Взаимодействие с Activiti

Предполагается использование Activiti в системе CM5 для управления различными процессами, предполагающими взаимодействие между различными пользователями системы.

Основная идея использования бизнес-процессов Activiti заключается в том, чтобы прохождение документов в системе было максимально автоматизировано и требовало минимальных усилий со стороны пользователей (отказ от ручных передач с этапа на этап). С помощью процессов можно настроить контроль сроков выполнения задач и рассылку уведомлений пользователям. Графический редактор бизнес-процессов представляет их в виде, удобном для понимания, а отчасти и для изменения аналитикам (не-программистам). Кроме этого, API Activiti позволяет легко построить административный интерфейс для контроля состояния бизнес-процессов в системе.

## Технические аспекты

Каждый процесс привязан к карточке системы. Если процесс может менять статус карточки, то он называется управляющим процессом. К одной карточке в каждый момент времени может быть привязан только один управляющий процесс.

Идентификатор экземпляра управляющего процесса сохраняется в специальном поле карточки. По завершению процесса это поле очищается. Объект карточки делается доступным коду задач через переменные процесса.

Основные типы задач (активностей), из которых будут состоять процессы, это:

* UserTask – задачи, назначаемые пользователям (например, «Согласовать»). Activiti позволяет задавать в настройках задачи пользователей, которым будет назначена эта задача (в т.ч. через переменные процесса). Предполагается добавить собственные настройки, указывающие возможные варианты завершения пользователем задачи («Согласен», «Не согласен», «Согласен с замечаниями»). Назначенные пользователю задачи будут отображаться в специальной(ых) папке(ах), для каждой будут предложены кнопки в соответствии с вариантами завершения задачи. При выборе пользователем одного из этих действий код системы информирует Activiti о завершении задачи, и выполнение процесса продолжается.
* ServiceTask – задачи, выполняемые системой автоматически, некоторый Java-код (класс), указанный в настройках задачи. Предполагается разработать ряд стандартных «кубиков» - классов, выполняющих типовые действия, – чтобы составлять из них различные процессы. Такими действиями могут быть: изменение статуса карточки, вычисление полей, создание дочерней(их) карточки(ек), добавление карточки в папку и т.п. В некоторых случаях понадобятся также специальные классы, выполняющие некоторые нестандартные действия, например, регистрация документа, импорт документа из МЭДО и др.
* Sub-process call – вызовы других процессов. Планируется использовать для вызова стандартных процессов (как согласование, подписание) из макро-процессов, управляющих жизненным циклом документа в целом (например, внутреннего).

Возможно использование и других типов задач, предлагаемых Activiti:

* EmailTask – отсылка сообщений (например, уведомлений) по электронной почте.
* ScriptTask – возможно, простые задачи вроде изменения полей карточки документа будет проще описывать на скриптовом языке (JavaScript, Groovy), а не создавать собственный простой класс со сложными настройками.
* JavaReceiveTask – приостановка процесса до получения сигнала от Java-кода. В некоторых случаях может оказаться удобнее, чем посылка сообщений.

Создание процесса для карточки может производиться:

1. пользователем через вызов специального действия. Как и любое другое действие в системе, оно задаётся для определённого типа карточки и определённого статуса, а также является объектом назначения прав доступа;
2. системой в процессе обработки других действий пользователя или выполнения других процессов (включая периодические).

## Процессы для обеспечения ЖЦ внутреннего документа

Таким образом, для реализации жизненного цикла внутреннего документа видится необходимым создание следующих процессов в Activiti:

1. Общий ЖЦ внутреннего документа: согласование (подпроцесс), подписание (подпроцесс), регистрация (ServiceTask), исполнение (включая рассмотрение – подпроцесс), передача в дело (Service или UserTask – по мере необходимости). Это управляющий процесс карточки внутреннего документа. Следует отметить, что в нём отсутствуют логические этапы подготовки документа и ознакомления. Первый из них слабо формализован и либо не ограничен по срокам и не требует контроля со стороны системы, либо такое ограничение наложено другой задачей (выданным поручением) и контроль сроков осуществляется в рамках той задачи. Что же касается ознакомления, то оно может проводиться в разные моменты ЖЦ документа (после регистрации) и инициируется пользователем, т.е. является просто самостоятельным неуправляющим процессом для карточки документа. Впрочем, если регламент работы с внутренними документами будет предусматривать автоматическую отправку документа на ознакомление при выполнении заданных условий, то, разумеется, будет целесообразным включить этот процесс в общий ЖЦ документа.
2. Согласование. Содержит схему процесса, позволяющую реализовать комбинированный (последовательный + параллельный) процесс согласования документа. При этом в документе для каждого согласующего создаётся специальная дочерняя карточка согласования, в которой он и будет указывать свою визу. Основная задача, входящая в этот процесс – UserTask для согласующего, для завершения которой ему необходимо заполнить форму визы. В зависимости от решения согласующих, процесс перед завершением установит статус документа в «Согласовано», либо в «Подготовка»/«Доработка».
3. Подписание. Процесс во многом похож на согласование, только основной UserTask предполагает выдачу формы «Подписание», также связанной с соответствующей дочерней карточкой документа, а статус документа, устанавливаемый при успешном завершении процесса – «Подписано».
4. Исполнение документа. Этот подпроцесс объединяет в себе процедуры наложения резолюций (рассмотрение документа), отправки их на исполнение и контроля исполнения. При этом для лица, рассматривающего документ (адресата или руководителя) представляется разумным создать карточку рассмотрения, по которой ему будет поставлена задача рассмотрения документа. Резолюции/поручения создаются в виде отдельных карточек в системе (привязанных к документу, конечно), и для каждой создаётся отдельный процесс исполнения поручения (см. ниже). Процесс исполнения документа ожидает завершения задачи рассмотрения и всех дочерних процессов, созданных в ходе выполнения этой задачи, устанавливает статус «Исполнен» или «Готов к списанию в дело» и завершается.
5. Исполнение поручения – управляющий процесс для карточки поручения/резолюции. Включает UserTasks для исполнителя поручения (исполнить) и контролёра (утвердить отчёт об исполнении), ServiceTasks для изменений статуса карточки поручения, рассылки уведомлений. С помощью таймерных событий можно контролировать сроки исполнения поручения (например, для рассылки дополнительных уведомлений). Процесс исполнения поручения может также порождать собственные копии в качестве подпроцессов для реализации процедуры исполнения дочерних (связанных) поручений.
6. Ознакомление. Один из самых простых процессов, главная часть которого – UserTask для лиц, которые должны открыть документ и нажать кнопку «Ознакомлен». Вопрос о том, должен ли этот процесс включаться в качестве подпроцесса в общий ЖЦ документа, или остаться отдельным неуправляющим процессом для карточки внутреннего документа, подробно рассматривался выше.

## О карточках и статусах

Один из важных моментов, который необходимо понимать в архитектуре CM5, это то, что понятие карточки (BusinessObject, как он сейчас назван в коде) не полностью совпадает с понятием РКК в предметной области. Карточками в системе представляются и РКК, и различные другие объекты, включая достаточно мелкие. Например, к карточке внутреннего документа привязана масса дочерних карточек, таких как карточки согласования (по одной на каждого согласующего и каждую итерацию согласования), подписания, ознакомления, поручений и т.п. Кроме того, по причине отсутствия полей со множественными значениями они также заменяются дочерними мини-карточками.

Каждая карточка имеет обязательное поле статуса, главное назначение которого – определение прав доступа к ней. Изменение поля статуса через операцию сохранения карточки невозможно, для этого в API предусмотрены отдельные операции. Главным инициатором выполнения этих операций должны быть задачи процессов Activiti. Для некоторых переходов можно также создать бизнес-действия, которые будут доступны пользователям в соответствии с правами доступа. Однако, важно не допускать неявного подключения сложного кода, реагирующего на такие «ручные» перехода, чтобы логика системы оставалась понятной и хорошо поддерживаемой.

Поскольку некоторые (но не все!) события смены статуса документа представляют особый интерес с точки зрения его истории, целесообразно предусмотреть возможность протоколирования таких переходов в виде отдельных дочерних карточек (истории).