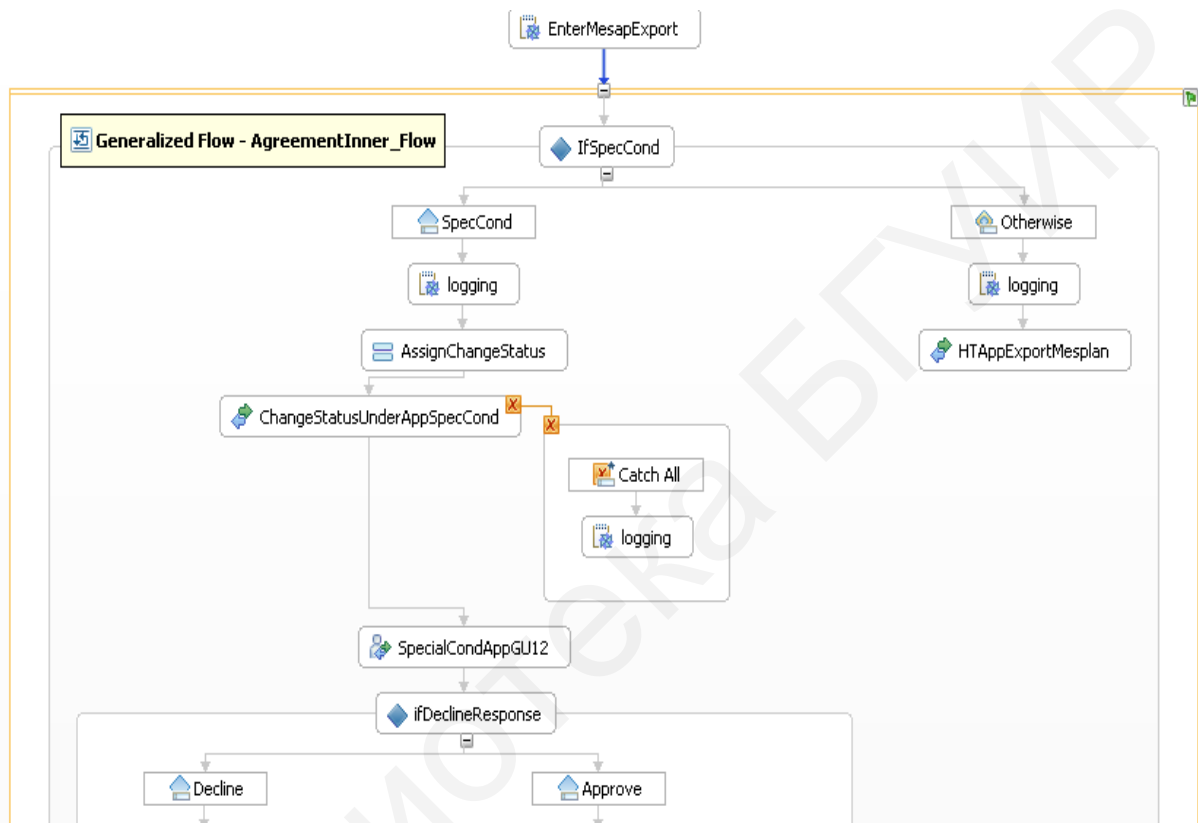


Рис. 29. Механизм автоматического отклонения заявки

Реализация бизнес-процесса на языке BPEL позволяет отслеживать возможность возникновения ошибки в процессе выполнения задачи, ее компенсации и отладки. Отметим, что отладка может происходить на любом уровне, при этом процесс не будет прерван. Ошибка может быть обработана как на уровне выполнения компонента, так и на уровне общего бизнес-процесса. Мощнейший механизм компенсации позволяет бизнес-процессу вернуться к срезу до возникновения ошибки, то есть при отключении электричества (технические неполадки) либо ошибке реализации сервиса (ошибка функционала) бизнес-процесс способен совершить откат до последнего верно-логического выполнения (рис. 30).

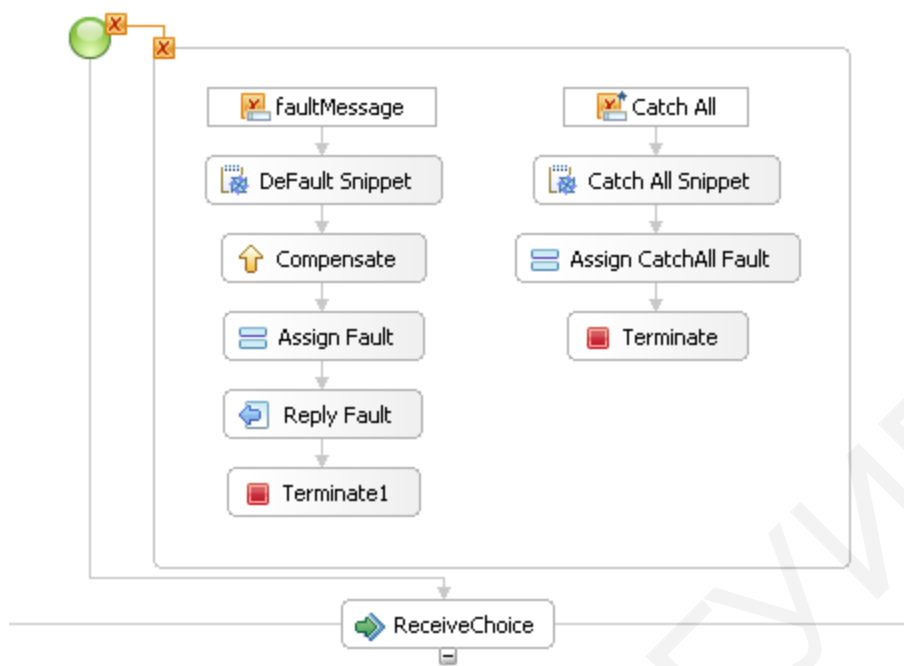


Рис. 30. Обработчик ошибок корневого уровня процесса «Согласование экспортных перевозок»

Проблема вызова внешних сервисов с WSDL-интерфейсом и SDO, отличными от используемых в бизнес-процессе, легко решается с помощью карты преобразования интерфейсов и карты преобразования бизнес-объектов (рис. 31). При реализации бизнес-процесса «Создание подъездного пути» компонента «Ведение договоров» для объекта «Дорога» активно использовался данный механизм.

