Уведомления

Главной чертой подсистемы уведомлений в системе CompanyMedia версий 4.x является её тесная взаимосвязь с бизнес-логикой документооборота, с механизмами контроля исполнительской дисциплины. В рамках разработки уведомлений для CM 4/5 представляется разумным провести разделение функций между платформой и Сочи-сервером. Платформа должна взять на себя задачи, универсальные для большинства информационных систем различного назначения. К таким задачам можно отнести:

* формирование уведомлений по шаблонам при различных событиях в системе (в первую очередь, связанных с изменением данных);
* рассылку уведомлений по различным каналам, включая внешние (e-mail, SMS, мгновенные сообщения) и внутрисистемные (создание специальных доменных объектов, всплывающие окна в GUI);
* форматирование и локализацию сообщений;
* фильтрацию и, возможно, агрегирование сообщений в соответствии с системной политикой и настройками пользователя.

Последние три функции (рассылку, форматирование и фильтрацию) предполагается возложить на специальную службу уведомлений. Важно отметить, что служба уведомлений не является активным компонентом, т.е. инициатором посылки уведомлений. Как и любой другой сервис, она выполняет свою работу только в ответ на запросы от других частей системы или её клиентов.

Формирование уведомлений, т.е. определение того, кому, когда и какие уведомления должны посылаться, является функцией бизнес-логики системы. Платформа предоставляет несколько способов реализации бизнес-логики.

1. Конфигурация. Она позволяет использовать стандартные компоненты, входящие в платформу, и реализующие некоторые обобщённые функции, необходимые для информационных систем различного назначения. С точки зрения уведомлений, наиболее типичными (и необходимыми для системы электронного документооборота) являются посылка сообщений при некоторых изменениях документа лицам, связанным с этим документом определённым образом, а также напоминания о событиях намеченных на некоторую дату или время. Подсистема уведомлений в платформе ECMp должна включать компоненты, выполняющие эти задачи и настраиваемые через конфигурацию.
2. Бизнес-процессы (BPMN). Отправка уведомления – одна из наиболее распространённых задач (task), включаемых в бизнес-процесс. Платформа должна включать компонент, реализующий эту задачу (task) с использованием службы уведомлений.
3. Бизнес-модули (Java-код). Это наиболее гибкий, но и наиболее трудоёмкий вариант реализации бизнес-логики системы. Он используется в тех случаях, когда бизнес-логика не может быть реализована с использованием других методов. Бизнес-модули могут обращаться непосредственно к службе уведомлений, поэтому дополнительный компонент в платформе для этого сценария не требуется.

Таким образом, в платформу включаются только компоненты, реализующие механизмы генерации и рассылки уведомлений. Конфигурация событий и шаблонов уведомлений, а также бизнес-процессы и бизнес-код, содержащие обращения к службе уведомлений, помещаются в бизнес-модули, реализующие функционал конкретной информационной системы (например, СЭД CompanyMedia).

# Важные аспекты уведомлений

## Каналы доставки

Уведомления могут доставляться пользователям различными способами. Среди них могут быть как способы, требующие привлечения внешних ресурсов (e-mail, SMS, мгновенные сообщения), так и чисто внутрисистемные (создание специальных доменных объектов, всплывающие окна в GUI). Набор каналов может отличаться от системы к системе и даже от установки (installation) к установке. Некоторые каналы могут поставляться в виде отдельных модулей или разрабатываться специально для конкретного заказчика.

## Управление доставкой

Выбор канала доставки уведомления может зависеть как от самого сообщения (его типа и важности), так и от настроек пользователя или системной политики. Одно и то же уведомление может доставляться разными способами в разных случаях (например, всплывающее окно, если пользователь вошёл в систему, и e-mail в ином случае), и может отправляться несколькими способами одновременно. Кроме того, зачастую полезно предоставить возможность пользователю с помощью персональных настроек выбирать каналы доставки уведомлений и необходимость их отправки в зависимости от типа. При этом нужна возможность ограничивать выбор пользователя некоторой системной политикой, задаваемой администратором.

## События

Теоретически, уведомления могут отправляться в самые разные моменты времени. На практике же подавляющее большинство уведомлений формируются при изменении данных в системе. Поскольку все бизнес-данные платформа хранит в виде доменных объектов, наиболее важным для подсистемы уведомлений является событие изменения (сохранения) доменного объекта. При этом, как правило, уведомление конкретного типа формируется при соблюдении следующих условий:

1. изменилось значение одного или несколько конкретных полей объекта;
2. значения некоторых полей этого или связанных объектов удовлетворяют определённым условиям (равно, не равно, больше, меньше, пустое, не пустое и т.п.; аналогичные проверки могут быть на количество связанных объектов того или иного типа).

## Содержание

Содержание уведомления зависит, в первую очередь, от произошедшего события, и определяется бизнес-логикой. Однако, в силу различия возможностей разных каналов доставки, оформление уведомления может быть своим для каждого канала. Например, сообщение e-mail должно содержать осмысленную тему (subject) и тело в формате HTML, содержащее, помимо прочего, ссылку, позволяющую открыть изменившийся документ в GUI системы для просмотра или редактирования, тогда как SMS-уведомление состоит из одной строки и должно быть достаточно кратким. Кроме того, в мультиязычных системах уведомления должны отправляться на языке, удобном для конкретного пользователя.

Чаще всего текст уведомления должен включать значения некоторых полей изменившегося (или каким-то иным образом связанного с произошедшим событием) доменного объекта. Этот текст формируется на основе шаблона, который может задаваться администратором.

Таким образом, отдельный шаблон должен быть задан для каждой возможной комбинации трёх элементов:

* тип уведомления;
* канал доставки;
* поддерживаемый язык системы.

Для упрощения администрирования желательно поддерживать вариант по умолчанию для каждого из этих элементов.

## Внешние адресаты

Основной пункт назначения уведомлений – это пользователь системы. Тем не менее, иногда возникает необходимость формировать уведомления для лиц, не являющихся её пользователями. По сути, эта задача почти не отличается от посылки уведомления пользователю системы, только объектом назначения является не пользователь системы, а некий другой доменный объект. Служба уведомлений должна поддерживать получение таких объектов в качестве адресатов, а каналы – уметь определять адрес назначения по этому объекту. Разумеется, не каждый канал может (и обязан) поддерживать внешних получателей.

# Компоненты подсистемы уведомлений

## Служба уведомлений

Служба уведомлений является одним из публичных сервисов (EJB) системы. Она принимает запросы на уведомление пользователей о каких-либо событиях. Параметрами запроса являются:

* пользователь, группа пользователей или контекстная роль – получатели уведомления;
* информация о событии, включающая его тип, категорию, важность, идентификатор документа(ов) и др.

В зависимости от содержания запроса, настроек пользователя и системной политики служба выбирает один или несколько каналов, которым передаёт запрос на выполнение уведомления.

Служба уведомлений обрабатывает запросы в асинхронном режиме.

## Каналы доставки

Канал доставки уведомлений – это программный код, выполняющий отправку сообщений некоторым способом. Несколько каналов доставки (изначально один – e-mail) входят в состав ядра системы, но модули могут добавлять собственные каналы.

Для внешних каналов доставки необходима дополнительная информация о пользователе – адрес (для e-mail; номер телефона для SMS и т.п.). Определение адреса по пользователю является одной из задач канала при отправке уведомления. Также канал должен позаботиться о добавлении в конфигурацию элементов, необходимых для хранения нужного ему адреса пользователей, или хотя бы проверить их наличие и корректность.

### E-mail

Канал доставки уведомлений по электронной почте входит в ядро системы. Он формирует сообщение e-mail с использованием шаблона сообщения и передаёт его внешнему почтовому серверу с использованием технологии Java Mail (код может использовать надстройку Spring Mail для работы с этим API). Шаблон выбирается по типу сообщения, а также языковым настройкам пользователя.

Данные, используемые каналом:

* ДО Person – e-mail пользователя.
* Профиль пользователя (ДО Person\_Profile?) – предпочитаемый язык (опционально).
* Конфигурация – шаблон, соответствующий типу сообщения.

### Генератор задач CMJ

Такой канал может быть разработан в составе модуля сопряжения платформы ECMp5 с Сочи-сервером. Он должен формировать объекты задач и помещать их в персональные коллекции данных соответствующих пользователей. Правила создания этих объектов и используемые при этом данные будут определены архитекторами проекта «Сочи».

### Агрегирующий e-mail

Альтернативный способ доставки уведомлений через электронную почту – не отдельными сообщениями, но накапливая данные за некоторый временной интервал. Этот канал доставки может входить в состав ядра системы или поставляться в виде отдельного дополнительного модуля.

Канал, принимая запросы на отправку уведомлений, не формирует сразу сообщения и не отсылает их, а сохраняет в специальных доменных объектах. Также в состав канала включается компонент, запускаемый через механизм периодических задач, и выполняющий формирование сообщений по накопленным запросам уведомлений, отсылку этих сообщений пользователям, а также удаление сохранённых запросов. Следует отметить, что этот канал доставки не может использовать те же шаблоны, что и канал «e-mail», т.к. в одном почтовом сообщении будет присутствовать информация не об одном, а о нескольких событиях, возможно, даже разнотипных.

## Компонент раскрытия шаблонов

Внутренняя служба платформы, предоставляющая функцию формирования текста по шаблонам. Основными клиентами службы являются каналы доставки, которым необходимо сформировать «красивое» сообщение, содержащее различные данные о произошедшем событии.

Шаблон сообщения – это текст, оформленный по тем же правилам, которым должно удовлетворять сформированное сообщение (HTML, XML, CSV, простой текст), но содержащий в себе определённым образом оформленные ссылки на данные уведомления или поля доменных объектов, указанных в этих данных. Поддержка нетекстовых (бинарных) форматов сообщений пока не планируется.

Необходимо определить язык задания шаблонов. В качестве вариантов можно рассмотреть Apache Velocity и FreeMarker. Крайне желательна также поддержка DOEL-выражений.

### Формирование ссылок

Важным элементом уведомлений, в особенности, получаемых пользователем на устройствах, с которых возможен доступ к системе, являются ссылки, позволяющие открыть нужный документ в системе. Компонент раскрытия шаблонов обязан уметь создавать такие ссылки.

Как правило, для создания ссылки нужен идентификатор документа и некоторая информация, зависящая от экземпляра системы (доменное имя сервера, контекст приложения). В некоторых случаях может потребоваться также какая-то специфичная для пользователя информация (в распределённой системе – имя его «домашнего» сервера, например).

Ещё один аспект, который должен быть учтён, – перспектива наличия нескольких разных клиентов у системы, для каждого из которых требуется отдельная ссылка. При этом желательно дать пользователю возможность выбирать клиенты, ссылки на которые он хочет получать.

## Генератор уведомлений

Генератор уведомлений предоставляет возможность настройки формирования уведомлений через конфигурацию. Он состоит из 2 компонентов, похожих по назначению, но различающихся по способу использования.

Каждый из этих компонентов является клиентом службы уведомлений.

### Уведомления по событиям

Компонент генерации уведомлений по событиям активируется через механизм точек расширения при происхождении различных внутрисистемных событий, в первую очередь, изменениях доменных объектов.

Компонент управляется тегами <notification> в конфигурации. Каждый такой тег содержит 2 группы настроек:

* Фильтр событий: тип доменного объекта; список полей, изменение которых требует отправки уведомления; ограничения на значения полей этого или связанных объектов. Возможно также подключение дополнительного кода для сложной фильтрации событий.
* Правила формирования запроса в службу уведомлений: получатель(и); канал(ы) отправки; тип, категория и другие параметры уведомления.

При отправке уведомлений по событиям следует учитывать транзакционный характер событий в системе. Отправка уведомлений должна производиться только в случае успешного завершения транзакции.

### Напоминания

Компонент генерации уведомлений-напоминаний работает не в ответ на какое-то событие в системе, а периодически. Он осуществляет выборку доменных объектов, по которым должны быть сгенерированы уведомления, и производит рассылку уведомлений.

Информация, необходимая для генерации напоминания одного типа, помещается в тег <reminder>:

* запрос на выборку доменных объектов (коллекция);
* параметры, передаваемые в службу рассылки уведомлений – аналогично генератору уведомлений по событиям.

Существенным отличием напоминаний от уведомлений по событиям является работа одновременно не с одним исходным доменным объектом, а с их коллекцией. Коллекции можно организовать по-разному: можно выбирать все объекты по некоторому бизнес-признаку (например, поле «Исполнить до» содержат текущую дату) и рассылать уведомления по каждому объекту в отдельности (формируя актуальный для него список получателей), а можно, организовав цикл по пользователям и использовав их id как параметр фильтра коллекции, выбирать интересующие объекты на персональной основе. Во втором случае легко сделать объединение нескольких напоминаний в одно уведомление. (Впрочем, этой же цели можно достичь и в первом случае, использовав канал доставки «агрегирующий e-mail».)

С точки зрения взаимодействия со службой периодических заданий можно рассмотреть 2 варианта реализации генератора напоминаний:

* отдельная задача на каждый тип уведомлений;
* единственная задача – диспетчер.

В первом варианте при старте системы после загрузки конфигурации некий код (менеджер) обеспечивает проверку наличия или создание объектов задач для каждого тега <reminder> в конфигурации. Он транслирует параметры настройки времени выполнения задачи из конфигурации в поля этих объектов, а часть XML с настройками параметров уведомлений сохраняет в качестве конфигурации (настройки) этой задачи. Код же самой задачи, запускаемый под управлением службы периодических заданий, выполняет работу по выборке объектов и генерации уведомления.

Второй вариант предполагает наличие единственной задачи-диспетчера, вызывающейся достаточно часто (например, каждые 10 минут). При каждом запуске эта задача перебирает все теги <reminder>, отбирает по ним объекты и формирует уведомления.

Преимуществом первого варианта является возможность гибкой настройки графика формирования уведомлений для каждого типа напоминаний и даже конкретного пользователя. Кроме того, реализация этого варианта представляется более простой. Однако, большое количество периодических задач создаёт трудности при их администрировании, а их одновременный запуск может приводить к падению производительности системы. Если же идти вторым путём, то для реализации задачи-диспетчера придётся создать механизмы, похожие на те, что имеются в подсистеме периодических задач: задание и обработка расписания и учёт обработанных документов (по которым отправлены уведомления). Создание единственной задачи при этом обеспечит сама подсистема периодических задач, а генерация напоминаний – задача, как правило, низкоприоритетная – будет выполняться одним фоновым потоком. Кроме того, в рамках одной задачи можно организовать объединение разных напоминаний для одного пользователя в одном сообщении.

## BPMN task

BPMN (Activiti) содержит стандартную задачу, выполняющую отправку e-mail – наиболее распространённый способ уведомления пользователей. Однако, для подсистемы уведомлений платформы e-mail – лишь один из способов отправки уведомлений. Кроме того, при использовании этой задачи код процесса должен сам сформировать текст сообщения, определить адрес пользователя, проверить его настройки по получению уведомлений данного типа и т.п. Поэтому необходима функциональная замена данной задачи (оформленная, видимо, как service task), осуществляющая отсылку уведомлений с использованием службы уведомлений платформы. Параметры этой задачи будут совпадать с параметрами метода службы. Инфраструктура бизнес-процессов позволит вычислять значения этих параметров из полей процесса или связанного с ним документа.