

**Создание системы
электронного
документооборота на
основе рабочих потоков**

Ярёменко Ф. В. (ВЦ РАН)

Компоненты СЭД

- Поддержка территориально-распределённых структур организаций предприятий.
- Управление поручениями и резолюциями. Назначение сроков исполнения и ответственных лиц позволяет контролировать выполнение процессов документооборота.
- ЭЦП для юридически-значимого документооборота.
- Разграничение доступа к данным и функционалу системы на основе авторизации пользователей.
- Поддержка иерархии и связей между документами.
- Управление версиями документов.
- Web-интерфейс, который даёт возможность использовать систему на любой рабочей станции, в том числе с ПО с открытым исходным текстом.
- В СЭД кроме автоматизации документооборота и делопроизводства также должны входить подсистема электронного архива и нормативно-информационная система.
- Поддержка контекстного поиска в теле документов.
- Графический редактор маршрутов.

Этапы работы

- Построение модели документооборота
- Реализация ядра системы и разработка API
- Создание основных типов узлов маршрутов
- Подсистема контроля исполнения и оповещения
- Web-интерфейс

Начальный этап работы

На начальном этапе предлагается определить движение документа как процесс, в котором переход от одного узла к другому происходит в одной из трёх ситуаций:

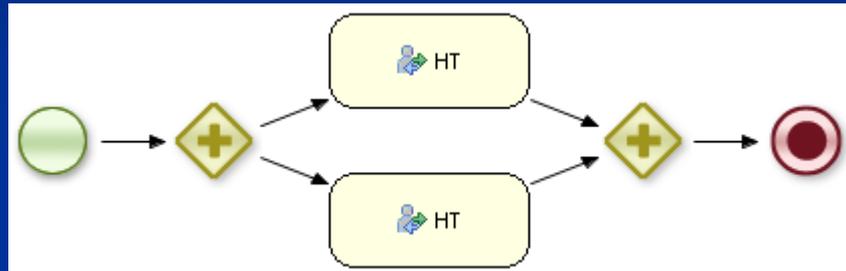
- Изменение состояния документа (состояния документа могут быть разными, и их набор определяется моделью; например: на рассмотрении, подписан, дорабатывается и т.д.)
- Изменение ответственного лица или лиц
- События по таймеру
- Списание документа в архив

Структура системы

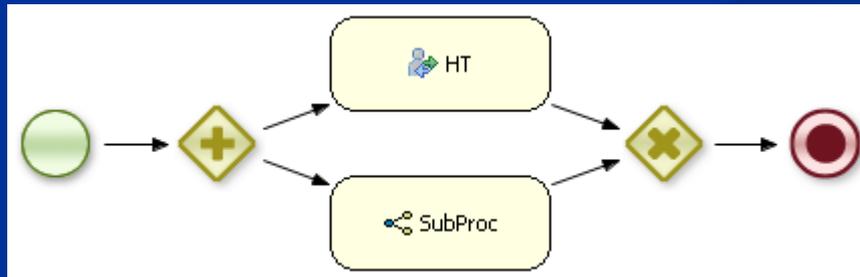
1. Пользовательский уровень работы с документами
2. Уровень управления рабочими потоками
3. ИТ-инфраструктура

Базовые элементы потоков (1)