Карты позволяют либо переопределить объект входящего поля в объект исходящего, либо вызвать внешний сервис, либо использовать дополнительную карту, взаимосвязи, объединение, подчинение определенному паттерну и т. д. Данный механизм позволяет минимизировать количество интерфейсов и повторно использовать внешние сервисы.

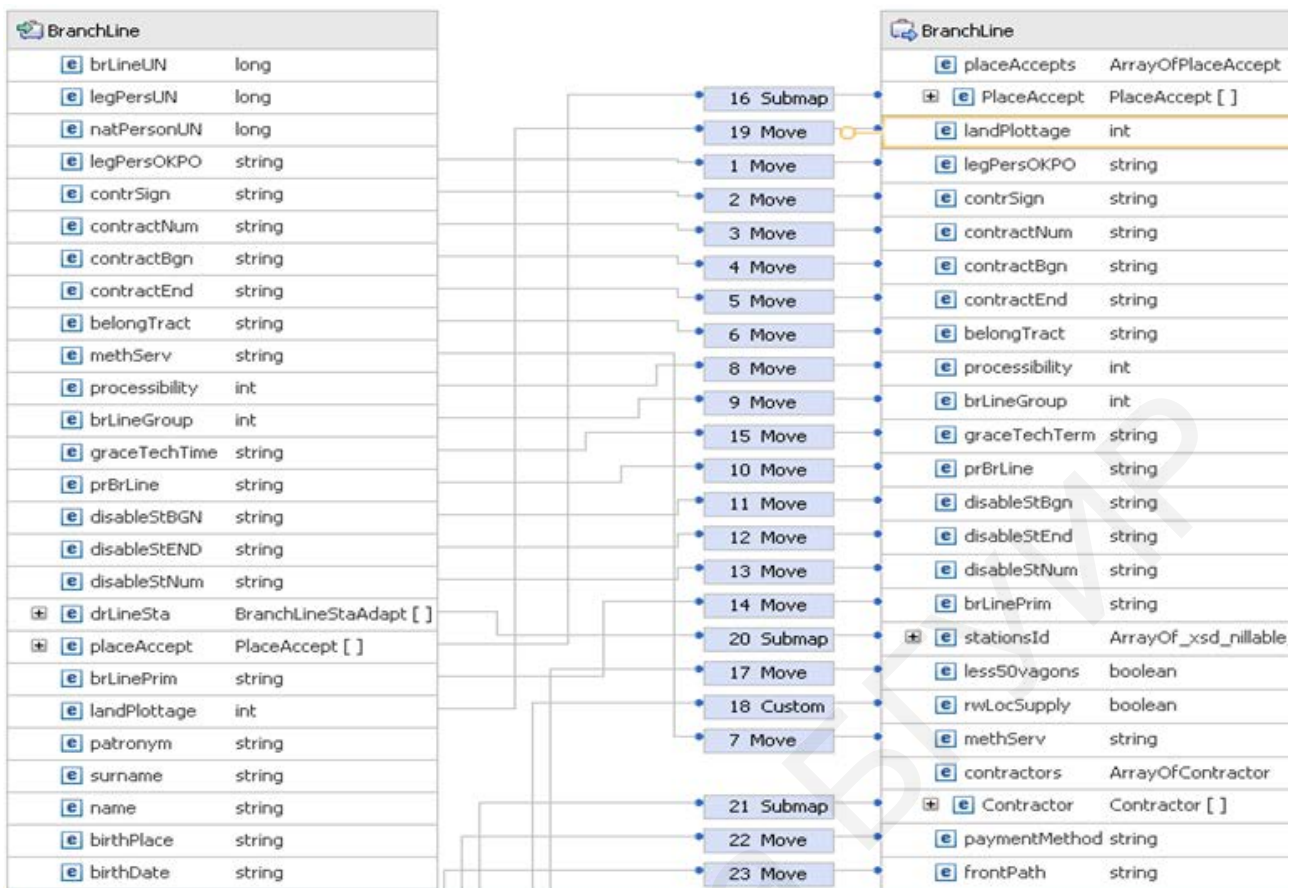


Рис. 31. Карта преобразования бизнес-объектов

Дополнительной гибкости выполнения бизнес-процессов можно достичь с помощью использования бизнес-правил, которые можно изменять и создавать во время выполнения экземпляра процесса. Для ХХХ в процессе «Формирование месячного плана» используется шаблон правила «Отмена проверки запретов». Данная функциональность позволяет администратору задавать параметры проверяемых запретов в процессе выполнения.

Итак, в данном подразделе было рассмотрено выполнение бизнес-процесса как последовательности действий, а также компоненты, которые содержит такая реализация, и различные механизмы управления процессом.

