

Язык VPEL в системах автоматизации бизнес-процессов

Илья Кулешов, 17 ноября 2006 г.

NAUMEN
разработка программного обеспечения

Содержание

- Что такое BPEL?
- BPEL и Workflow
- Интеграция с BPEL

Компания NAUMEN

- **NAUMEN** – разработка решений для бизнеса и органов власти
(Москва, Екатеринбург, Тверь, Челябинск)
- Мы **используем VPEL** в своих решениях и делимся опытом внедрения
- Предпочтение **Open Source** – независимость от поставщиков платформы

Постановка задачи

Дано:

- Приложения собственного производства (документооборот, автоматизация тех. поддержки, CRM, госзакупки и пр.)
- Сторонние приложения (бухгалтерия, биллинг...)

Задача:

- **Интегрировать** разнородные системы
- **Реализовать workflow** с участием различных систем

Введение в BPEL

Введение в BPEL

- BPEL: Business Process Execution Language
- Язык исполнения бизнес-процессов
- Длительные **асинхронные диалоги**
- Управляет вызовом **сервисов**
- Процесс не только взаимодействует с сервисами, он **сам предоставляет сервисы**
- Описывается **XML**, оперирует XML-данными
- Логика бизнес-процессов вынесена из программного кода, доступна для изменения **технологом** (не программистом)
- BPEL – ключевой компонент **SOA**

BPEL – открытый стандарт

- Действующая версия BPEL 1.1 (2003 г.)
- Разрабатывается WS-BPEL 2.0 (OASIS)

Инструментарий VPEL

Возможности языка:

- Поддержка асинхронного взаимодействия (Invoke/Reply)
- Условные переходы (Switch)
- Последовательное/одновременное выполнение действий
- Циклы
- Копирование, присваивание переменных (с помощью XPath)

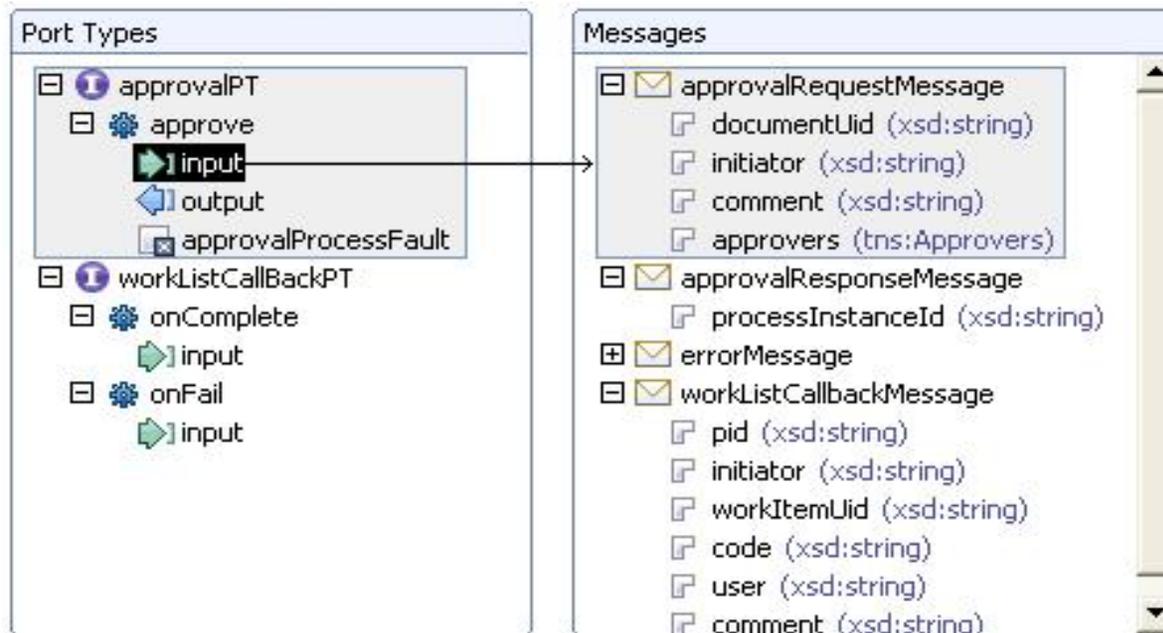
Инструментарий VPEL (2)

Возможности языка (продолжение):

- Ожидание событий (входящее сообщение, таймер)
- Механизм компенсаций (откат действий при ошибке)
- Обработка исключений
- Корреляция взаимодействий (связь между сообщением и экземпляром процесса)
- Графовые конструкции (явные связи между действиями, связи могут иметь условия перехода).