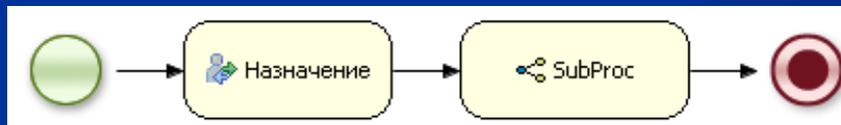
- Ожидание завершения всех задач



- Ожидание любой задачи

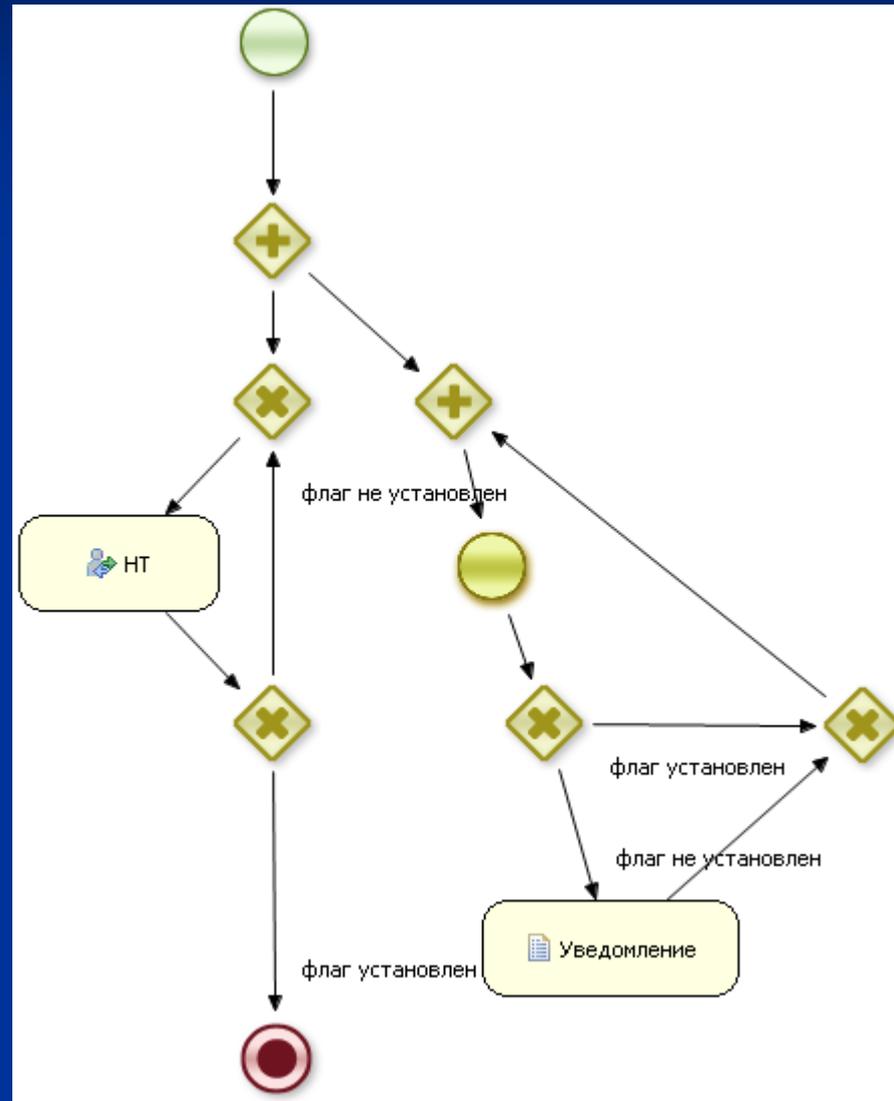


- Назначение исполнителя



Базовые элементы потоков (2)

- Установка флага состояния документа



Базовые элементы (обозначения)

- начало процесса
- окончание процесса
- 👤 задача пользователя
- 🔗 подпроцесс
- ⚡ переход по одной из исходящих веток по условию или переход по исходящей ветке при завершении любой из входящих
- ⚡ параллельное выполнение всех исходящих веток или переход по исходящей ветки при завершении всех входящих
- ⌚ задержка по таймеру
- 📄 выполнения действия системой

Бизнес-правила

Бизнес-правило – это выражение, определяющее некоторые аспекты бизнеса. Бизнес-правила могут применяться к людям, процессам, корпоративному поведению и вычислительным системам.

Пример простого бизнес-правила:

```
package droolsbook;
```

```
rule "basic rule"
```

```
when
```

```
    Account( balance < 100 ) // условие
```

```
then
```

```
    System.out.println("Ваш баланс меньше 100 р.");
```

```
    //результат
```

■ End

Преимущества бизнес-правил

- Правила проще в понимании для бизнес-аналитиков и начинающих программистов, чем написание программ на Java или другом языке.
- Так как правила более просты для понимания, у разработчика остаётся больше времени для решения актуальных задач.
- Благодаря алгоритму Rete (*лат. «сеть»*). Эффективный алгоритм сопоставления с образцом для продукционных систем, экспертных систем и баз знаний, созданный Чарльзом Форги в 1974 г.), который используется в Drools, в теории, производительность системы не зависит от количества правил.
- С каждой версией Drools производительность выполнения правил возрастает благодаря добавлению различных улучшений, таких как параллельное исполнение, индексированные узлы, упомянутый выше алгоритм Rete и др.
- Достаточно просто вызвать правила из программы и наоборот.
- Независимые жизненные циклы правил и системы.
- Drools легко может быть встроен в уже существующую систему.

Недостатки бизнес-правил

- Если проект небольшой, примерно с 20 правилами и меньше.
- Если бизнес-логика изменяется редко.
- Если почти все правила очень просты и в основном охватывают один объект. Если псевдо код большинства задач, описываемых правилами, не содержит больше двух выражений if-then.
- Если не планируется вторичное использование правил.

JBoss Drools

Drools обладает следующими преимуществами перед jBPM 4:

- широкий набор узлов в диаграммах для моделирования бизнес-логики
- простой способ создания новых предметно-ориентированных узлов, интегрированных в графический редактор
- средства отладки правил; в среде разработки eclipse можно выполнять правила шаг за шагом и создавать точки останова.
- использование концепции WS-NT
- использование нотации BPMN 2
- web-редактор с полностью открытым исходным текстом
- улучшенная поддержка MVEL

Спасибо за внимание.