3.6. Реализация бизнес-процессов в виде конечного автомата на языке VPEL

Для приложения «Дорога» компонента «Формирование месячного портфеля заявок» были разработаны следующие бизнес-процессы в виде конечного автомата:

- «Согласование заявок ГУ – 1х межобластные перевозки»;
- «Согласование заявок ГУ – 1х экспорт Китай»;

- «Согласование заявок ГУ – 1х экспорт через порт Актау»;
- «Согласование заявок ГУ – 1х международные перевозки»;
- «Согласование заявок ГУ – 1х на транзит, импорт через порт Актау»;
- «Согласование заявок ГУ – 1х на транзит, импорт из АС Месплан»;
- «Согласование заявок ГУ – через станцию Махамбет».

Названные бизнес-процессы были реализованы в виде машины состояний, так как данный метод самый быстрый для решения поставленных задач, а также архитектура, построенная на конечном автомате, легко модифицируема.

Конечные автоматы являются моделью и способом динамического описания поведения процесса путем фокусирования на событии, которое происходит при переходе от состояния в состояние. Концепция бизнес-машины состояния основывается на UML, реализация машины состояния описывается языком SACL (State Adaptive Choreography Language), который при развертывании на Process Server генерирует BPEL.

Бизнес-машина состояний – событийный процесс, который изменяет свое дискретное состояние по причине отработки внешней операции. Каждое состояние определяет, какие действия и операции могут быть вызваны. Конечный автомат графически отображает последовательность статусов, сущности, транзакции и результат перехода из статуса в статус.

Средства WebSphere Integration Developer позволяют легко создавать и манипулировать бизнес-машиной состояний.

Машина состояний является мощнейшим механизмом управления процессом в связи с наличием следующих признаков:

- является событийной моделью;
- в процессе моделирования есть возможность создавать машины состояний в зависимости от сценария прецедента;
- во время анализа и проектирования есть возможность смоделировать событийные объекты, которые реагируют на события вне контекста объекта;
- во время анализа и проектирования есть возможность использовать несколько диаграмм машин состояний, чтобы показать различные аспекты машины состояний и ее поведение.

Сравнительный анализ машины состояний и BPEL приведен в табл. 3.

Сравнительный анализ машины состояний и BPEL

Модуль	WS-BPEL	Конечный автомат
Обработчик событий	Некоторые возможности обработки событий достигаются стандартными обработчиками	Большинство транзакций являются событийными
Последовательное выполнение	Большинство задач реализуются как последовательность	Автоматическая транзакция обеспечивает последовательное выполнение
Цикл	Процесс не может быть зациклен, поддерживаются циклы для задач в процессе	Статусы могут выполняться в любом порядке
Статус процесса	Статус процесса может влиять на транзакцию в параллельном потоке исполнения	Реакция на событие зависит от статуса процесса
Обработчик	Поддерживается обработка событий и компенсаторы	Не поддерживает обработку ошибок и компенсаторы

Далее будут рассмотрены разработанные бизнес-машины состояний и их взаимодействие с BPEL.

3.6.1. Описание машины состояний

На рис. 32 представлена бизнес-машина, реализующая логику согласования заявок «ГУ-хх международные перевозки».

