

Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование
информационных систем

Кафедра системного программирования

Васильев Алексей Дмитриевич

Разработка мобильных интерфейсов для
систем документооборота на основе
настраиваемых шаблонных сервисов

Магистерская диссертация

Научный руководитель:
д. ф.-м. н., профессор Терехов А. Н.

Рецензент:
Директор ООО «НМТ – Новые Мобильные Технологии» Оносовский В. В.

Санкт-Петербург
2017

SAINT-PETERSBURG STATE UNIVERSITY

Software and Administration of Information Systems
Sub-Department of Software Engineering

Aleksei Vasilev

Development of mobile interfaces for
workflow systems based on customised
template services

Master's Thesis

Scientific supervisor:
Professor Andrey Terekhov

Reviewer:
CEO New Mobile Technologies LLC Valentin Onosovsky

Saint-Petersburg
2017

Оглавление

Введение	5
1. Постановка задачи	8
2. Литературный обзор	9
2.1. Существующие аналоги	9
2.2. UbiqMobile	11
2.3. Контроллеры UbiqMobile	12
2.4. Microsoft SharePoint API	13
2.5. Redmine API	13
2.6. Vanessa Sharp Library	14
3. Сравнительный анализ реализации workflow-компонентов в различных системах документооборота	15
3.1. Microsoft Sharepoint	16
3.2. Redmine	17
3.3. 1С: Документооборот	17
3.4. EMC Documentum	18
3.5. Oracle Workflow	18
4. Обобщенная логическая модель функциональности workflow-компонентов	20
4.1. Обобщение модели workflow-компонентов	22
4.2. Проблемы обобщенной модели workflow-компонентов	22
4.3. Концепция настраиваемых шаблонов	24
4.4. Конфигурирование шаблона	25
5. Архитектура мобильного интерфейса на основе настраиваемых шаблонов	26
5.1. Архитектура модели	27
5.2. Архитектура контроллера	29

6. Особенности реализации прототипа мобильного интерфейса на основе настраиваемых шаблонов	32
Заключение	34
Список литературы	35

Введение

Информация является одним из наиболее ценных активов любой компании. Она может накапливаться годами и представляется в виде большого объема неструктурированных документов. По мере роста компании в ней увеличивается объем информационных потоков. Возникает необходимость обмена информацией между подразделениями, совместного редактирования документов, разграничения прав доступа и т.д. Для решения описанных выше проблем используются ЕСМ-системы (Enterprise Content Management), иногда называемые системами документооборота. ЕСМ – это набор практик, процессов и методологий, которые превращают технологию в наиболее эффективный способ хранения, защиты и использования контента [21]. ЕСМ-системы упрощают оборот документов, упорядочивают потоки информации и помогают организовать полный жизненный цикл слабо структурированной информации.

Системы документооборота можно условно разделить на две категории - специализированные и общего назначения. Специализированные системы, как правило, применяются в средних и малых предприятиях. Зачастую они предназначаются для использования в компаниях, относящихся к какой-то одной конкретной индустрии. Примером такой специализированной системы может служить распространенная система Redmine [14]. Эта система применяется в небольших и средних компаниях, специализирующихся на разработке программного обеспечения. Системы общего назначения имеют широкий профиль применения и, как правило, хорошо масштабируются. К этому типу систем можно отнести, в частности, Microsoft SharePoint [6], EMC Documentum [9], Oracle Workflow [7] и 1С:Документооборот [23].

Во всех этих системах корпоративный контекст представлен в форме внутреннего портала. Устроенный таким образом центр внутренней документации компании представляет собой коллекцию сайтов. Такая коллекция может быть наполнена различным контентом и конфигурироваться для каждой компании по-своему, даже в рамках одной систе-

мы документооборота. «Мобильный» доступ к контенту часто также осуществляется через web-интерфейс. Зачастую такие web-решения теряют нативность – то, что кажется простым и удобным для десктопной версии, может оказаться непригодным для использования на планшете, не говоря уже об экране смартфона.

В настоящее время все более актуальной задачей становится работа с системами документооборота через полноценные мобильные приложения, работающие на устройствах пользователей. Разработка таких приложений значительно дороже разработки браузерных решений, так как они в гораздо большей степени зависят от платформы и особенностей ЕСМ-системы.

Все задачи, решаемые ЕСМ-системами, направлены на поддержание полного жизненного цикла информации. Условно их можно разделить на пять категорий, представленных ниже

1. Управление документами.
2. Управление записями.
3. Управление корпоративными метаданными.
4. Управление веб-контентом.
5. Управление потоками работ.

Первые четыре категории являются административными функциями, которые требуют большого объема работ: управление персоналом, внесения изменений в документацию, изменение сайтов компании и т.д. Пятый компонент – управление потоками работ - отвечает за поддержку бизнес-процессов, передачу контента по маршрутам, назначение рабочих задач и состояний [24]. Эти задачи являются наиболее актуальными для менеджеров компаний, проводящих много времени вне офиса. По этой причине, для обеспечения «мобилизации» ЕСМ-системы, необходимо реализовать управление потоками работ (workflow).

Функциональность workflow-компонент близка во всех ЕСМ-системах. Однако детали конкретной реализации – списки используемых сущно-

стей, возможные статусы документов и задач, роли пользователей системы – различны не только для разных платформ, но и для отдельных реализаций конкретных систем документооборота на базе одной платформы. В результате мобильный интерфейс для каждого конкретного внедрения системы документооборота должно быть своим, работающим не только с конкретной ЕСМ-системой, но и с конкретной сборкой.

Как следствие, разработка подобных мобильных интерфейсов требует значительных временных затрат и денежных средств. Соответственно, методы, позволяющие реализовать типовое решение, которое легко конфигурируется и интегрируется для конкретного заказчика, приобретают особую ценность. Такой интерфейс почти не теряет функциональности, по сравнению с интерфейсом workflow-компонента созданным для конкретной ЕСМ-системы, но может быть интегрирован с любой из ЕСМ-систем. Такой интерфейс, реализующий стандартную функциональность, актуальную для всех систем, может быть разработан один раз для целого семейства систем. Его настройка на конкретную систему не потребует участия разработчиков и может выполняться силами команды по внедрению продукта. Это позволит экономить средства при разработке таких интерфейсов. Применение такого шаблонного подхода к построению мобильных интерфейсов возможно во многих областях, где имеется несколько реализаций одного жестко стандартизированного процесса.

1. Постановка задачи

Целью данной работы является разработка мобильного интерфейса для систем документооборота на основе настраиваемых шаблонных сервисов. Для выполнения цели данной работы были поставлены следующие задачи.

1. Провести сравнительный анализ информации о workflow-компонентах в существующих системах документооборота.
2. Предложить типовую модель мобильных интерфейсов для систем документооборота.
3. Спроектировать архитектуру программного обеспечения, реализующего предложенные решения.
4. Разработан прототип мобильного интерфейса, реализующий спроектированную архитектуру.

2. Литературный обзор

Данный литературный обзор можно разделить на две части. В первой из них рассказывается о существующих мобильных интерфейсах для упомянутых ранее ЕСМ-систем. Во второй части описываются технологии, применяемые для реализации прототипа мобильного интерфейса.