Процесс – это сервис

- Процесс – тоже сервис
- Веб-сервис должен иметь WSDL-описание
- WSDL описывает программный интерфейс сервиса
- Сервисы BPEL не исключение, поэтому технолог должен создать WSDL-документ (Eclipse Web Tools)



BPЕL и Java

- Java-код в BPЕL-процессах – неплохая идея
- XPath – только для работы с XML, предоставляет минимум функций
- Зачем создавать сервисы для простых действий?
- Кто хочет писать XSLT трансформации?
- Для программиста BPЕL прост, но не для технолога/аналитика
- Нужно упрощать работу с BPЕL!

Способы упрощения:

- Возможно использование JavaScript, Python, Ruby в выражениях (стандарт BPЕL разрешает)
- BPЕLJ – расширение BPЕL, Java-код в BPЕL-процессе

Реализации BPEL-серверов

Открытые

- Active Endpoints ActiveBPEL
- Apache ODE (слияние Intalio PXC и Sybase BPEL engine)
- JBoss jBPM BPEL Extension
- SUN (SeeBeyond)

Коммерческие

- Oracle BPEL Manager
- Intalio BPM
- Active Endpoints ActiveBPEL Enterprise
- многие другие

ActiveBPEL

ActiveEndpoints ActiveBPEL 3.0, сервер

- Поддержка WS-BPEL 2.0
- Открытая (Open Source) и коммерческая версия
- Нет поддержки UDDI
- Версионность процессов доступна только в коммерческой версии

ActiveBPEL Designer, средство проектирования BPEL-процессов

- plugin для Eclipse IDE
- Интегрировано с ActiveBPEL
- Поддерживает отладку
- Автоматизированная установка процессов на сервер
- Исходные коды дизайнера недоступны

Альтернатива развития

Apache ODE

+

Eclipse BPEL Project (Eclipse Foundation)

- Open Source
- Наличие исходных кодов обоих продуктов позволяет создать специализированный инструмент
 - интеграция с собственными системами
 - возможность создать Domain Specific Language
- Требуется большое количество доработок

BPEL и Workflow

BPЕL и Workflow

- BPEL – развитое средство управления веб-сервисами
- BPEL – не workflow, в нем нет сущностей «задача», «документ», «клиент»

BPEL и Workflow (2)

На базе BPEL построен Workflow:

- Интеграция с системой электронного документооборота NAUMEN DMS (сервисы работы с документами)
- Создан компонент «Диспетчер задач» (Worklist Manager)

Диспетчер задач

- Процесс взаимодействует с пользователем через задачи
- Управление задачами – диспетчер задач
 - отдельное приложение
 - внешние интерфейсы доступны через веб-сервисы
 - пользователь видит свои задачи в соотв. приложениях
 - Соответствует рекомендациям BPEL4People (авторство SAP и IBM, 2005 г.)

Документы	Оргструктура	Мои настройки	Мои выписанные документы(0)	Мои задачи
Новые задачи				
	Задача	Дата начала	Дата окончания	
	Рассмотреть заявку ОК-20061005-23	26.10.2006 10:04	27.10.2006 00:00	
	 Устранить неисправность бетономешалки			

Диспетчер задач

Жизненный цикл задачи:

- Задача назначается потенциальным исполнителям;
- Любой из потенциальных исполнителей принимает задачу и становится ее владельцем;
- Владелец может отказаться от выполнения задачи, тогда ее может принять к исполнению другой пользователь;
- У задачи может быть только один владелец;