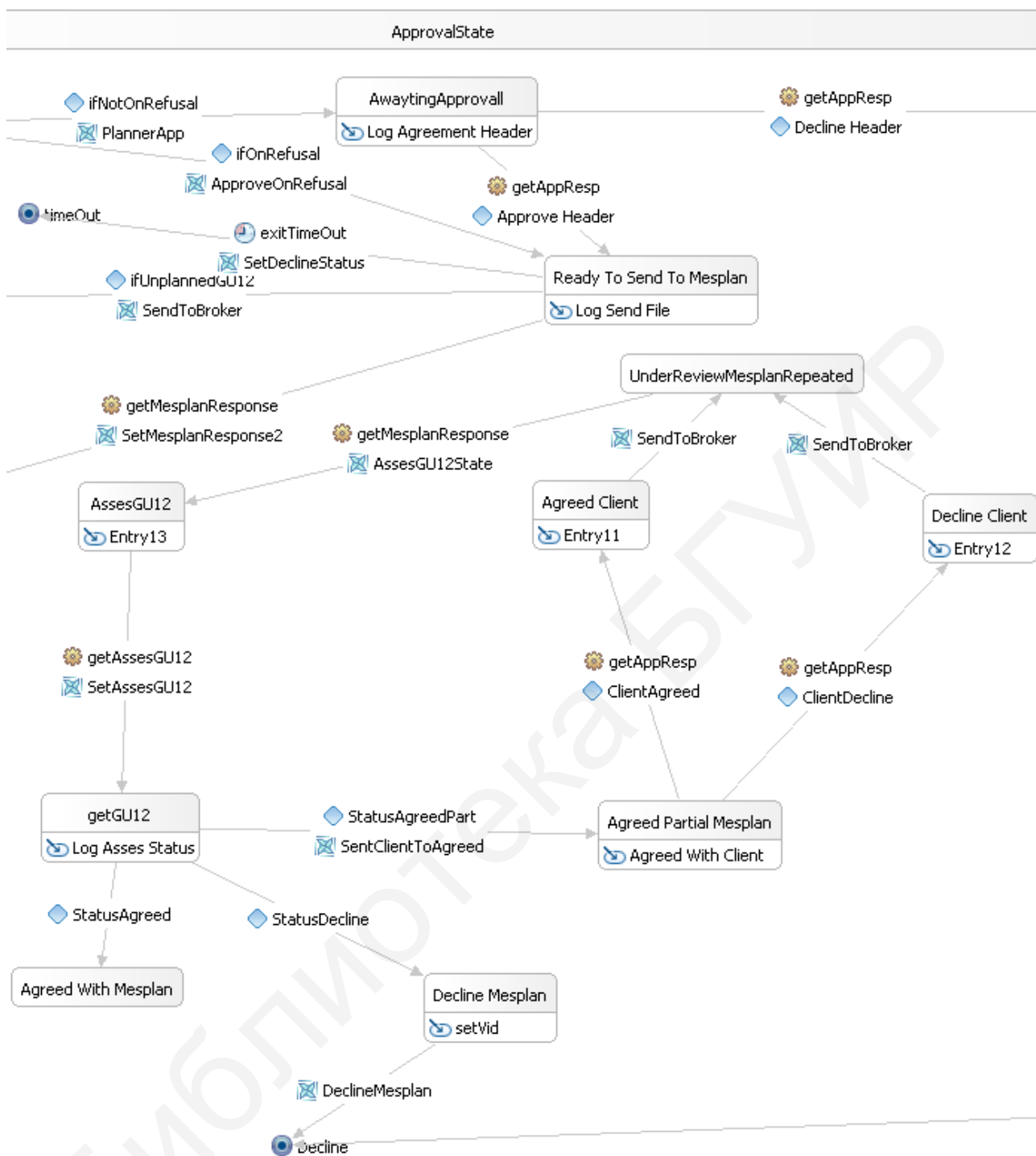
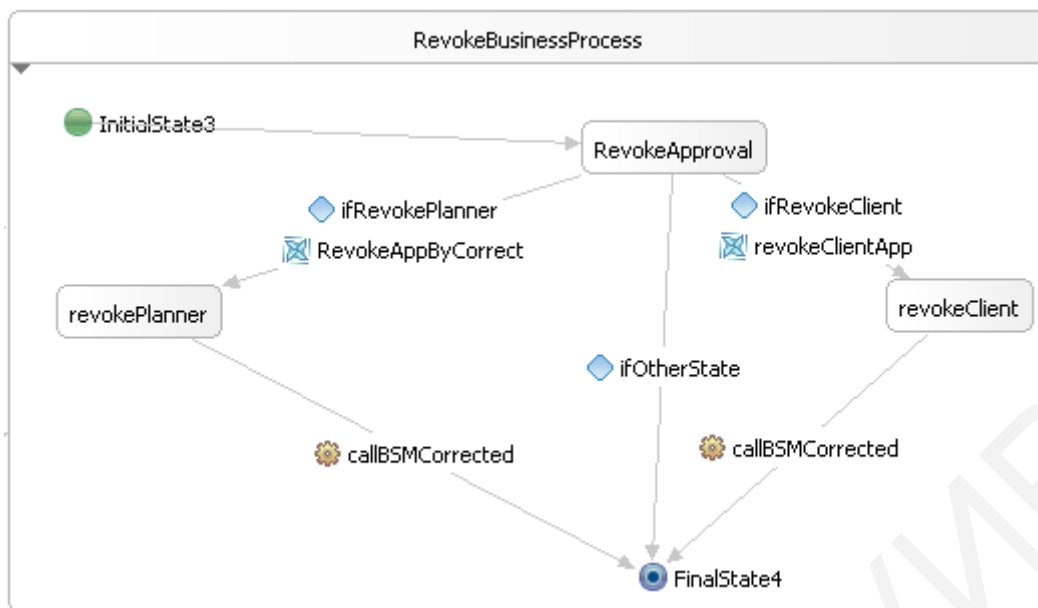
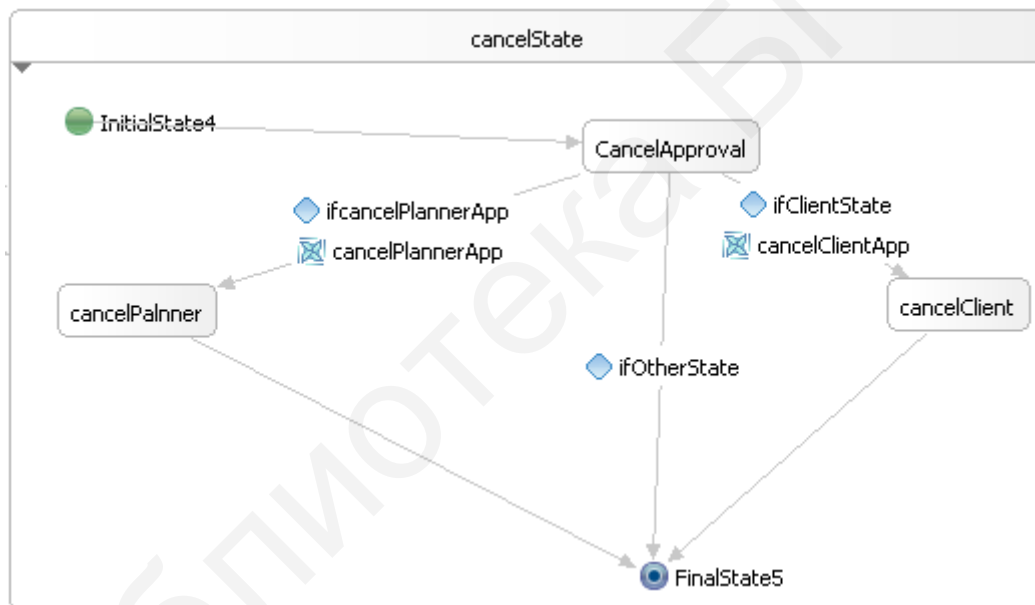


Рис. 32. Машина состояния для процесса «Согласование заявок ГУ-хх международные перевозки»

В машине состояний имеют место *статусы, операции, действия, решения и транзакции* (рис. 33).



a



б

Рис. 33. Составляющие элементы машины состояний

Статус может быть простым (AwaitingApproval) или композитным (ApprovalState).

Композиты используются для структурирования логики исполнения: для всех статусов в статусе-композите имеют место события correctGU12, cancelGU12.