2.1. Существующие аналоги

Описанные в данной главе мобильные интерфейсы лишь частично можно назвать аналогами разрабатываемого в данной работе приложения. Их функциональное назначение для конечного пользователя – мобильный доступ к ЕСМ-системе, – совпадает с назначением разрабатываемого интерфейса. Однако техническая идея – создание одного мобильного интерфейса, который позволил бы получить доступ к нескольким ЕСМ-системам с помощью простой настройки параметров, не реализуется ни в одном из описанных далее мобильных интерфейсов.

В качестве аналогов будут рассмотрены три мобильных приложения, обеспечивающие доступ к различным ЕСМ-системам, это Microsoft SharePoint для мобильных устройств, 1С:Документооборот для мобильных устройств и RedminePM. Все три приложения рассматриваются в реализации для платформы Android [3, 11, 19], хотя также существуют версии для iOS [5, 13, 20], а приложения для SharePoint и 1С:Документооборот имеют версии для платформы windows phone [4, 12].

Microsoft SharePoint mobile application предоставляет доступ к группе сайтов на основе корпоративного или учебной учетной записи. Доступ может быть осуществлен с помощью почтового аккаунта и интегрируется с таким продуктом Microsoft как Office365. Первая версия приложения появилась в 2016 году с целью сделать доступ к Microsoft SharePoint сайтам более нативным и защищенным по сравнению с доступом через веб-браузер. Данный мобильный интерфейс реализован следующим образом. Пользователю доступен список дочерних сайтов Microsoft SharePoint. Для каждого из этих подсайтов доступны следу-

ющие функции: отобразить все файлы пользователя, привязанные к сайту, отобразить контакты пользователя, отобразить недавние активности, отобразить все списки доступные пользователю на этом сайте.

Мобильное приложение для системы 1С:Документооборот интегрирует в себе функционал почтового клиента, расписания и интерфейса для управления workflow. Workflow-компонент данной системы позволяет увидеть все задачи доступные пользователю. Они разделены на три части: «Задачи мне», «Поручено мной», «На контроле». «Задачи мне» позволяет увидеть все задачи, где текущий пользователь системы является исполнителем задачи. В списке «Поручено мной» отображаются задачи, созданные пользователем. Список «На контроле» отображает задачи, где пользователь, не являясь создателем задачи, выступает в качестве наблюдателя. Сама сущность – «задача» позволяет увидеть ее описание и список пользователей, задействованных в ней. Для задач из списка «Задачи мне» и «Поручено мной» существует возможность изменения параметров задачи. В качестве недостатка данного интерфейса можно указать отсутствие возможности управления документами.

Мобильный клиент RedminePM строится на основе управления двумя списками. В первом из них отображаются доступные пользователю проекты, а во втором - все задачи исполнителем в которых является пользователь. Проекты отображают для пользователя списки всех задач независимо от их типа, исполнителя или создателя. Достаточно чтобы пользователь являлся участником в этом проекте. При управлении отдельными задачами пользователь имеет возможность изменять данные о состоянии задачи, прикреплять файлы, а также добавлять связи с другими задачами.

Наиболее корректным сравнением может считаться KWizCom mobile[2] - агрегатор данных для Microsoft SharePoint. Сервис KWizCom позволяет строить таблицы и формы на основе заданных параметров, выводя в них заданные списки из SharePoint. В сервисе существует возможность задания параметров, используемых при построении форм таких как тип списка, свойства которые необходимо извлечь из SharePoint,

а также различные фильтры. Мобильный клиент позволяет агрегировать данные из уже настроенных в веб-приложении таблиц. К сожалению, данные настройки нельзя производить только для мобильного клиента, они требуют наличия веб-приложения. В остальном функционал данной системы близок к приложению, выпущенному Microsoft, т.е. он остается довольно громоздким.

2.2. UbiqMobile

UbiqMobile – это платформа, которая в течении нескольких лет разрабатывалась на Математико-механическом факультете СПбГУ. Её основным назначением является быстрая и эффективная разработка мобильных интерфейсов в различных бизнес-приложениях, обеспечивающих нативный пользовательский интерфейс.

Основной особенностью данной технологии является возможность хранить и гибко управлять бизнес-логикой, графическими интерфейсами и данными приложений со стороны сервера. На мобильных устройствах пользователей работают тонкие клиенты, получающие данные от сервера и отображающие графические элементы интерфейса на экран устройства; при этом данные и элементы интерфейса кэшируются в локальной памяти устройства. Это в чем-то делает клиент UbiqMobile похожим на браузер, но для конечного пользователя он выглядит как обычное «нативное» мобильное приложение. Достоинством такой терминальной архитектуры является сочетание низкой нагрузки на сеть и высокой интерактивности. Технология является кросс-платформенной: разработчик приложения при таком подходе программирует пользовательские интерфейсы в терминах единого графического API, ничего не зная о специфике конкретного мобильного устройства. Отображение на нативные элементы конкретной мобильной платформы происходит средствами UbiqMobile.

Платформа UbiqMobile обеспечивает разработчикам приложений следующий набор возможностей:

1. Высокая интерактивность разрабатываемых приложений;

2. Широкий набор доступных функций пользовательского интерфейса;
3. Эффективная оптимизация трафика;
4. Устойчивость пользовательских сессий;
5. Работа с различными типами мобильных устройств. [15]

Основным ограничением платформы UbiqMobile, прямо вытекающим из ее «терминальной» архитектуры, является ограниченная возможность автономной работы приложений в режиме офф-лайн. На данном этапе развития технологии эта проблема решается на уровне кэширования загруженных в мобильное устройство пользовательских интерфейсов и задания оффлайн-логики при помощи механизма триггеров. Такой подход позволяет обеспечить возможности оффлайна в объеме, достаточном для большинства бизнес-приложений. Решение данной проблемы активно ищется на данном этапе развития технологии.

2.3. Контроллеры UbiqMobile

Построение приложений на основе UbiqMobile контроллеров – это архитектурная методология, используемая UbiqMobile начиная с 2015 года. Данный подход напоминает конечный автомат, где каждый UbiqMobile контроллер, в свою очередь, также является конечным автоматом.

Каждый Ubiq Mobile контроллер реализует законченный фрагмент бизнес логики - со своими внутренними состояниями, экранами и т.д. Для каждого контроллера определяются: имя, начальное состояние, условия входа в контроллер и выхода из него, некоторое действие, которое он может выполнять по таймеру.

Внутри себя UbiqMobile контроллер содержит несколько обработчиков (handlers), представляющих собой отдельные состояния. Так, например, для UbiqTaskControler есть handler для отображения всего списка задач, есть обработчик для отображения активных задач, а есть

handler для отображения отдельной задачи. Таким образом, внутри каждого контроллера инкапсулирована вся необходимая функциональность для получения данных из модели и их обработки.

2.4. Microsoft SharePoint API

Одним из основных преимуществ платформы .NET является большое количество уже реализованных библиотек для взаимодействия с различными внешними системами. И, конечно же, Microsoft не могли не реализовать API для своей системы документооборота. SharePoint API, как и любые другие пакеты для .NET Framework, можно получить с помощью механизма NuGet пакетов.