Диспетчер задач

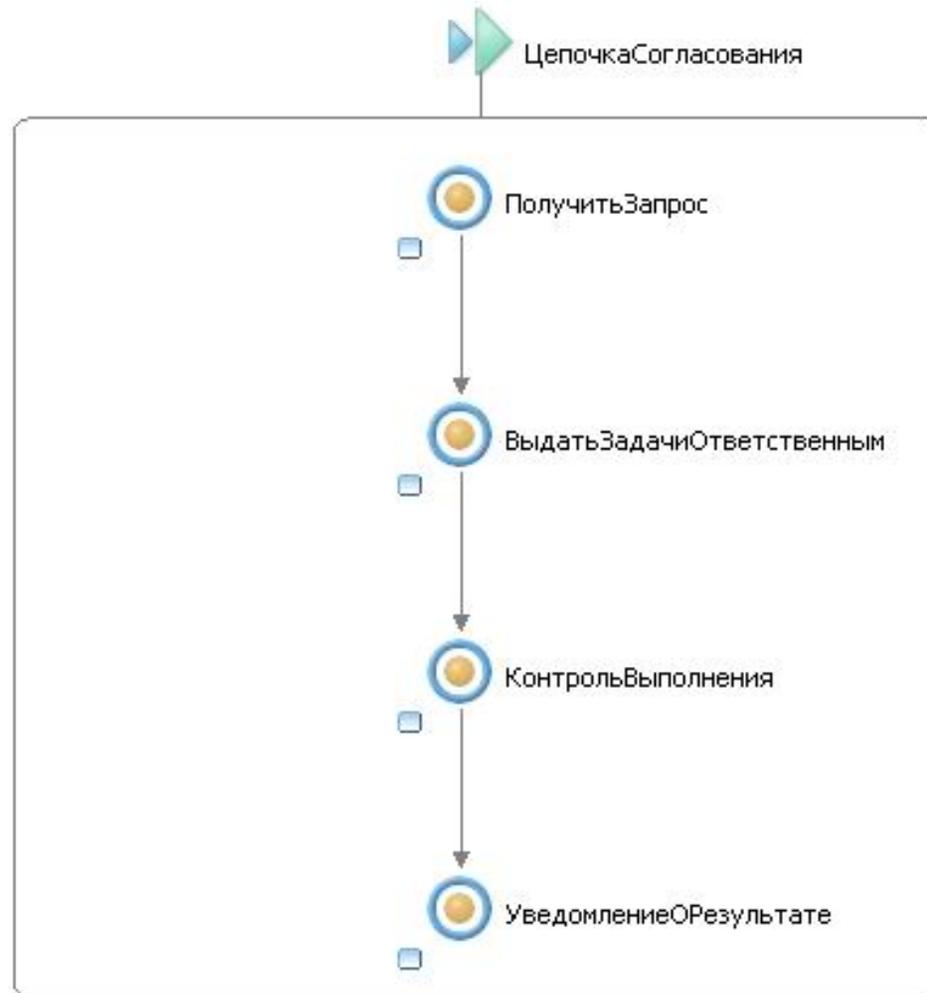
Жизненный цикл задачи (продолжение):

- Владелец может завершить задачу как «успешную» либо как «неуспешную»;
- О результате выполнения задачи уведомляется породивший задачу BPEL-процесс;
- Логика выполнения бизнес-процесса может зависеть от результата выполнения задач;
- Задачи не принимают решений.