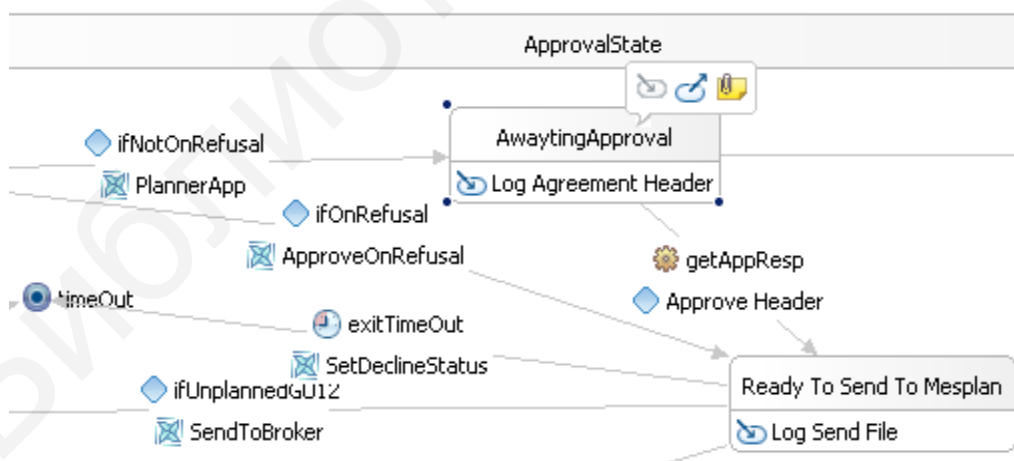
Непосредственно *операция* отвечает за событийность выполнения процесса. Операция может быть вызвана из внешнего сервиса, другой машины

состояний, WS-BPEL, любой внешней системы как один из методов WSDL для SCA. Если событие имело место, вызывается условие перехода. Оно может быть задано как X-Path, Java, вызов внешнего сервиса. Если условие принимает значение «Истина», вызывается действие, которое также может быть реализовано как X-Path, Java, вызов внешнего сервиса.

Для каждой *транзакции* можно установить таймер, который определяет длительность пребывания в статусе либо конкретный день. Таймер может быть задан календарем либо кодом на Java. Также для каждого конкретного статуса можно указать активность при входе и выходе.

На рис. 34 описан статус AwaitingApproval:

- getAppResp – операция: событие перехода;
- ifNotOnRefusal – решение: реализует логику проверки создания ГУ-хх на основании отказной;
- PlannerApp – активность: в данной реализации вызывается WS-BPEL «Согласование ГУ-хх международные перевозки плановиком ЦНП»;
- SendToBroker – активность: реализует взаимодействие с MessageBroker как внешним сервисом;
- exitTimeOut – таймер: с помощью Java высчитывается конкретная дата для конкретного объекта класса машины состояния;
- Log Agreement Header – активность, выполняющаяся при переходе в статус AwaitingApproval.



6

Рис. 34. Составляющие элементы машины состояний

Таким образом, каждый статус может реализовывать вызов сложной последовательности действий. Например, BPEL, представленный на рис. 35,

позволяет построить гибкую архитектуру, изменяющуюся по требованию и легко модифицируемую.

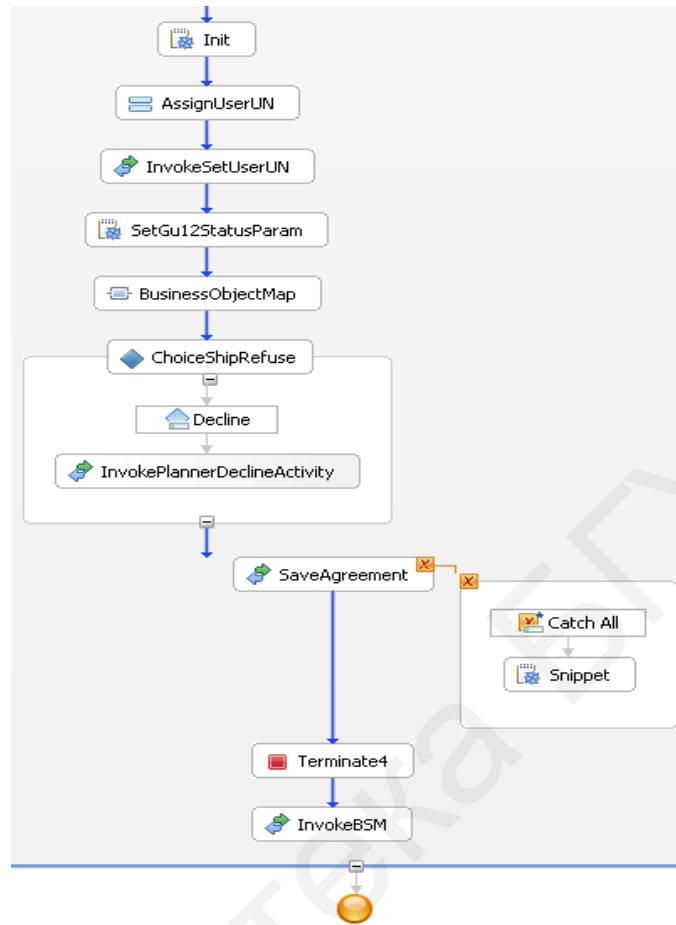


Рис. 35. Статусы и вызов действий в BPEL

Машина состояний переходит в статус *AwaytingApproval*. В данном состоянии машина пробудет до тех пор, пока она не будет вызвана извне при помощи операции *getAppResp*. В таком случае машина состояний будет вызываться из бизнес-процесса, реализованного в виде последовательности выполняемых заданий *AgreementExpMesplanCNP*. Решение, по какой ветке операции *getAppResp* пойдет бизнес-процесс, принимается в результате ответа согласования, принятого человеком.