Данное API строится на основе REST-сервиса, который совместим с моделью SharePoint. Сборка Microsoft.SharePoint реализует пространство имен Microsoft.SharePoint.ApplicationRuntime. В этом пространстве имен содержится набор классов, интегрированных с ASP.NET. Они содержат методы для обработки, запроса, аутентификации и обработки ошибок, позволяющие управлять данными из SharePoint [17].

Также API содержит множество инструментов для управления серверной частью. Главный из этих инструментов - SharePoint Server Object Model. Он позволяет проводить интеграцию с SharePoint через большое количество библиотек и классов [17].

2.5. Redmine API

Redmine API, так же как и Microsoft SharePoint API, строится на основе REST-сервиса. Данное API распространяется в виде бесплатной сторонней библиотеки для .NET с открытым кодом. Сборка этой библиотеки также может быть получена в виде NuGet-пакета [16].

В отличие от Microsoft SharePoint API, в котором получение данных реализовано в виде возможности отправки запросов к серверу, Redmine API предоставляет пользователю объектный интерфейс, т.е. получение данных инкапсулировано в запросы внутри API. Пользователь же

получает объекты и функции через которые может манипулировать данными.

Очевидным плюсом такого подхода является удобство, простота и изящество получаемого кода, свойственная объектно-ориентированному подходу. Однако к минусам можно отнести отсутствие доступа к некоторым видам сущностей Readmine.

2.6. Vanessa Sharp Library

Vanessa Sharp – библиотека классов для платформы Microsoft.NET, обеспечивающая получение данных платформы 1С. Данная библиотека относится к открытому программному обеспечению. В этой библиотеке реализуется ряд стандартных интерфейсов ADO.NET и LINQ, что упрощает доступ к данным. В отличие от других способов интеграции .NET Framework с платформой 1С, библиотека Vanessa Sharp не требует реализации вспомогательного кода, а сразу предоставляет объектную модель для доступа к данным.

Библиотека Vanessa Sharp построена на основе COM-технологии. Эта технология активно использует технику позднего связывания кода, что в свою очередь выразилось в реализации библиотеки на основе Dynamic Language Runtime - DLR. DLR - это среда исполнения, которая добавляет набор служб для динамических языков в Common Language Runtime и облегчает динамическую разработку на платформе .NET. Использование DLR технологии налагает дополнительные ограничения в виде требования к .NET Framework не ниже 4.0.

Ещё одним ограничением, налагаемым использованием COM-объектов, является требование к операционной системе. COM-объекты, являющиеся разработкой Microsoft, поддерживаются только Windows – платформами[18].

3. Сравнительный анализ реализации workflow-компонентов в различных системах документооборота

Для первоначального рассмотрения в данной работе были выбраны следующие системы документооборота: EMC Documentum [8], Microsoft SharePoint [1], Oracle Workflow. Исследование, проведенное компанией Gartner в 2015 году [10] показывает, что данные системы входят в число наиболее распространенных и перспективных. Традиционно компания Gartner отражает результаты в виде «Магического квадрата» (Magic Quadrant). В этом квадрате наиболее престижной является правая-верхняя четверть. В нее попадают компании «лидеры», обладающие «способностями к реализации» и «полнотой видения». «Магический квадрат», отражающий итоги исследования компании Gartner в области ЕСМ-систем, представлен на рис. 1.

В рамках данной работы также интерес представляет система 1С: Документооборот [23], являющаяся одной из самых популярных workflow-систем в России. Также большой интерес со стороны сообщества разработчиков программного обеспечения может вызвать мобильный интерфейс Redmine - приложения управления проектами и задачами [22], которое распространяется по GNU General Public License.

Целью данной главы является описание workflow-компонентов в указанных выше ЕСМ-системах, выявление общих сущностей, процессов и свойств. На основе такого анализа можно будет выделить общий набор сущностей и связей между ними, который может послужить основой для построения внутренней логической модели разрабатываемого сервиса. Интеграция сервиса с конкретной системой документооборота будет заключаться в построении отображения внутренней модели на логическую модель «целевой» workflow-системы.



Рис. 1: «Магический Квадрат», составленный компанией Gartner в 2015

3.1. Microsoft Sharepoint

Логическая модель MS SharePoint представляет собой коллекцию сайтов. Каждая из коллекций состоит, как минимум, из одного корневого сайта и произвольного количества подсайтов. Все они связаны единым URL-именем корневого сайта. Каждая коллекция напрямую связана с веб-сервером IIS и базой данных MS SQL, хранящей весь контент.

Информация в SharePoint представляется в виде списков. Каждый список хранит информацию определенного типа. Списки позволяют хранить текстовые документы, задачи, мультимедийные файлы, изображения, календари.

К элементам workflow в SharePoint относятся объекты типа task (задача) и document (документ). Для каждой задачи задан ряд свойств (properties): автор, описание, статус и ряд дополнительных парамет-

ров. Документы имеют схожие логические объекты. При этом реализация этих объектов отличается. Так, например, перевод задачи из состояния в состояние похож на процесс изменения статуса документа. Основным структурным отличием документов от задач является возможность прикрепления к ним различных типов файлов и отсутствие комментариев.

3.2. Redmine

Redmine является открытым серверным веб-приложением для управления проектами. С точки зрения логической модели верхнего уровня, система представляет собой сайт. Сайт содержит несколько разделов, каждый из которых представляет списки различных типов.

Redmine позволяет поддерживать ведение сразу нескольких проектов, учитывать временные затраты, использовать роли. Развертывание системы возможно на основе различных СУБД - MySQL, MS SQL, PostgreSQL, Oracle и т.д.

В рамках workflow можно выделить два типа объектов – issue и document. Issue являются центральным элементом системы. Данные сущности служат для описания задач, ошибок или пользовательских запросов, которые направляются разработчику. Они описывают задачу, ее автора, описание. Каждая из них имеет статус и приоритет. Особенностью Redmine является возможность привязки файлов как к сущности document так и к сущности issue.

3.3. 1С: Документооборот

1С:Документооборот, являющийся частью 1С:Предприятия, представляет собой пакет приложений, разворачиваемых на сервере. После этого доступ к нему может осуществляться с помощью десктопного тонкого клиента или web-интерфейса.

Workflow- компонент представляется здесь процессами и документами. Сущность «документ» прямо соответствует сущности документа из

MS SharePoint и обладает такими свойствами как заголовок, описание, автор, вид, прикрепленные файлы, состояние и т.д.

Сущность «процесс» соответствует сущности «задача» и выполняет те же функции. Эта сущность имеет такие свойства как заголовок, автор, исполнитель, описание, статус и ряд дополнительных параметров.

Сущность «документ» определяется автором, заголовком, описанием, а также позволяет прикреплять к себе файлы.

3.4. EMC Documentum

EMC Documentum - это комплексный набор программных продуктов, которые позволяют поддерживать определение и выполнение процессов. Кроме того, платформа позволяет управлять различными потоками информации, которая необходима для процессов.

Workflow в системе Documentum реализована на основе трех составляющих. Это хранение документов, управление очередями задач на основе

EMC Documentum Process Engine, а также приложение EMC Documentum TaskSpace, позволяющее управлять отдельными задачами.

Задачи и документы, которыми можно управлять с помощью EMC Documentum, содержат стандартный набор полей: автор, описание, приоритет, статус, приложенные файлы и т.д.