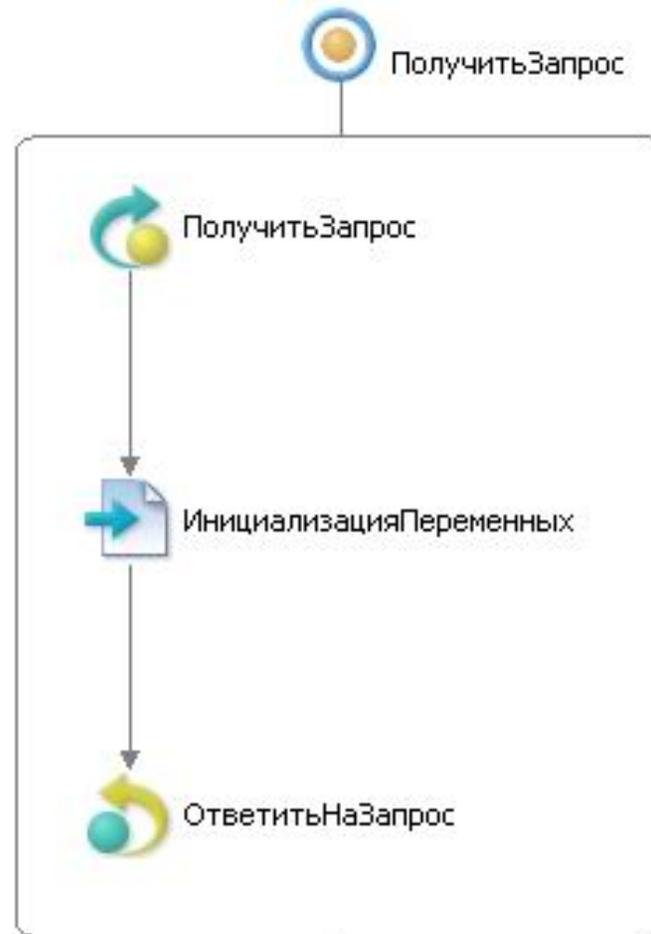
Пример

Визуальное проектирование

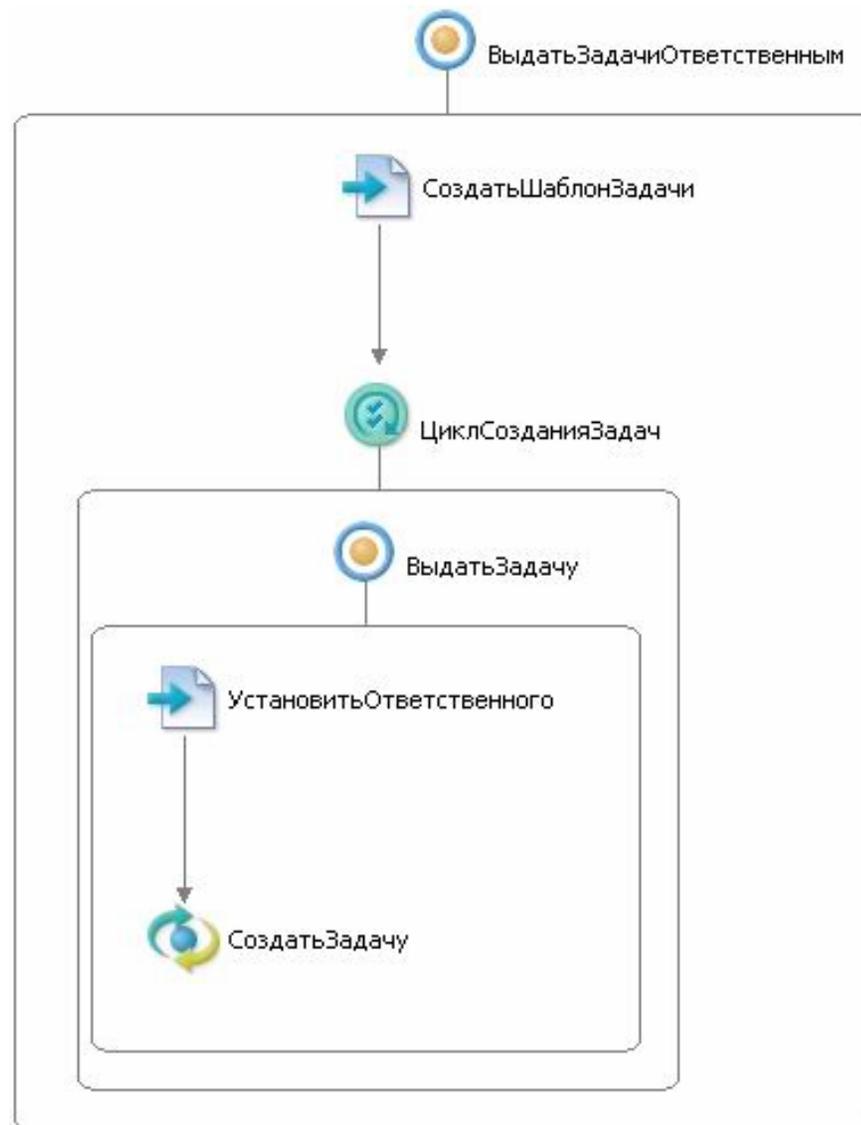
Процесс согласования документа



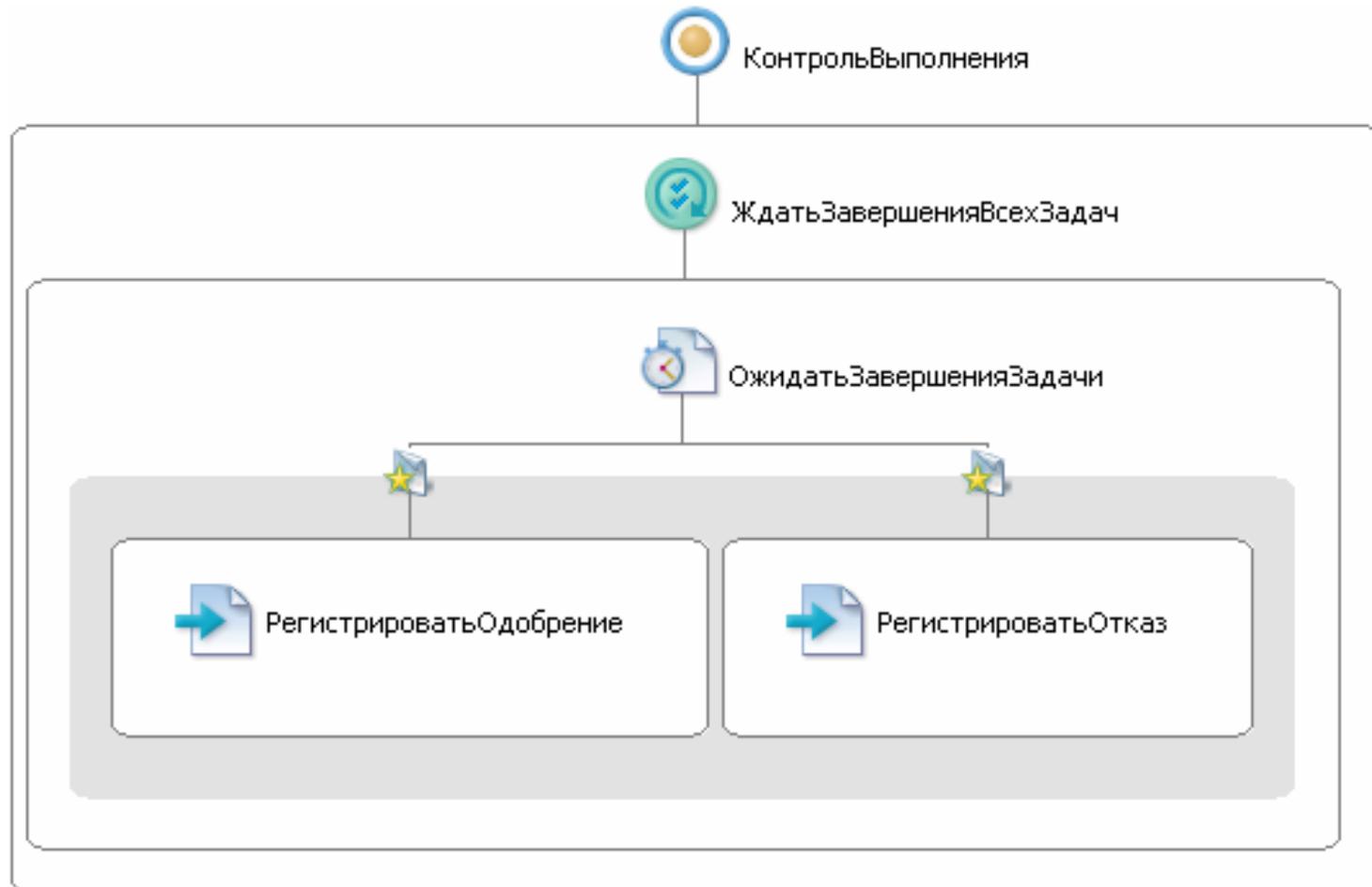
Визуальное проектирование



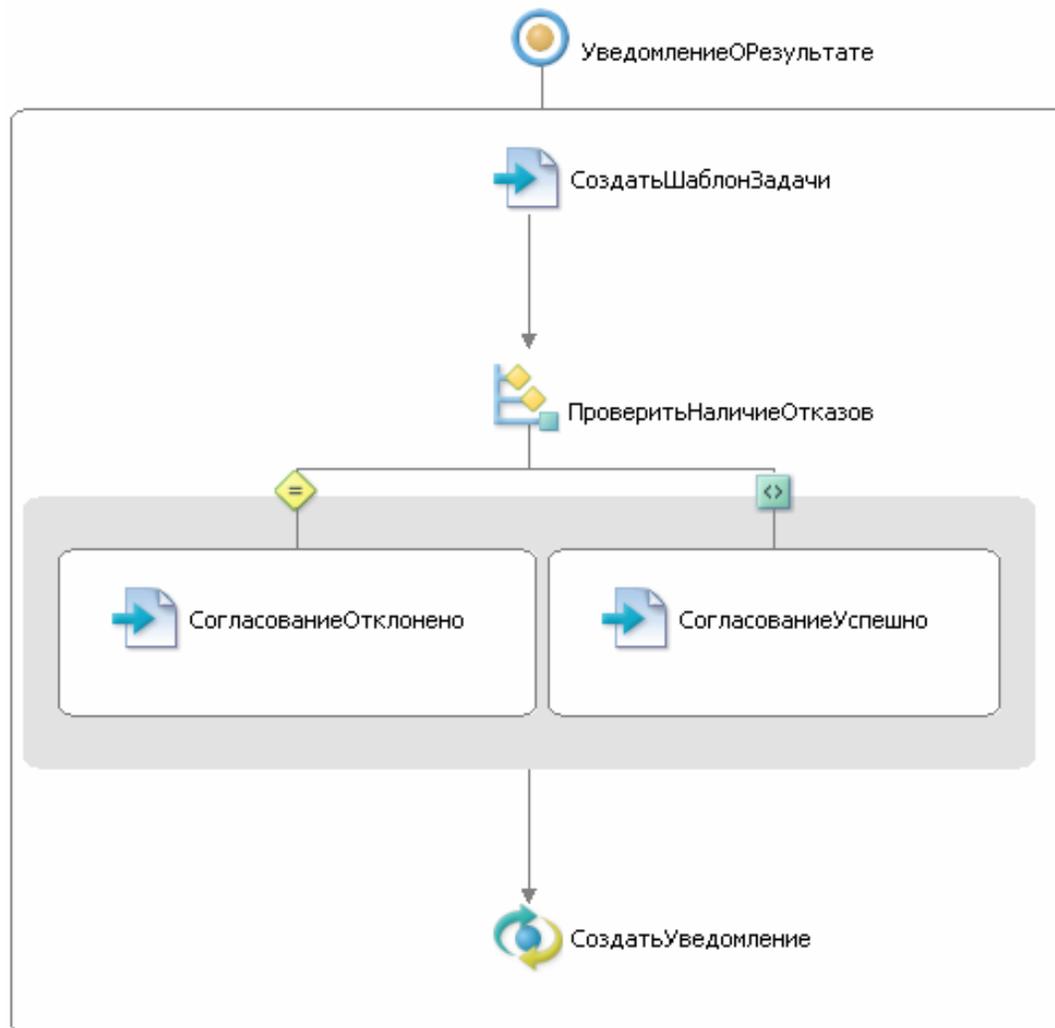
Визуальное проектирование



Визуальное проектирование



Визуальное проектирование





Интеграция ВРЕЛ и собственного ПО (SECR – конференция разработчиков!)

Интеграция с ВРЕЛ

Требование:

- В собственном приложении иметь **список установленных бизнес-процессов**
- Пользователь имеет возможность **запустить процесс** (создать новый экземпляр процесса)