Вопрос корреляции решается путем введения коррелирующих данных, в данном случае ими является идентификационный номер клиента. Корреляционные данные должны быть определены как в машине состояний, так и в бизнес-процессе, реализованном как последовательность выполняемых заданий.

Как только бизнес-машина вновь запущена, она переходит в следующее состояние. В данном случае состояние определяется в графе «решение», где можно задать любой метод принятия решения.

Таким образом, машина состояний переходит из статуса в статус с помощью механизма вызова других бизнес-процессов, которые могут быть реализованы либо как последовательность выполняемых заданий, либо как другая машина состояний. Наблюдается очень большая гибкость данной архитектуры.

Перед разработчиком стоит главная задача: составить правильные состояния переходов машины. После этого при помощи мощного механизма вызова других бизнес-процессов можно реализовать любую логику перехода. Таким образом, построив «костяк», разработчику не придется полностью менять код или архитектуру, можно будет лишь добавить новое состояние бизнес-машины либо новый вызов другого бизнес-процесса и на основе сгенерированного «скелета» Java-кода разработать сервисы, реализующие задания в бизнес-процессах.

3.6.2. Инициализация бизнес-процесса по расписанию. Пример взаимодействия SCA и EJB

Задача вызова BPEL по расписанию позволит изучить возможность взаимодействия SCA и EJB, а также задание расписания на исполняющем сервере Process Server. Далее приведен код вызова из EJBBean.

```
public void process(TaskStatus task) {  
    System.out.println("ServiceTaskBean started");  
  
    try {  
        Date date = new Date();  
        String in = "This is a process created at " + date;  
  
        // Create the EJB  
        LocalBusinessFlowManager flowManager = bpcHome.create();  
  
        // Query the process template  
        String whereClause = "PROCESS_TEMPLATE.NAME =  
'StartPlannedMesplan";  
        String orderByClause = "PROCESS_TEMPLATE.VALID_FROM  
DESC";
```



```

ProcessTemplateData[] templates = flowManager.queryProcessTemplates(
    whereClause, orderByClause, null, null);

if (templates == null || templates.length == 0) {
    System.out
        .println("Error, process template \"StartPlannedMesplan\" not found.
Ensure that the process is installed correctly and the process template is started!");
    } else {
        ProcessTemplateData processTemplate = templates[0];

        // Create the input ClientObjectWrapper
        ClientObjectWrapper cow =
flowManager.createMessage(processTemplate
        .getID(), processTemplate.getInputMessageType());

        // Get the Input DataObject from the ClientObjectWrapper
        DataObject input = (DataObject) cow.getObject();

        // Create the second/inner DataObject
        DataObject dataIn = input.createDataObject("input");

        // set the Data
        dataIn.setString("name", in);

        // Call the Process and get the output ClientObjectWrapper
        ClientObjectWrapper cowProcessOut = flowManager.call(
            "StartPlannedMesplan", cow);

        // get DataObject
        DataObject inner = (DataObject) cowProcessOut.getObject();

        // get output Data
        String resp = inner.getDataObject("output").getString("done");

        // print result
        System.out.println("Process returned with: " + resp);
    }
} catch (Exception e) {

```

```
e.printStackTrace();
} finally {
    System.out.println("ServiceTaskBean ended");
} }
```

Для непосредственного задания расписания используется конфигурационный файл профайла на процесс-сервере.

```
# 'frequency' specifies WHEN to run the repeating task
# See the InfoCenter documentation for the Scheduler for details
# about the format.
#
# Examples:
# every minute
# frequency=0 * * * * ?
# every hour
# frequency=0 0 * * * ?
# every saturday at 2 am
# frequency=00 00 02 ? * SAT
# every last day in the month at 3 am
# frequency=00 00 03 L * ?
#
frequency=0 * * * * ?
```

ПРИЛОЖЕНИЕ

Пример WSDL-интерфейса для SCA-компонента GenerateMonthlyPlanGU12

```
<?xml version="1.0" encoding="UTF-8"?>

<wsdl:definitions
    name="GenerateMonthlyPlanGU12Interface"
    targetNamespace="http://DkrKtgENG/Processes/ProcessNEW/GenerateMonthlyPlan
    GU12/GenerateMonthlyPlanGU12Interface"
    xmlns:bons1="http://ASUDKR_Library"
    xmlns:bons3="http://ASUDKR_Library/Mesplan"
    xmlns:tns="http://DkrKtgENG/Processes/ProcessNEW/GenerateMonthlyPlanGU12/
    GenerateMonthlyPlanGU12Interface"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">

    <wsdl:types>

        <xsd:schema
            targetNamespace="http://DkrKtgENG/Processes/ProcessNEW/GenerateMonthlyPlan
            GU12/GenerateMonthlyPlanGU12Interface">
            <xsd:import
                namespace="http://ASUDKR_Library/Mesplan"
                schemaLocation="Mesplan/trImpReqCreated.xsd"/>
            <xsd:import
                namespace="http://ASUDKR_Library/Mesplan"
                schemaLocation="Mesplan/sendResponseToBP.xsd"/>
            <xsd:import
                namespace="http://ASUDKR_Library"
                schemaLocation="Message.xsd"/>
            <xsd:element name="transitImportRequestsCreated">
            <xsd:complexType>
            <xsd:sequence>
                <xsd:element
                    name="trImpCreateReq"
                    nillable="true"
                    type="bons3:trImpReqCreated"/>
            </xsd:sequence>
            </xsd:complexType>
            </xsd:element>
            <xsd:element name="exportRequestsChanged">
            <xsd:complexType>
            <xsd:sequence>
```

```

        <xsd:element          name="exportReqChanged"          nillable="true"
type="bons3:sendResponseToBP"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
    <xsd:element name="sendPlanSumInfo">
    <xsd:complexType>
    <xsd:sequence>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
    <xsd:element name="sendPlanRequests">
    <xsd:complexType>
    <xsd:sequence/>
    </xsd:complexType>
</xsd:element>
    <xsd:element name="transitImportRequestsChanged">
    <xsd:complexType>
    <xsd:sequence>
    <xsd:element          name="trImpReqChanged"          nillable="true"
type="bons3:sendResponseToBP"/>
    </xsd:sequence>
    </xsd:complexType>
</xsd:element>
    <xsd:element name="sendExportPlanRequests">
    <xsd:complexType>
    <xsd:sequence/>
    </xsd:complexType>
</xsd:element>
    <xsd:element name="gu12RequestsCreated">
    <xsd:complexType>
    <xsd:sequence>
    <xsd:element name="gu12Un" nillable="true" type="xsd:long"/>
    </xsd:sequence>
    </xsd:complexType>
</xsd:element>
    <xsd:element name="gu12RequestsCreatedResponse">
    <xsd:complexType>

```

```

    <xsd:sequence>
      <xsd:element name="response" nillable="true" type="bons1:Message"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
  <xsd:element
    name="gu12RequestsCreatedFault1_faultResponse"
nillable="true" type="bons1:Message"/>
</xsd:schema>
</wsdl:types>
  <wsdl:message name="transitImportRequestsCreatedRequestMsg">
    <wsdl:part
      element="tns:transitImportRequestsCreated"
name="transitImportRequestsCreatedParameters"/>
  </wsdl:message>
  <wsdl:message name="exportRequestsChangedRequestMsg">
    <wsdl:part
      element="tns:exportRequestsChanged"
name="exportRequestsChangedParameters"/>
  </wsdl:message>
  <wsdl:message name="sendPlanSumInfoRequestMsg">
    <wsdl:part
      element="tns:sendPlanSumInfo"
name="sendPlanSumInfoParameters"/>
  </wsdl:message>
  <wsdl:message name="sendPlanRequestsMsg0">
    <wsdl:part
      element="tns:sendPlanRequests"
name="startByScheduleParameters1"/>
  </wsdl:message>
  <wsdl:message name="transitImportRequestsChangedRequestMsg">
    <wsdl:part
      element="tns:transitImportRequestsChanged"
name="transitImportRequestsChangedParameters"/>
  </wsdl:message>
  <wsdl:message name="sendExportPlanRequestsMsg0">
    <wsdl:part
      element="tns:sendExportPlanRequests"
name="startByScheduleParameters11"/>
  </wsdl:message>
  <wsdl:message name="gu12RequestsCreatedRequestMsg">
    <wsdl:part
      element="tns:gu12RequestsCreated"
name="gu12RequestsCreatedParameters"/>
  </wsdl:message>
  <wsdl:message name="gu12RequestsCreatedResponseMsg">

```

```

    <wsdl:part
name="gu12RequestsCreatedResult"/>
    element="tns:gu12RequestsCreatedResponse"
</wsdl:message>
    <wsdl:message name="gu12RequestsCreated_faultResponseMsg">
    <wsdl:part
name="faultResponse"/>
    element="tns:gu12RequestsCreatedFault1_faultResponse"
</wsdl:message>
    <wsdl:portType name="GenerateMonthlyPlanGU12">
    <wsdl:documentation>Generated by WebSphere Business Modeler
V6.2</wsdl:documentation>
    <wsdl:operation name="transitImportRequestsCreated">
    <wsdl:input
name="transitImportRequestsCreatedRequest"/>
    message="tns:transitImportRequestsCreatedRequestMsg"
</wsdl:operation>
    <wsdl:operation name="exportRequestsChanged">
    <wsdl:input
name="exportRequestsChangedRequest"/>
    message="tns:exportRequestsChangedRequestMsg"
</wsdl:operation>
    <wsdl:operation name="sendPlanSumInfo">
    <wsdl:input
name="sendPlanSumInfoRequest"/>
    message="tns:sendPlanSumInfoRequestMsg"
</wsdl:operation>
    <wsdl:operation name="sendPlanRequests">
    <wsdl:input
name="sendPlanRequestsRequest"/>
    message="tns:sendPlanRequestsMsg0"
</wsdl:operation>
    <wsdl:operation name="transitImportRequestsChanged">
    <wsdl:input
name="transitImportRequestsChangedRequest"/>
    message="tns:transitImportRequestsChangedRequestMsg"
</wsdl:operation>
    <wsdl:operation name="sendExportPlanRequests">
    <wsdl:input
name="sendExportPlanRequestsRequest"/>
    message="tns:sendExportPlanRequestsMsg0"
</wsdl:operation>
    <wsdl:operation name="gu12RequestsCreated">
    <wsdl:input
name="gu12RequestsCreatedRequest"/>
    message="tns:gu12RequestsCreatedRequestMsg"

```

```
<wsdl:output message="tns:gu12RequestsCreatedResponseMsg"
name="gu12RequestsCreatedResponse"/>
  <wsdl:fault message="tns:gu12RequestsCreated_faultResponseMsg"
name="faultResponse"/>
</wsdl:operation>
</wsdl:portType>
</wsdl:definitions>
```

Библиотека БГУИР

Перечень принятых терминов

Сервис-ориентированная архитектура (Service-Oriented Architecture, SOA) представляет собой стиль создания архитектуры ИТ, направленный на превращение бизнеса в ряд связанных сервисов – стандартных бизнес-задач, которые можно при необходимости вызывать через сеть.

Портлет – это подключаемый сменный компонент пользовательского интерфейса любого web-портала. Динамический элемент web-страницы, произвольная прямоугольная область, содержимое которой меняется без перезагрузки всей страницы.

Фундамент SOA IBM (IBM SOA Foundation) – это интегрированный открытый набор программного обеспечения, лучшие методы и шаблоны, которые предоставляют все необходимое для создания SOA.

Сервисы – законченные модули программного обеспечения, каждый из которых выполняет определенную бизнес-задачу. У них есть четко определенные интерфейсы, и они не зависят от приложений и платформ, на которых они работают.

ESB (Enterprise Service Bus) – корпоративная сервисная шина. Она обеспечивает связь при значительно меньших затратах, чем у традиционных способов связи. ESB позволяет компании объединить всю внутреннюю и внешнюю инфраструктуру надежным и расширяемым способом.

BPMN (Business Process Modeling Notation) – нотация для наглядного изображения бизнес-процессов.

BPM (Business Process Management) представляет собой стратегию управления и повышения эффективности бизнеса посредством непрерывной оптимизации бизнес-процесса в замкнутом цикле моделирования, выполнения и мониторинга.

BPEL (Business Process Execution Language) – графический язык для реализации бизнес-процессов, который основан на web-сервисах и XML, обеспечивающих открытый и гибкий способ общения в распределенной сети, и задачей которого является объединение сервисов, предоставляющих доступ в среде Интернет.

SCA (Service-Component Architecture) – унифицированные и формализованные компоненты разработки бизнес-приложений.

SDO (Service Data Object) – унифицированные и формализованные структуры передачи данных между компонентами бизнес-приложений.

Web-сервисы (web-службы) – это технология, которая позволяет приложениям взаимодействовать друг с другом независимо от платформы, на

которой они развернуты, а также от языка программирования, на котором они написаны.

Web-сервис – программный интерфейс, описывающий набор операций, которые могут быть вызваны удаленно по сети посредством стандартизированных XML-сообщений. Для описания вызываемой операции или данных используются протоколы, базирующиеся на языке XML.

WSDL (Web Services Description Language) – язык описания web-сервисов, основанный на языке XML.

WebSphere – семейство программных продуктов фирмы IBM. Часто WebSphere употребляется в качестве названия одного конкретного продукта: **WebSphere Application Server (WAS)**. WebSphere относится к категории Middleware – промежуточного программного обеспечения, которое позволяет приложениям электронного бизнеса (e-business) работать на разных платформах на основе web-технологий.

Список использованных источников

1. Биберштейн, Н. Компас в мире сервис-ориентированной архитектуры (SOA) / Н. Биберштейн, С. Боуз. – СПб. : КУДИЦ-Пресс, 2007. – 256 с.
2. Service-Oriented Architecture [Электронный ресурс]. – 2018. – Режим доступа : <http://www.oracle.com/us/products/middleware/soa/overview/>.
3. SOA и web-сервисы. IBM developerWorks Россия [Электронный ресурс]. – 2018. – Режим доступа : <https://www.ibm.com/developerworks/ru/webservices>.
4. Как работает Oracle Service Bus. Oracle [Электронный ресурс]. – 2018. – Режим доступа : <http://www.oracle.com/technetwork/ru/middleware/service-bus/service-bus-1534004-ru.html>.
5. WebSphere Enterprise Service Bus – Library [Электронный ресурс]. – 2018. – Режим доступа : <http://www-01.ibm.com/software/integration/wsesb/library>.
6. Using IBM WebSphere Message Broker as an ESB with WebSphere Process Server / C. Sadtler [et al.]. – IBM Redbooks, 2008. – 368 с.
7. BPMN 2.0 Tutorial / Daniel Brookshier, Distinguished Fellow [Электронный ресурс]. – 2018. – Режим доступа : http://www.omg.org/news/meetings/workshops/SOA-НС/presentations-2011/14_MT-2_Brookshier.pdf.
8. Documents Associated with BPMN Version 2.0.1 [Электронный ресурс]. – 2018. – Режим доступа : <http://www.omg.org/spec/BPMN/2.0.1>.
9. Использование WebSphere Business Modeler для создания модулей бизнес-интеграции в среде WebSphere Integration Developer [Электронный ресурс]. – 2018. – Режим доступа : https://www.ibm.com/developerworks/ru/library/0611_wen/index.html.
10. Transforming BPMN into BPEL: Why and How. Oracle [Электронный ресурс]. – 2018. – Режим доступа : <http://www.oracle.com/technetwork/articles/dikmans-bpm-101437.html>.

Учебное издание

Пилецкий Иван Иванович
Козуб Виктор Николаевич

**ТЕХНОЛОГИЯ ИНТЕГРАЦИИ ДАННЫХ
КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ.
ОСНОВЫ ТЕХНОЛОГИИ РЕАЛИЗАЦИИ
БИЗНЕС-ПРОЦЕССОВ ИНФОРМАЦИОННЫХ
СИСТЕМ, ЯЗЫКИ BPMN И BPEL**

ПОСОБИЕ

Редактор *М. А. Зайцева*
Корректор *Е. Н. Батурчик*
Компьютерная правка, оригинал-макет *М. В. Касабуцкий*

Подписано в печать 25.09.2019. Формат 60×84 1/16. Бумага офсетная. Гарнитура «Таймс».
Отпечатано на ризографе. Усл. печ. л. 4,53. Уч.-изд. л. 4,5. Тираж 150 экз. Заказ 137.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий №1/238 от 24.03.2014,
№2/113 от 07.04.2014, №3/615 от 07.04.2014.
Ул. П. Бровки, 6, 220013, г. Минск