3.5. Oracle Workflow

Oracle Workflow – это набор программных продуктов, которые позволяют централизованно управлять документами. Как и другие ЕСМ-системы Oracle ЕСМ предоставляет web-сервисы для взаимодействия с сервером развертывания ЕСМ-системы. Основное отличие данной ЕСМ-системы от остальных - это тесная интеграция с продуктами oracle для управления, хранения и защиты слабо структурированной информации.

В рамках workflow существует единственный объект – «документ». Данная сущность агрегирует в себе свойства, присущие документам

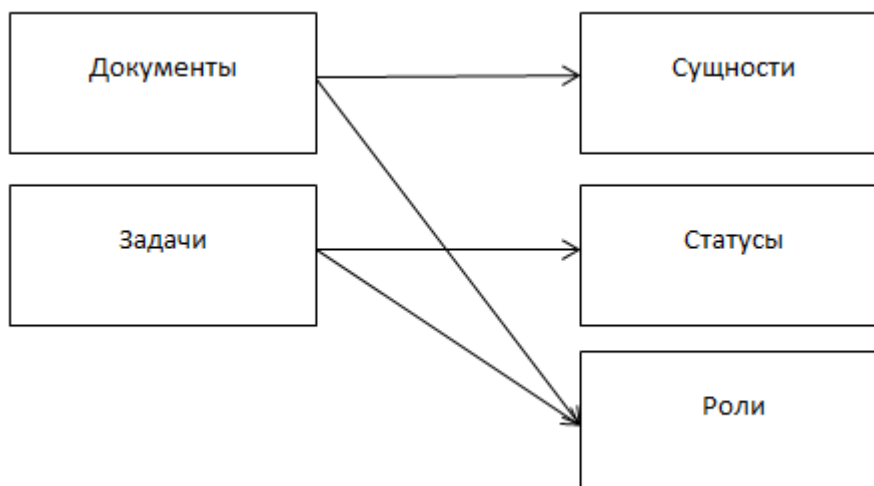
и задачам в других ЕСМ-системах. Существует несколько типов документов: входящие, исходящие и информационно-справочные. Входящие документы имеют набор свойств близкий к свойствам задач в других системах: автор, описание, приоритет, исполнитель, статус и т.д. В исходящих документах присутствуют такие свойства как автор, описание, адресат и прикрепленные файлы.

4. Обобщенная логическая модель функциональности workflow-компонентов

В разделе 3 был проведен сравнительный анализ workflow-компонентов пяти распространенных ЕСМ-систем. Как можно заметить из описания workflow-компонентов ЕСМ-систем, все они имеют схожие свойства. Это позволяет разработать общую концепцию настраиваемого интерфейса для ЕСМ-систем.

Однако детали реализации для задач и документов различны. Их можно определить с помощью следующих объектов: сущности, статусы, роли. Отношение этих объектов представлено на рис. 2.

Рис. 2: Соотношение логических объектов и их реализации



Реализация объектов из правой колонки индивидуальна для каждой конкретной системы и требует настройки. Различия в каждой реализации системы определяют проблему индивидуальной настройки пользовательского интерфейса.

По этой причине нельзя получить массовое и широко тиражируемое решение, удовлетворяющее потребности каждого пользователя – для каждой конкретной системы требуется настройка. Индивидуальная разработка приложений под каждого заказчика, решающих задачу доступа к конкретной коллекции сайтов, требует больших затрат времени и средств.

Поэтому предлагается рассматривать два уровня настройки мобильного интерфейса. Первый из них - настройка для одной из ЕСМ-систем (MS Share Point, EMC Documentum и т.д.). Второй - настройка на конкретную реализацию системы документооборота, развернутую у отдельно взятого заказчика. Таким образом, задача настройки мобильного интерфейса разделяется на две подзадачи.

Решение первой подзадачи видится в создании обобщенной логической модели, которая позволяет воспользоваться схожестью процессов управления workflow-компонентами во всех ЕСМ-системах. Такая модель будет выполнять роль своего рода онтологии для разрабатываемого настраиваемого сервиса, определяя систему базовых сущностей и взаимоотношений между ними, предоставляя общий интерфейс для получения доступа и управления данными. При интеграции с каждой конкретной системой документооборота потребуется реализация модулей-коннекторов, выполняющих отображение сущностей модели на объекты базовой ЕСМ-системы, добавляя реализацию модулей-коннекторов нижнего уровня на основе АРІ конкретной ЕСМ-системы. Понятно, что возможности интеграции при таком подходе не ограничиваются рассмотренными ранее системами документооборота – точно таким же образом можно интегрироваться и с новыми видами ЕСМ-систем, реализуя соответствующие коннекторы.

Решение второй подзадачи (настройка сервиса под конкретного заказчика) лежит в области использования шаблонов приложений, реализующих общую логику и допускающих настройку на каждого конкретного клиента. При этом желательно, чтобы настройка не требовала программирования, а выполнялась либо через конфигурационные файлы, либо через веб-интерфейс. Такое приложение-шаблон должно включать в себя как мобильную (клиентскую), так и серверную часть. Разработка таких шаблонных приложений на основе кросс-платформенных технологий позволяет уменьшить трудоемкость и сложность разработки. В рамках данной работы настраиваемое шаблонное приложение разрабатывается на основе технологии Ubiq Mobile, ориентированной на разработку кроссплатформенных клиент-серверных приложений для

бизнеса. Платформа Ubiq Mobile позволяет создавать настраиваемые мобильные сервисы, в которых можно «на лету» конфигурировать настройки приложения на основе предложенного шаблона.

4.1. Обобщение модели workflow-компонентов

Как видно из результата сравнительного анализа ЕСМ-систем, представленных в главе 3, во всех описанных системах сущности, отвечающие за workflow, близки друг другу. Однако эта близость не полная: не в каждой из этих систем есть задачи (tasks, issues) или документы (documents). Состав полей, свойств и атрибутов, за исключением самых основных (автор, описание и т.д.), также может заметно различаться в каждой конкретной системе документооборота. Как результат, функциональность некоторых систем может не полностью укладываться в предлагаемую модель.

Однако в целом объекты для построения workflow можно свести к двум типам сущностей: задачи (tasks) и документы (documents). На основе этих двух типов сущностей можно создать общую, высокоуровневую модель, которая позволит управлять функциональностью, связанной с workflow.

Также можно выделить ещё одну сущность, представляющую пользователя системы, который характеризуется своим логином и паролем. Назовем эту сущность мандатом (credential). Таким образом, общая (стандартизированная) модель может быть построена на основе трех типов сущностей: task, document, credential.

Итоговое представление обобщённой логической модели показано в таблице 1.

4.2. Проблемы обобщенной модели workflow-компонентов

Предлагаемая модель позволяет реализовать основные функциональные возможности workflow-компонентов различных систем документо-

оборота. К сожалению предложенное решение содержит ряд ограничений.

Самым очевидным ограничением является «отсечение» функциональных особенностей отдельных систем документооборота. К таким потерям можно отнести свойства, которые могут быть уникальны для конкретной системы документооборота. Решение этой проблемы видится в реализации концепции настраиваемых шаблонов. Эта концепция позволит ввести специфические – пользовательские свойства, присущие документам или задачам только в отдельных ЕСМ-системах. Подробнее о механизме настраиваемых шаблонов пойдет речь в следующей главе.