- Перед запуском у пользователя **запросить начальные параметры** (ссылка на согласуемый документ, дату окончания согласования, список согласователей и пр.)
- Отображать **список выполняющихся, завершенных процессов**
- Отображать **состояние запущенных экземпляров процессов**

Интеграция с BPEL

Для создания экземпляра BPEL-процесса:

- отправить запрос к определенному веб-сервису
- передать ему входные данные

Каждый сервис, предоставляемый BPEL-процессом, имеет собственный адрес и программный интерфейс (определяется WSDL)

- 1. Каким образом узнать адрес сервиса для запуска процесса?**
- 2. Какие входные данные передать процессу (в каком формате)?**

Интеграция с BPEL

- Неприемлемо реализовывать поддержку API каждого бизнес-процесса в программном коде системы
- BPEL не рассматривает вопросы управления процессами и предоставления вспомогательной информации
- Нет стандартного способа получить информацию о процессах

Реестр процессов

Решение проблем поручено компоненту
«Реестр процессов»:

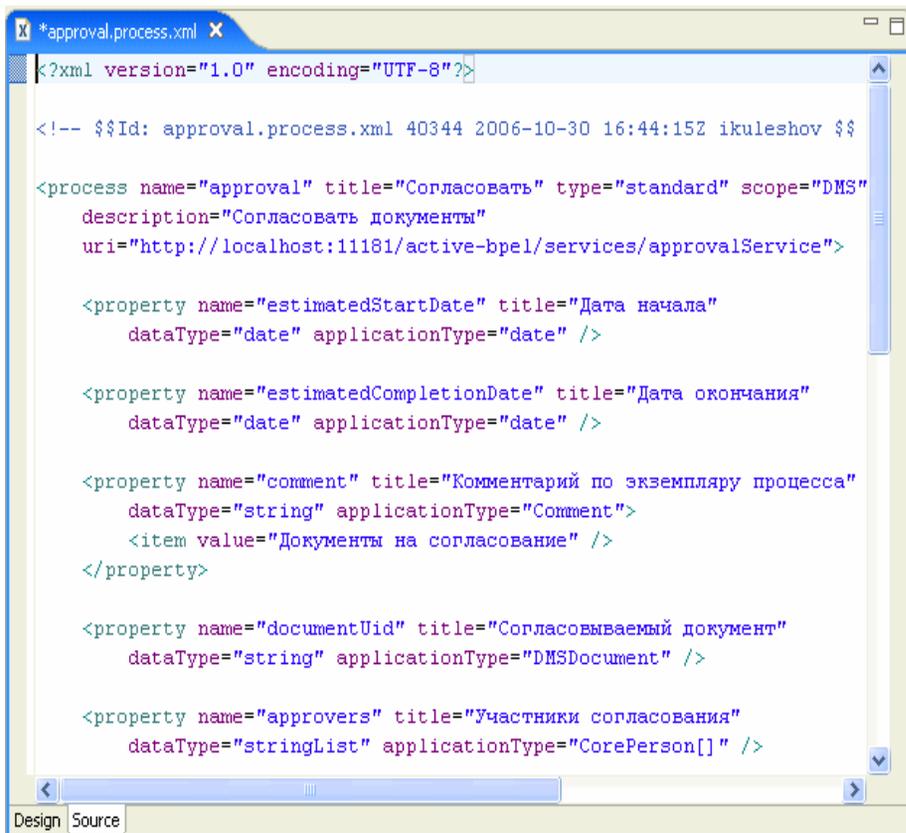
- Создает новые экземпляры процессов (по просьбе других приложений)
- Ведет учет запущенных экземпляров
- Предоставляет информацию об установленных в системе процессах
- Изолирует остальные приложения системы от особенностей реализации BPEL-сервера

Реестр процессов

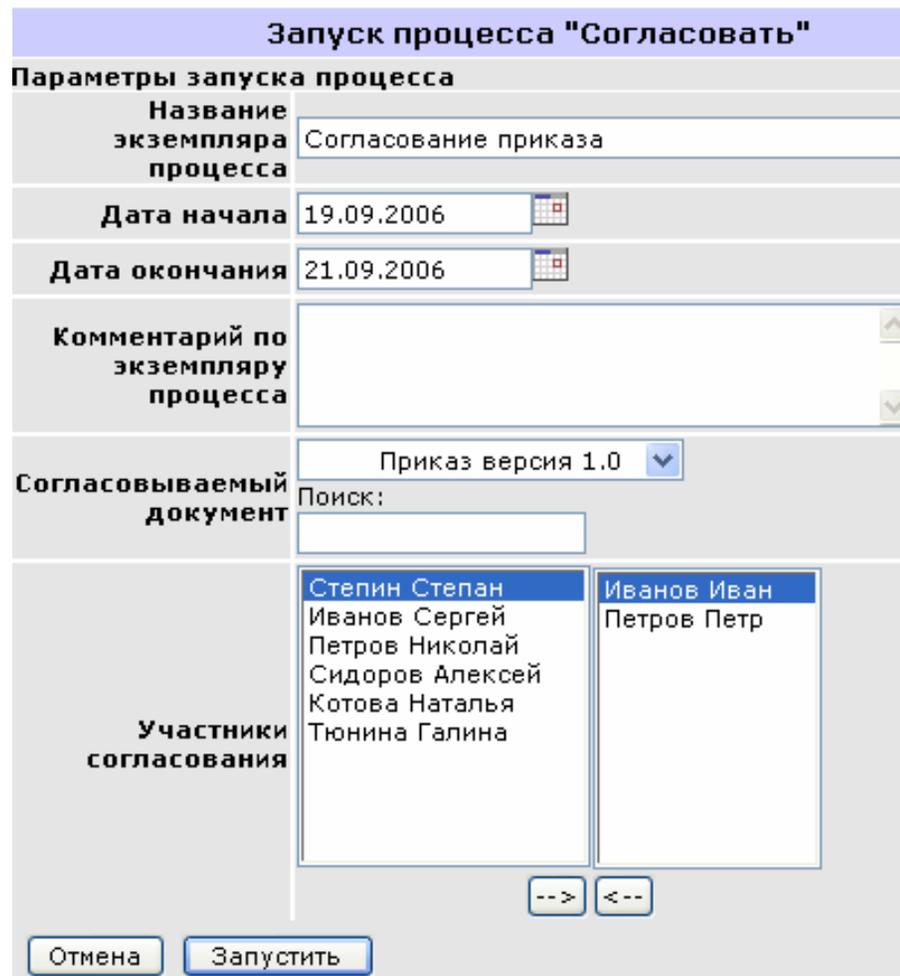
- Реестр сервисов получает расширенную информацию о процессе из **дескриптора процесса**.
- Дескриптор (XML-файл) создается технологом.
- Содержание дескриптора процесса:
 - Адрес сервиса для запуска экземпляра процесса
 - Описание начальных параметров (название, тип, способ отображения в интерфейса пользователя)
 - Инструкции к процессу
 - Параметры безопасности