Выбор платформы UbiqMobile, в качестве платформы для разработки, определил ряд сложностей для процесса интеграции с ЕСМ системами. Данная платформа построена на основе фреймворка .NET, в связи с чем возникает проблема совместимости с API некоторых систем документооборота. Так например API 1С: Документооборота реализовано с помощью 1С. К счастью в данном случае возможно использование СОМ объектов, подробнее об этом будет рассказано в главе, посвященной особенностям реализации. В тоже время API Oracle традиционно реализуется на языке Java, и по этой причине не поддается столь простой интеграции с .NET. Для их совмещения потребуется «прослойка», реализованная к примеру на C++, что увеличивает сроки разработки. К сожалению, на момент написания данной работы эти затраты я считаю не целесообразными. По этой причине интеграция Oracle Workflow не была реализована в разработанном прототипе и не рассматривается далее.

В тоже время интеграция с Microsoft SharePoint была реализована на уровне C# API и не вызвала сложностей, а интеграция с Redmine была выполнена на уровне веб-сервисов, с которыми удобно работать из .NET.

4.3. Концепция настраиваемых шаблонов

Использование механизма шаблонов предполагает легкую замену одних данных на другие, а также простую настройку параметров (например, списка свойств). Исходные данные для настройки берутся из файла конфигурации.

Конфигурационный файл состоит из двух разделов. Первый раздел содержит настройки для конфигурирования пользовательского интерфейса приложения, а второй – настройки, определяющие тип используемой ЕСМ-системы и ее параметры. Он позволяет легко скрывать или добавлять новые параметры в зависимости от выбранной ЕСМ-системы или свойств конкретной реализации.

Пример записи в файле конфигурации показан на рис. 4.3. Каждая запись имеет вид «ключ – значение». Первая запись относится к настройкам данных из ЕСМ-системы, отображаемых интерфейсом. Она состоит из имени поля и значения, в котором содержатся данные о типе поля, его значении по умолчанию и атрибута, показывающего есть ли необходимость отображать текущее поле. Вторая запись относится к настройкам пользовательского интерфейса. В ней отображается имя поля и его значение указывающее на то, что в нем содержится информация о цвете в формате RGB, цвет и атрибут, показывающие активна ли данная настройка и типа поля.

```
<add key="Author" value ="FieldString:Not supported:true"/>  
<add key="TopLayoutColor" value ="RGB:0;206;103:true"/>
```

Рис. 3: Пример записей в файле конфигурации.

Данный подход хорошо сочетается с архитектурой мобильных приложений технологии UbiqMobile. Мобильные приложения UbiqMobile состоят из двух частей. Тонкий клиент находится на мобильном устройстве пользователя. Серверная часть содержит в себе всю бизнес логику. Такой подход позволяет мгновенно отображать все изменения, вносимые в файл конфигурации пользователем.

4.4. Конфигурирование шаблона

Для редактирования файла конфигурации возможно использовать веб-интерфейс. Он должен состоять из нескольких секций, соответствующих секциям файла конфигурации: UI, настройки ЕСМ-системы. Такой интерфейс может быть реализован на основе паттерна MVC технологии ASP.NET.

Архитектура такого решения может быть построена на основе контроллера управляющего файлом конфигурации. Контроллер позволит извлечь данные из файла конфигурации и заполнить объект модели. На основе этой модели строится веб-страница с настройками данной модели. После сохранения изменений контроллер перезапишет файл конфигурации, учитывая изменения, внесенные пользователем в модель. Благодаря использованию технологии UbiqMobile все изменения сразу будут получены пользователями.

Пример кастомизации интерфейса с помощью шаблона представлен на рис. 4.4. На рисунке показано изменение пользовательского интерфейса и списка отображаемых полей для одной из задач.

Изменения, происходящие при выборе ЕСМ-системы Redmine, вместо Microsoft SharePoint, используемой в прошлом примере, показаны на рис. 4.4. Список полей, как и настройки интерфейса, не изменился. При этом отсутствующие поля помечены как «Not supported», и пользователь при желании может скрыть их, изменив настройки шаблона в файле конфигурации.

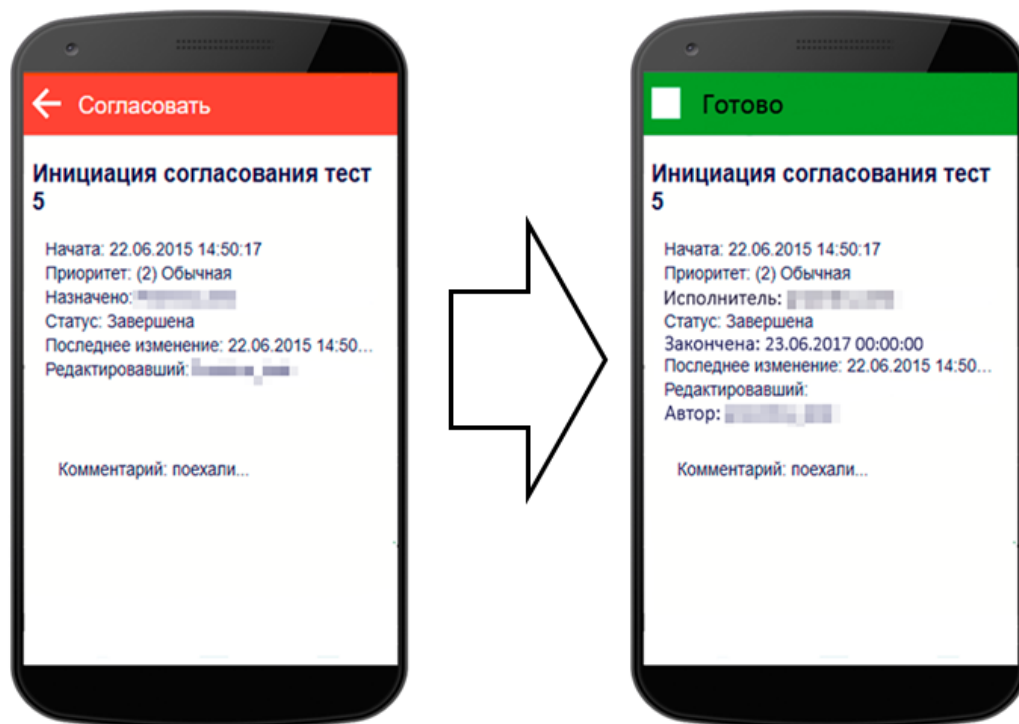


Рис. 4: Кастомизация приложения на примере «задачи» в ЕСМ-системе Microsoft SharePoint.

5. Архитектура мобильного интерфейса на основе настраиваемых шаблонов

Высокоуровневое представление разработанного приложения показано на рис. 5. Главное место в нем занимает сервер UbiqMobile. Сервер содержит несколько отдельных виртуальных защищенных сред - песочниц (SandBox). Каждая из них содержит все серверные компоненты и данные размещенных в ней приложений. Данные, получаемые из ЕСМ-системы, обрабатываются приложением, логика которого исполняется в одной из песочниц UbiqMobile сервера и отправляются на клиент пользователя - «User device».

Архитектура разработанного мобильного интерфейса состоит из двух крупных логических частей модели и контроллера. Модель является имплементацией обобщенной модели workflow-компонентов, описанной в предыдущих главах. Контроллер строится на основе технологии Ubiq mobile контроллеров, которая подробно описывается в литературном



Рис. 5: Отображение задачи при переходе с Microsoft SharePoint на Redmine.

обзоре. Модель и контроллер реализуют идею настраиваемых шаблонов, описанную в прошлой главе.

5.1. Архитектура модели

Предлагаемая модель представлена на рис. 7. В ней описаны интерфейсы на примере трех систем MS Sharepoint, 1С: Документооборот и Redmine. Она, условно, может быть разделена на три части. Первая – workflow model, реализует методы для извлечения данных из ЕСМ-системы. В ней реализуется ряд «надстроек» над API различных ЕСМ систем. Вторая часть – credential model, отвечает за хранение данных пользователей, требуемых для доступа к ЕСМ-системе. Третья часть – base objects, реализует контейнеры для хранения данных.

Модель workflow инкапсулирует детали логических различий между системами документооборота и технические различия в их API. Она

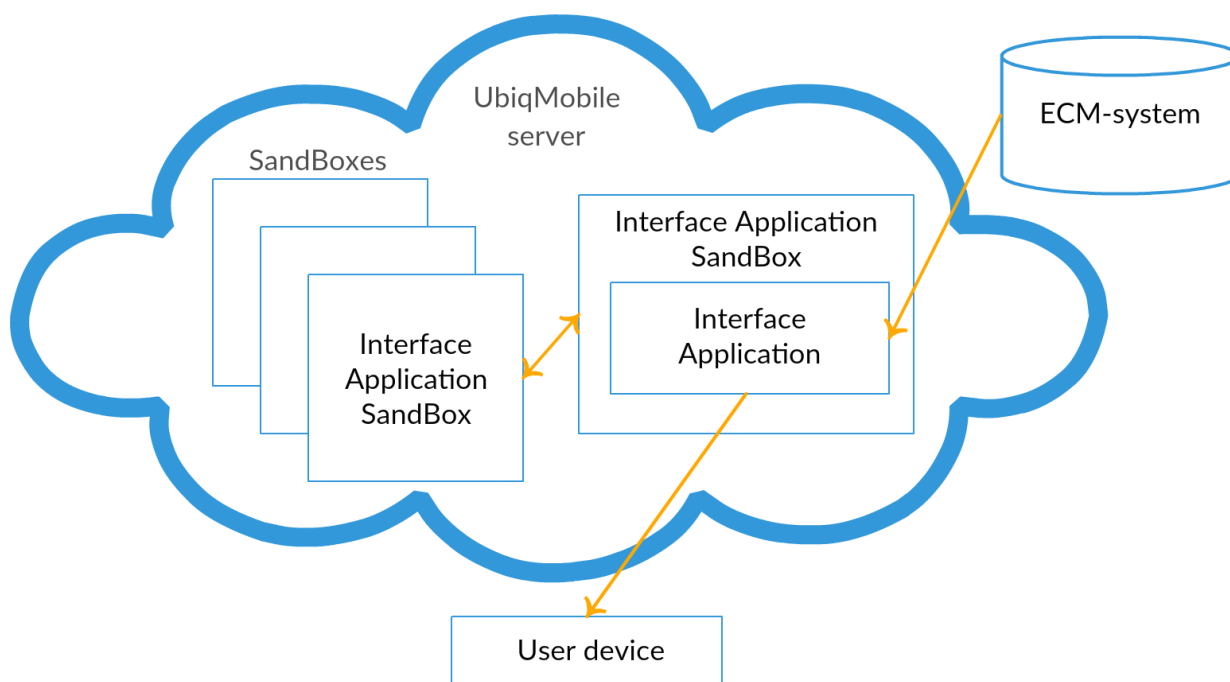


Рис. 6: Общая архитектура реализуемого мобильного интерфейса.

строится на основе общего интерфейса IModel. Этот интерфейс позволяет получить доступ к спискам задач и документов. На рис. 7 можно видеть реализации для Microsoft SharePoint (SPModel), 1С:Документооборот (1SModel) и Redmine (RMModel). Каждая модель позволяет получить доступ к конкретной системе документооборота. В каждой модели используется API для конкретной системы документооборота. Возвращаемые данные представляются в виде списков объектов, реализующих интерфейсы из Base model. Таким образом, интерфейс IModel обеспечивает унификацию получаемых данных. В рамках текущей реализации пользователь может получить список всех задач (которые видны пользователю), список задач пользователя (предназначенных ему), список активных задач, список документов (которые видны пользователю) и список документов пользователя (автором которых он является).

Base objects реализует три интерфейса, это IDocObject, ITaskObject и IFileObject. Типы, реализующие эти интерфейсы, предназначены для хранения объектов, задач и файлов соответственно. Для каждой системы документооборота реализуются свои типы элементов на основе этих трех интерфейсов. Каждая реализация этих интерфейсов имеет свои

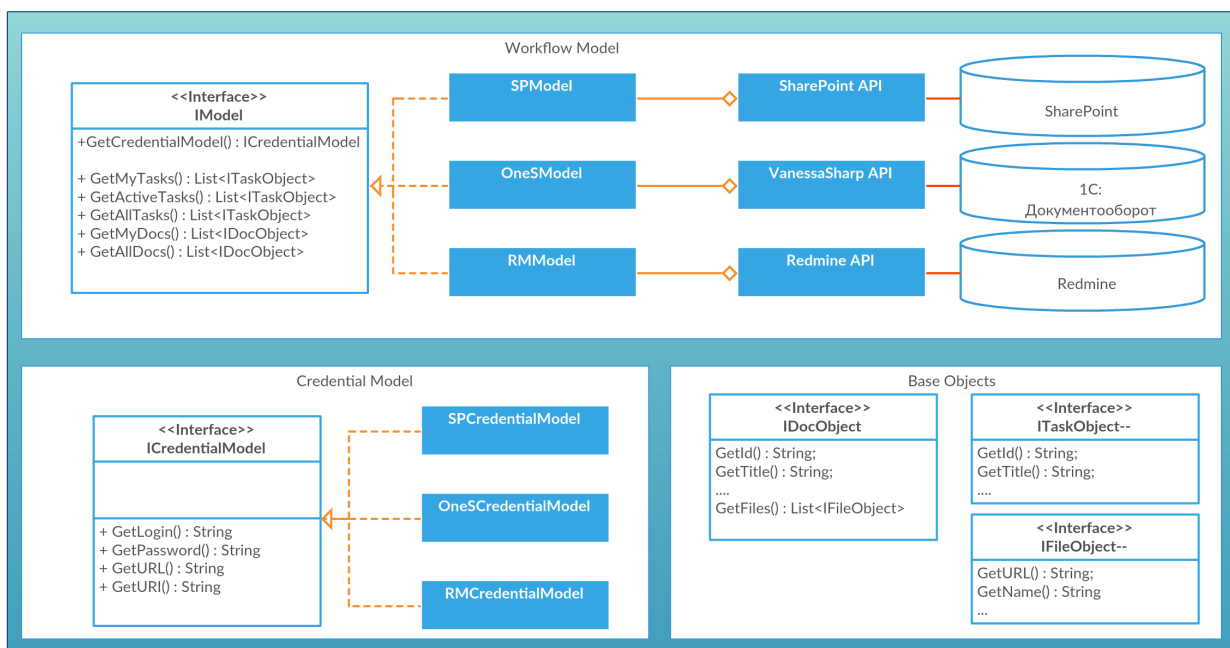


Рис. 7: Диаграмма модели

различия. Но реализуемый интерфейс гарантирует получение основных данных, описанных в обобщенной логической модели и присущих одинаковым по типу объектам, в каждой из систем документооборота. В частности, общими полями являются такие поля как заголовок, id, список прикрепленных файлов, создатель и т.д.

Credential модель представляется интерфейсом ICredentialModel. Реализации данной модели хранят данные, необходимые для доступа к конкретной системе документооборота. Объект данной модели передается в реализацию IModel. Затем полученные данные используются для авторизации в API конкретной системы документооборота. Общими для всех реализаций здесь являются login, password, URL.

5.2. Архитектура контроллера

Диаграмма контроллера представлена на рис. 8. Он состоит из нескольких Ubiq Mobile контроллеров.

При старте приложения запускается базовый UbiqController. Он с помощью AppConfigController'a получает данные из конфигурационного файла о системе документооборота, с которой ему предстоит рабо-

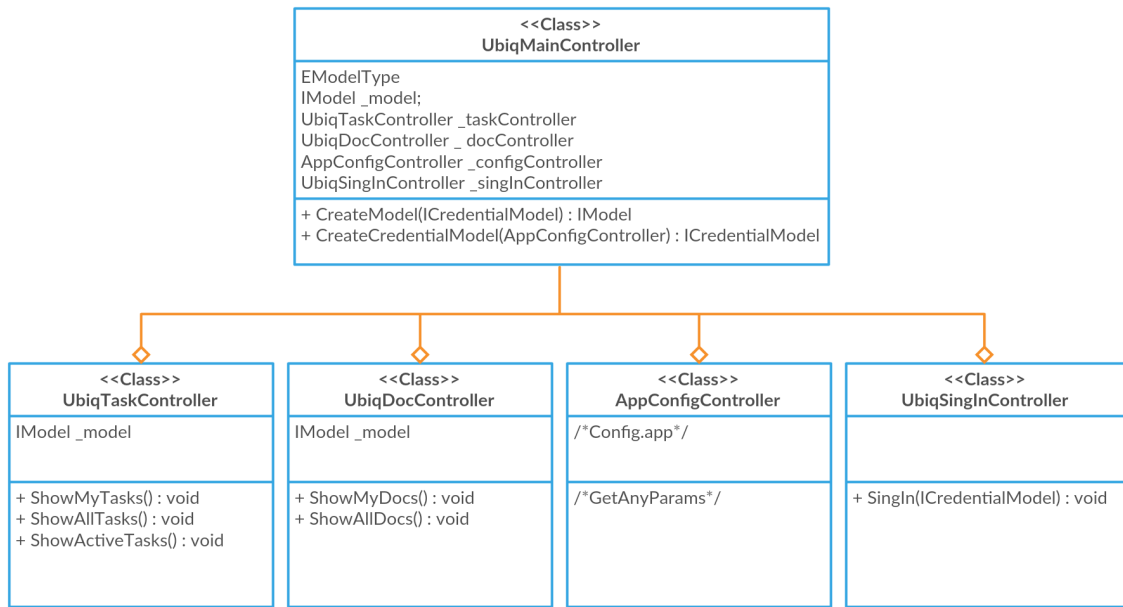


Рис. 8: Диаграмма контроллера

тать, и данные своего UI. На основе полученных данных определяется тип системы документооборота, и с помощью SimpleModelFactory создается модель для работы с ней.

Полученные от пользователя login и password передаются в UbiqSingInController при авторизации. Он на основе данных, полученных от AppConfigController и данных пользователя, генерирует объект ICredentialModel. В случае удачной аутентификации credential model передается в ранее созданную модель. Таким образом, пользователь получает доступ к задачам и документам. Он осуществляется с использованием UbiqTaskController и UbiqDocController. UbiqTaskController содержит обработчики, позволяющие отображать списки всех доступных пользователю задач, активных задач и задач, назначенных пользователю. В зависимости от выбранной ЕСМ-системы обработчик запрашивает данные из модели и отображает их для пользователя. Также существует обработчик для отображения наполнения отдельной задачи. Аналогичным образом UbiqDocController содержит обработчики, отвечающие за обработку списков всех доступных пользователю докумен-

тов и документов, назначенных пользователю. Отличием обработчика отображающего отдельные документы, от обработчика, отображающего отдельные задачи, является использование «ленивых» методов. Они используются для получения файлов прикрепленных к документу.

6. Особенности реализации прототипа мобильного интерфейса на основе настраиваемых шаблонов

В ходе реализации ранее описанной архитектуры возникло несколько проблем. К моменту написания данной работы большинство из них были успешно решены. Однако некоторые из них требуют дополнительного изучения отдельных особенностей ЕСМ-систем. Далее описываются некоторые технические решения, примененные при разработке.

Особенности реализации модели документов. При реализации документов возникло сразу несколько проблем. Основной из них стал выбор между асинхронной загрузкой файлов, прикрепленных к документам, сразу после начала работы приложения и отложенной – «ленивой» загрузкой.

У каждого из вариантов были очевидные плюсы. В случае «одноразовой» загрузки сбор файлов происходил сразу после включения. Это позволяло кешировать сразу все файлы. В результате пользователь мог получить доступ к необходимым ему документам, даже в случае потери интернет соединения. Ценой такого решения стала долгая начальная загрузка приложения и бесполезные траты интернет-трафика, в случае, если пользователю необходимы всего несколько файлов из сотен, привязанных к его документам.

Метод отложенной загрузки запрашивал файлы только при открытии документа. Данный подход лишен недостатков предыдущего. Но, в тоже время, в случае отсутствия интернет соединения, пользователь не сможет загрузить необходимый документ.

Для решения данной проблемы был проведен ряд экспериментов. В результате этих экспериментов было установлено, что для ЕСМ-системы Microsoft SharePoint, использование данного способа уже на 200 файлах приводит почти к 15 секундной задержке в начале работы приложения. Это объясняется «бутылочным горлышком», которое образуется на стыке API и реализации модели, и связано с ограничением количе-

ства потоков.

В результате для использования был выбран именно алгоритм «ленивой» загрузки файлов. Однако, возможно, в рамках параметризации приложения, выбор алгоритма загрузки также стоит позволить выбрать пользователю. Данное решение хорошо подходит к «идеологии» решения описываемого в данной работе.

Другой крупной проблемой стало отсутствие доступа к документам через стандартное API Redmine. В данный момент мы пытаемся найти решение этой задачи. Одним из рассматриваемых вариантов является использование REST API. (возможно далее про REST API)

Ещё одной крупной проблемой с документами стало использование Open Source библиотеки классов для доступа к данным в 1С – Vanessa Sharp. Данный фреймворк облегчает доступ к 1С данным путем поддержки стандартных интерфейсов ADO.NET и LINQ. В результате его использования значительно упростилось получение данных из базы 1С.

Однако, возникла проблема, связанная с невозможностью загрузки файлов в базу 1С. Для решения этой проблемы пришлось использовать механизм COM-компонентов. Данный механизм требует соблюдения ряда правил, которые осложняют работы с ним. Кроме того, COM компоненты активно используют механизм позднего связывания и .NET технология Reflection. Это делает использование COM-компонентов не типобезопасным. По этой причине данный механизм был использован только для загрузки файлов, в то время как остальная часть модели для 1С: Документооборота выполнена на основе API Vanessa Sharp.

Заключение

В рамках данной дипломной работы были получены следующие результаты:

1. Собрана и проанализирована информация об основных сущностях workflow, присутствующих в системах документооборота Microsoft SharePoint, Redmine, 1С:Документооборот, EMC Documentum и Oracle workflow.

2. Предложена типовая модель мобильных интерфейсов для систем документооборота с использованием настраиваемых шаблонов на основе платформы UbiqMobile.

3. Спроектирована архитектура мобильного интерфейса для систем документооборота на основе настраиваемых шаблонов.

4. Разработан прототип мобильного интерфейса, реализующий спроектированную архитектуру для Microsoft SharePoint, Redmine, 1С:Документооборот.

Список литературы

- [1] Cawood Stephen, Wallbridge Sean. Microsoft SharePoint 2013. — McGraw-Hill/Osborne, 2013. — P. 304.
- [2] Corporation KWizCom. KWizCom mobile. — 2017. — [Online; accessed 29-апрель-2017]. URL: <http://www.kwizcom.com/sharepoint-add-ons/sharepoint-mobile-extensions/overview/>.
- [3] Corporation Microsoft. Microsoft SharePoint для Android. — 2017. — [Online; accessed 29-апрель-2017]. URL: <https://play.google.com/store/apps/details?id=com.microsoft.sharepoint&hl=ru>.
- [4] Corporation Microsoft. Microsoft SharePoint для Windows Phone. — 2017. — [Online; accessed 29-апрель-2017]. URL: <https://www.microsoft.com/en-us/store/p/sharepoint/9nblggh510hb>.
- [5] Corporation Microsoft. Microsoft SharePoint для iOS. — 2017. — [Online; accessed 29-апрель-2017]. URL: <https://itunes.apple.com/ru/app/microsoft-sharepoint/id1091505266?mt=8>.
- [6] Corporation Microsoft. SharePoint. — 2017. — [Online; accessed 29-апрель-2017]. URL: <https://products.office.com/ru-ru/sharepoint>.
- [7] Corporation Oracle. Oracle Workflow Developer's Guide. — 2017. — [Online; accessed 29-апрель-2017]. URL: https://docs.oracle.com/cd/B19306_01/workflow.102/b15853/toc.htm.
- [8] EMC. EMC Documentum xCP для управления бизнес-процессами. Детальный обзор. — 2010. — [Online; accessed 7-апрель-2017]. URL: <https://russia.emc.com/collateral/software/white-papers/h3352-bpm-wp.pdf>.
- [9] EMC Dell. EMC Documentum. — 2017. — [Online; accessed 29-апрель-2017]. URL: <http://www.emc.com/enterprise-content-management/index.htm>.

- [10] Gartner. Magic Quadrant for Enterprise Content Management. — 2015. — [Online; accessed 7-апрель-2017]. URL: http://www.project-consult.de/files/Gartner_MQ_ECM_Enterprise_Content_Management_2015.pdf.
- [11] LLC 1C-SOFT. 1C:Документооборот для Android. — 2017. — [Online; accessed 29-апрель-2017]. URL: <https://play.google.com/store/apps/details?id=com.microsoft.sharepoint&hl=ru>.
- [12] LLC 1C-SOFT. 1C:Документооборот для Windows Phone. — 2017. — [Online; accessed 29-апрель-2017]. URL: <https://www.microsoft.com/ru-ru/store/p/1C-Документооборот/9nblggh68jps>.
- [13] LLC 1C-SOFT. 1C:Документооборот для iOS. — 2017. — [Online; accessed 29-апрель-2017]. URL: <https://itunes.apple.com/ru/app/1c-Документооборот/id660236705?mt=8>.
- [14] Lang Jean-Philippe. Redmine. — 2017. — [Online; accessed 29-апрель-2017]. URL: <http://www.redmine.org/>.
- [15] Onossovski Valentin, Terekhov A. Ubiq Mobile—a new universal platform for mobile online services // Proceedings of 6th seminar of FRUCT Program. — 2009. — P. 96–105.
- [16] Pavić Aleksandar. Redmine Cookbook. — Packt Publishing Ltd, 2016. — P. 274.
- [17] Pialorsi Paolo. Microsoft SharePoint 2013 Developer Reference. — Microsoft Press, 2013. — P. 794.
- [18] Project Vanessa Sharp. Vanessa Sharp Project. — 2016. — [Online; accessed 7-апрель-2017]. URL: <http://vanessa-sharp.ru/>.
- [19] Project mode Inc. RedminePM для Android. — 2017. — [Online; accessed 29-апрель-2017]. URL: <https://play.google.com/store/apps/details?id=jp.co.projectmode.redminepm>.

- [20] Project mode Inc. RedminePM для iOS. — 2017. — [Online; accessed 29-апрель-2017]. URL: <https://itunes.apple.com/us/app/redminepm-redmine-client-app/id631915897?mt=8>.
- [21] Riley Christopher, White Shadrach. Enterprise content management with Microsoft SharePoint. — Microsoft Press, 2013.
- [22] Wikipedia. Redmine — Wikipedia, The Free Encyclopedia. — 2016. — [Online; accessed 7-April-2017]. URL: <https://en.wikipedia.org/w/index.php?title=Redmine&oldid=747682808>.
- [23] «1С» ООО. 1С:Документооборот. — 2017. — [Online; accessed 7-апрель-2017]. URL: <http://v8.1c.ru/doc8/>.
- [24] Википедия. Управление корпоративным контентом — Википедия, свободная энциклопедия. — 2016. — [Online; accessed 7-апрель-2017]. URL: <http://ru.wikipedia.org/?oldid=80249896>.

Таблица 1: Сущности и основные свойства обобщенной логической модели.

Сущность	Свойство	Описание
Задача (Task)	Автор (Author)	Создатель задачи - ссылается на аккаунт
	Исполнитель (Executor)	Исполнитель задачи - ссылается на аккаунт
	Время создания (Creation time)	Время создания
	Время начала (Start time)	Время когда "Исполнитель" перевел задачу в активное состояние
	Время завершения (Finish time)	Время когда "Исполнитель" перевел в завершенное состояние
	Заголовок (Title)	Заголовок задачи
	Описание (Description)	Описание задачи
	Комментарии (Comments)	Комментарии оставленные пользователями
	Состояние (State)	Текущее состояние задачи
Документ (Document)	Автор (Author)	Создатель документа - ссылается на аккаунт
	Получатель (Recipient)	Получатель документа - ссылается на аккаунт
	Время создания (Creation time)	Время создания
	Заголовок (Title)	Заголовок документа
	Описание (Description)	Описание документа
	Комментарии (Comments)	Комментарии оставленные пользователям
	Файлы (Files)	Файлы приложенные к документу
Мандат (Credential)	Имя пользователя (User name)	
	Пароль (Password)	
	URL	URL ссылающийся на точку доступа к ЕСМ-системе