Реестр процессов

Дескриптор и форма запуска процесса, созданная на базе дескриптора



```
<?xml version="1.0" encoding="UTF-8"?>
<!-- $$Id: approval.process.xml 40344 2006-10-30 16:44:15Z ikuleshov $$ -->
<process name="approval" title="Согласовать" type="standard" scope="DMS"
description="Согласовать документы"
uri="http://localhost:11181/active-bpel/services/approvalService">
  <property name="estimatedStartDate" title="Дата начала"
dataType="date" applicationType="date" />
  <property name="estimatedCompletionDate" title="Дата окончания"
dataType="date" applicationType="date" />
  <property name="comment" title="Комментарий по экземпляру процесса"
dataType="string" applicationType="Comment">
    <item value="Документы на согласование" />
  </property>
  <property name="documentUId" title="Согласовываемый документ"
dataType="string" applicationType="DMSDocument" />
  <property name="approvers" title="Участники согласования"
dataType="stringList" applicationType="CorePerson[]" />
</process>
```



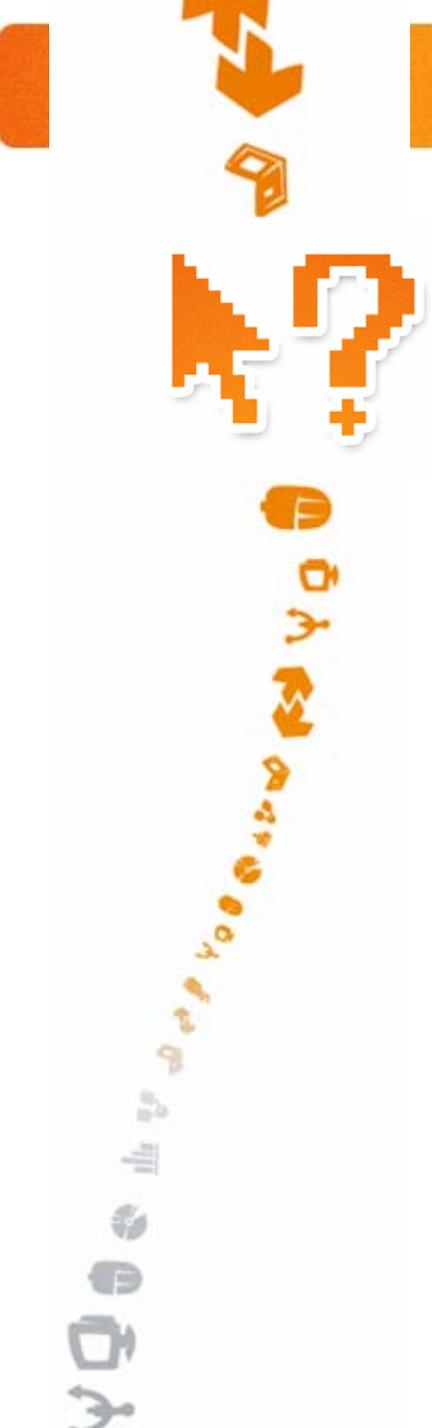
Запуск процесса "Согласовать"

Параметры запуска процесса

Название экземпляра процесса	<input type="text" value="Согласование приказа"/>
Дата начала	<input type="text" value="19.09.2006"/>
Дата окончания	<input type="text" value="21.09.2006"/>
Комментарий по экземпляру процесса	<input type="text"/>
Согласовываемый документ	<input type="text" value="Приказ версия 1.0"/>
Участники согласования	<input type="text" value="Степин Степан, Иванов Иван, Иванов Сергей, Петров Николай, Сидоров Алексей, Котова Наталья, Тюнина Галина"/>

Итого

- BPEL:
 - не замена для традиционных нотаций. Аналитик продолжает пользоваться ARIS/BPMN/IDEF, преобразование в BPEL выполняет технолог
 - чудес не бывает, BPEL – это не решение проблемы BPM/Workflow в один момент, работы хватит надолго
 - не программирование мышкой для «чайников»
 - требует развитой инфраструктуры корпоративных веб-сервисов
 - несовершенен, но является лучшей альтернативой для основы BPM на текущий момент
 - позволяет вынести бизнес-логику из приложения и использовать при этом опыт и наработки большого комьюнити



Спасибо, вопросы?

За дополнительной информацией
обращайтесь:

ikuleshov@naumen.ru

Схема решения

Итак, общая схема реализации:

