

Санкт-Петербургский государственный университет

Гуданова Варвара Сергеевна

Выпускная квалификационная работа

Интеграция ЕСМ в операционную систему Windows

Уровень образования: бакалавриат

Направление *02.03.03 «Математическое обеспечение и администрирование
информационных систем»*

Основная образовательная программа *СВ.5006.2017 «Математическое обеспечение и
администрирование информационных систем»*

Профиль *Системное программирование*

Научный руководитель:
к.ф.-м.н., доцент кафедры системного программирования Д. В. Луцев

Рецензент:
мастер программных технологий ЗАО «ДИДЖИТАЛ ДИЗАЙН» С. В. Березников

Санкт-Петербург
2021

Saint Petersburg State University

Varvara Gudanova

Bachelor's Thesis

Integration of ECM into Windows operating system

Education level: bachelor

Speciality *02.03.03 "Software and Administration of Information Systems"*

Programme *CB.5006.2017 "Software and Administration of Information Systems"*

Profile: *System Programming*

Scientific supervisor:
C.Sc., docent Dmitry Luciv

Reviewer:
Software Development Master "Digital Design" Sergey Bereznikov

Saint Petersburg
2021

Содержание

Введение	4
1. Постановка задачи	7
2. Требования	8
3. Обзор	9
3.1. Предметная область	9
3.1.1. Docsvision	9
3.1.2. Shell Namespace Extension	10
3.2. Обзор существующих решений	11
3.2.1. Клиентские приложения для работы с Docsvision ECM	11
3.2.2. Другие ECM-системы	12
3.2.3. Решения, интегрированные в проводник Windows	13
3.2.4. Вывод	13
4. Описание решения	14
4.1. Адаптация объектов Docsvision для расширения	14
4.2. Архитектура	15
4.2.1. Ограничения	15
4.2.2. Разработанная архитектура	16
4.2.3. Технологии для взаимодействия между компонентами	18
4.3. Особенности реализации: преобразование моделей данных	20
4.4. Проводник Windows как графическая оболочка для контента	
Docsvision ECM	21
5. Тестирование	23
Заключение	25
Список используемой литературы	26

Введение

В настоящее время в бизнес-компаниях производится работа с огромным количеством разнообразного контента. И это оказывается все более сложной задачей из-за увеличивающихся объемов информации, с которой необходимо работать.

ЕСМ-системы (Enterprise Content Management Systems) предназначены для управления корпоративным контентом, то есть для решения задач, связанных с управлением, хранением и интеграцией документов и другой неструктурированной информации компании. К примеру, они позволяют маршрутизировать документооборот — предоставляют возможность отправлять документы на согласование, подписание или отправлять задания на исполнение конкретным сотрудникам с указанием важных деталей. Таким образом, ЕСМ-система позволяет создать единое цифровое пространство в компании, позволяя прервать зависимость компании от документов на бумажных носителях, организовать контроль версий электронных документов и осуществить автоматизацию ряда других процессов компании. Все это увеличивает производительность бизнеса, уменьшает организационные риски и позволяет улучшить обслуживание клиентов.

В настоящее время существует большое количество ЕСМ-систем, которые обладают разными функциональными возможностями. Примерами являются Elma ECM¹, Docsvision ECM², Directum RX³. Однако для работы с большинством подобных систем необходимо наличие полноценного клиентского приложения на устройстве пользователя или веб-клиент для удаленного доступа. Данный подход имеет ряд недостатков.

1. Для использования клиентского приложения новым пользователям необходимо познакомиться с его интерфейсом, изучить возможности приложения для решения своих задач. Процесс обучения может занимать зна-

¹<https://www.elma-bpm.ru/product/ecm/>

²<https://docsvision.com/ecm-bpm/docsvision-ecm/>

³<https://www.directum.ru/products/directum>

чительное время, и это усугубляется тем фактом, что часто интерфейс приложения оказывается перегружен.

2. Обычно создание электронных документов происходит в сторонних приложениях, и только потом они загружаются в ЕСМ-систему. В таких случаях использование полноценного клиента может добавлять «лишние» шаги в пользовательские сценарии. К примеру, для добавления нового документа в систему может потребоваться следующая цепочка действий.

- 2.1 Создать документ в стороннем приложении (текстовом редакторе).

- 2.2 Сохранить документ на диск, запомнить его расположение в файловой системе.

- 2.3 Открыть клиентское приложение ЕСМ-системы.

- 2.4 Подготовить все необходимое для добавление нового документа (создать карточку, определить атрибуты и пр.).

- 2.5 Добавить документ в систему с помощью интерфейса клиентского приложения.

При этом обычно нет необходимости в том, чтобы созданный документ находился на локальном компьютере пользователя, следовательно, данный файл можно удалить после выполнения описанных действий.

В этой ситуации удобным видится интеграция ЕСМ-системы с операционной системой (в данном случае — Windows), а именно, с ее файловой системой. Предлагается добавить в Проводник Windows виртуальные папки и файлы и реализовать определенную функциональность для обработки взаимодействия пользователя с этими виртуальными элементами, что позволило бы легко выполнить необходимые сценарии взаимодействия с ЕСМ-системой.

Для пользователей, знакомых с Windows, разобраться с интерфейсом Проводника и правилами работы с виртуальными папками и файлами будет несложной задачей. Кроме того, такой подход избавит пользователей от лишних шагов, так как появится возможность сохранять созданные документы непосред-

ственно в ЕСМ-системе, минуя шаги по сохранению файла на локальном компьютере.

Компания ООО «ДоксВижн» стремится улучшить опыт своих пользователей, и видит удобным решением перечисленных проблем интеграцию Docsvision ЕСМ в Проводник Windows.

1. Постановка задачи

Цель данной работы — разработка прототипа системы, который позволит интегрировать ЕСМ-систему Docsvision в Проводник Windows.

Для достижения поставленной цели необходимо выполнить следующие задачи.

- Изучить возможности, предлагаемые Windows для расширения оболочки; адаптировать пользовательские сценарии и объекты Docsvision к функциональности проводника Windows на основе результатов исследования.
- Разработать архитектуру системы.
- Реализовать прототип системы в разработанной архитектуре, который позволит пользователю взаимодействовать с объектами Docsvision через интерфейс проводника Windows.
- Провести тестирование созданного прототипа.

2. Требования

При интеграции Docsvision ЕСМ в операционную систему Windows необходимо поддерживать основные функциональные возможности уже существующих клиентских приложений по работе с документами, а также сократить количество шагов, которые необходимо выполнить пользователям для выполнения своих задач. Поэтому базовыми функциональными требованиями является предоставление пользователю следующих возможностей.

- Получение доступа к дереву папок и документам, хранящимся в ЕСМ-системе Docsvision, через интерфейс Проводника Windows.
- Создание и удаление объектов Docsvision через Проводник Windows.
- Копирование-вставка файлов из файловой системы в расширение.
- Копирование-вставка файлов из расширения в файловую систему.
- Возможность редактирования файла и сохранения изменений сразу на сервере Docsvision ЕСМ.

Кроме того, важны следующие нефункциональные требования.

- Требование к удобству использования: отсутствие сильной видимой задержки между запросом пользователя и отрисовкой результата.
- Простота поддержки системы и достаточно низкий порог вхождения в выбранные технологии.
- Гибкость архитектуры системы.

3. Обзор

3.1. Предметная область

3.1.1. Docsvision

Система управления корпоративным контентом Docsvision обладает классической трехуровневой архитектурой, которая включает в себя: клиентское приложение, сервер и базу данных. В качестве клиентского приложения на данный момент имеются: Windows-клиент, веб-клиент и некоторые дополнительные клиенты, такие как «Пульс» для мобильных устройств и почтовый клиент для работы с помощью почтовой программы.

Основными объектами Docsvision являются карточки и файлы. Карточки служат для хранения информации в структурированном виде (атрибуты документа или другого моделируемого объекта во внешней системе), могут хранить ссылки на файлы, карточки или другие объекты. У каждого типа карточки есть схема данных, определяющая набор секций со списком атрибутов. В экземпляре карточки в отдельной секции может быть создана одна или более строк, хранящих значения атрибутов.

Файл в Docsvision представлен на двух уровнях:

1. Бинарные данные файла и его атрибуты (размер, дата создания и изменения, признаки архивирования и предварительного удаления и др.).
2. На более высоком — карточка файла с версиями, которая содержит ссылки на версии файла, а именно на бинарные данные разных версий, номера версий, комментарии к версиям и данные авторов версий.

Платформа также предоставляет системные карточки, которые добавляют новые сущности. Примерами таких сущностей могут быть: папки, поисковые запросы, поисковые папки (папки, с которыми ассоциированы поисковые запросы), пользовательские настройки.

Таким образом, основная работа в Docsvision ведется с карточками различных видов, которые обладают своими атрибутами и содержат в себе файлы и их версии. При этом карточки, в свою очередь, могут располагаться в папках для удобства навигации.

3.1.2. Shell Namespace Extension

Пользовательский интерфейс Windows позволяет работать с большим количеством разнообразных объектов. Основными являются папки и файлы. Помимо них существуют еще виртуальные объекты, такие как «Корзина», для выполнения дополнительных задач. Windows Shell (оболочка Windows) [2] организует все эти объекты в иерархическое пространство имен для эффективного доступа и управления ими как со стороны пользователей, так и со стороны приложений.

Windows также предоставляет инструменты, которые позволяют расширять свои возможности и интегрировать приложения в операционную систему. Что, в свою очередь, дает возможность работать с данным приложения через графическую оболочку операционной системы, представляя их в виде виртуальных файлов и папок. Для такой интеграции необходимо разработать расширение пространства имен оболочки (Shell Namespace Extension) [3] [4].

Расширение должно состоять из двух частей.

1. Менеджер данных.
2. Интерфейс между менеджером данных и проводником Windows.

Первая часть может быть реализована полностью на усмотрение разработчика, в соответствии с логикой приложения, которое планируется интегрировать с Windows.

Для работы с папкой в проводнике Windows используется определенный СОМ-объект [5]. При любом взаимодействии пользователя с папкой в проводнике Windows, оболочка обращается к нужному объекту папки, используя

стандартные интерфейсы. Объект реагирует на запрос оболочки согласно своей логике, и оболочка далее обновляет вид Проводника Windows. Таким образом, для реализации второй части требуется организовать данные приложения в такую структуру папок-объектов.

Для создания объекта-папки необходимо реализовать набор интерфейсов: IPersistFolder, IShellFolder, IEnumIDList и другие, в зависимости от желаемых возможностей расширения. Оболочка будет обращаться к ним за обработкой запросов пользователя.

3.2. Обзор существующих решений

3.2.1. Клиентские приложения для работы с Docsvision ECM

Работа над интеграцией ведется в рамках проекта компании ООО «Докс-Вижн», поэтому необходимо рассмотреть имеющиеся на данный момент клиентские приложения, которые можно использовать для работы с ECM-системой Docsvision.

1. Web-клиент — пользователи подключаются к системе через браузер.
2. Windows-клиент — работа с десктопным приложением, которое необходимо установить на клиентский компьютер.
3. Почтовый клиент — позволяет выполнять операции по обработке заданий и участия в согласовании без использования электронной подписи. Передача информации осуществляется с помощью электронных писем, которые формируются серверной частью системы.
4. Мобильное приложение «Пульс» — позволяет выполнять основные операции с документами, используя мобильное устройство.
5. Дополнительный клиент: панель инструментов для MS Office — позволяет создавать карточки напрямую из документов Microsoft Office. Данный клиент позволяет ускорить процесс добавления новой карточки, но только при работе в MS Office, что сильно ограничивает возможности.

3.2.2. Другие ЕСМ-системы

По данным исследования от 2020 года, проведенного TAdviser⁴, по количеству реализованных проектов на российском рынке лидируют СЭД/ЕСМ-системы Directum⁵, Docsvision⁶ и Дело⁷ (рис. 1).



Рис. 1. Источник: https://www.tadviser.ru/index.php/Статья:Самые_популярные_СЭД/ЕСМ-системы

Перечисленные системы для работы с ними также предлагают использовать соответствующие веб-клиенты, десктопные или мобильные приложения. Некоторые решения имеют возможность интеграции в конкретные продукты для создания документов. Однако, ни одна из перечисленных систем не предлага-

⁴https://www.tadviser.ru/index.php/Статья:Самые_популярные_СЭД/ЕСМ-системы

⁵<https://www.directum.ru/>

⁶<https://docsvision.com/>

⁷https://www.eos.ru/eos_products/eos_delo/

ет полной интеграции в проводник Windows для упрощения взаимодействия с системой.

3.2.3. Решения, интегрированные в проводник Windows

Среди популярных продуктов, которые интегрированы в проводник Windows, можно выделить облачные сервисы, такие как Яндекс.Диск⁸ или Google Drive⁹. Интеграция данных сервисов в проводник Windows решает недостатки, описанные в разделе «Введение». Однако сами сервисы не являются ЕСМ-системами и, соответственно, не могут быть использованы как готовое решение заявленной проблемы.

Стоит обратить внимание, что расширение пространства имен оболочки, интегрирующее Google Drive в проводник Windows, имеет открытый исходный код. Интересным решением является создание немонолитного приложения. Система разбита на две части: Google Drive Shell Extension¹⁰ и Windows-службу⁹ Google Drive Proxy¹¹, с которым расширение взаимодействует через пайпы¹¹. Служба, являясь промежуточным звеном, позволяет упростить взаимодействие с Google API и облегчить само расширение.

3.2.4. Вывод

В настоящий момент не существует клиентского приложения для работы с ЕСМ-системой Docsvision, которое бы позволяло избавиться от описанных в разделе «Введение» недостатков. Кроме того, альтернативные ЕСМ-системы, популярные на российском рынке, также не предлагают такого решения.

⁸<https://yandex.ru/support/disk/>

⁹https://www.google.com/intl/en_in/drive/

¹⁰<https://github.com/google/google-drive-shell-extension>

¹¹<https://github.com/google/google-drive-proxy>

4. Описание решения

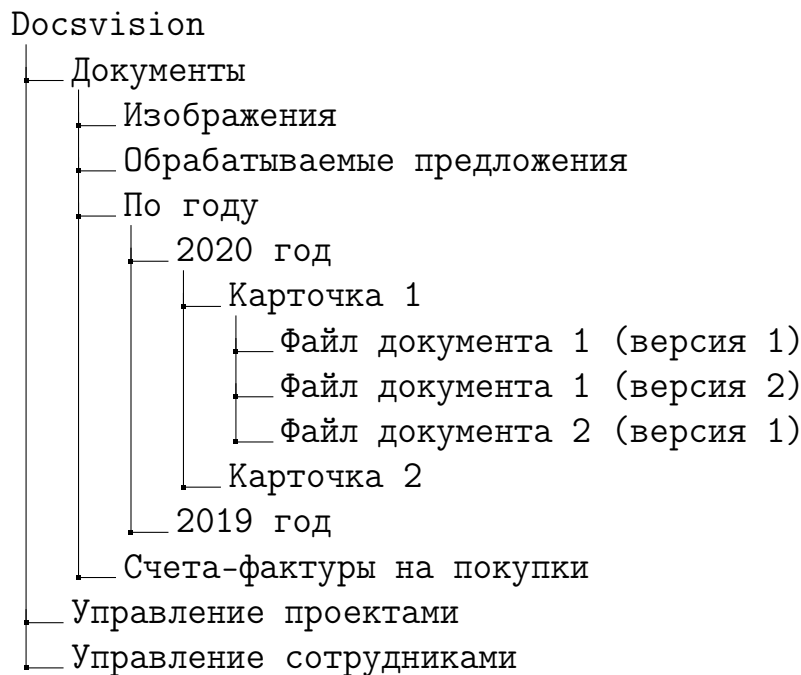
4.1. Адаптация объектов Docsvision для расширения

Базовыми объектами при работе с Docsvision ЕСМ являются карточки и файлы. При этом карточки могут располагаться в различных папках и быть найдены через поисковые запросы с определенными параметрами. Файлы обладают атрибутами и могут иметь несколько версий.

При этом привычными объектами при работе через проводник Windows являются папки и файлы.

Для реализации расширения пространства имен оболочки, необходимо организовать контент модуля Docsvision ЕСМ, который представлен в виде базовых объектов, в древовидную структуру привычных для Windows объектов. Для этого было решено выделить несколько уровней: уровень папок, уровень карточек и уровень файлов. Папки и поисковые папки Docsvision преобразуются в виртуальные папки проводника Windows. При этом папки могут содержать в себе объекты только своего уровня или объекты уровня ниже: папки или карточки. Карточки тоже преобразуются в объекты типа папка проводника, однако обладают другими атрибутами и отличной от обычных папок логикой взаимодействия. Внутри карточек могут находиться файлы, которые преобразуются в виртуальные файлы проводника Windows.

Основываясь на данных рассуждениях, контент модуля Docsvision можно представить следующим образом:



4.2. Архитектура

4.2.1. Ограничения

При разработке архитектуры решения необходимо учитывать ограничения, которые накладываются как со стороны Docsvision, так и со стороны возможностей разработки Shell Namespace Extension. Среди таких ограничений можно выделить следующие.

1. Microsoft не рекомендует использовать .NET для реализации внутрипроцессных расширений оболочки¹², в связи с:
 - возможными проблемами с различными версиями загружаемого рантайма;
 - возможными проблемами с производительностью;
 - проблемами с блокировками на уровне STA;
 - недетерминированностью времени жизни объектов в .NET.
2. Для решения поставленной задачи наиболее подходит предоставляемое платформой Docsvision API на базе платформы .NET. Предоставляемое

¹²<https://docs.microsoft.com/en-us/windows/win32/shell/shell-and-managed-code>

COM API является низкоуровневым, и в нем отсутствует реализация бизнес-логики по работе с карточками типа документ, а самостоятельная реализация аналога приведет к дублированию кода.

Помимо этого, на выбор архитектуры влияют особенности взаимодействия с компонентами. Для работы с контентом модуля Docsvision необходимо получить объект «пользовательская сессия». Процесс создания данного объекта на стороне Docsvision может занимать значительное время. Если при каждом взаимодействии пользователя с виртуальной папкой расширения будет происходить создание новой сессии, производительность такого клиента будет сильно снижена.

4.2.2. Разработанная архитектура

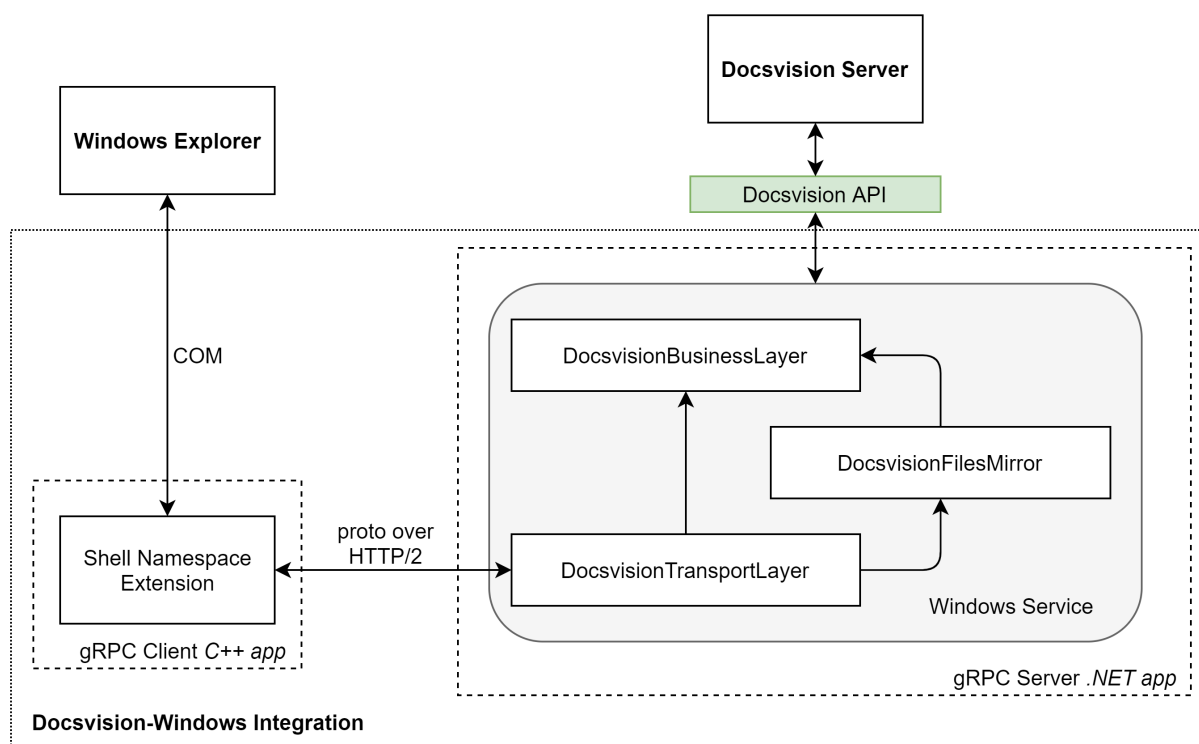


Рис. 2. Архитектура системы

Так как разработка Shell Namespace Extension является достаточно сложной задачей, разделение системы на компоненты используется в известных продуктах для упрощения разработки и удобства поддержки (пример: Google

Drive Shell Extension, описанный в подразделе «Решения, интегрированные в проводник Windows»). Основываясь на описанных ограничениях и возможных проблемах, было решено создать клиент-серверную архитектуру (рис. 2).

Система разбита на две большие компоненты — C++ клиент и .NET сервер. Клиент представляет из себя расширение пространства имен оболочки. Он реализует требуемые интерфейсы, но лишен основной логики обработки данных. Он является промежуточным звеном между проводником Windows и серверной частью. Основной задачей клиента является передача всех запросов, которые приходят от Проводника Windows, на серверную часть системы. После чего клиент получает обновленную информацию, которая будет использоваться Проводником для отображения.

Большая часть логики обработки запросов и взаимодействие с API Docsvision описаны на серверной части системы, реализованной с использованием .NET. Сервер является менеджером данных, обрабатывает запросы от пользователя и обращается к Docsvision ECM через API за необходимой информацией.

При этом серверная часть реализована в уровневом стиле. Транспортный уровень отвечает только за взаимодействие с клиентом и обращается к бизнес-уровню за получением информации. Бизнес-уровень ничего не знает о транспортном уровне, но предоставляет интерфейс для взаимодействия с собой. Бизнес-уровень общается с сервером Docsvision и реализует основную логику менеджера данных. Он работает с моделями данных, соответствующими базовым объектам Docsvision — карточки, папки, файлы.

Клиент работает уже с другими моделями данных — только папками и файлами. Поэтому транспортный уровень также отвечает за преобразование моделей бизнес-уровня в модели, подходящие для проводника Windows.

Протокол взаимодействия между клиентом и сервером позволяет передавать запросы на такие операции как: скачивание и загрузка файлов, получение

ние содержимого папки, операции удаления, переименования и создания новых объектов, а также некоторые другие для поддержки всех необходимых функциональных возможностей.

На рис. 2 видно, что в серверной части, помимо бизнес и транспортного уровня, присутствует модуль DocsvisionFilesMirror. Это вспомогательный модуль, реализованный для поддержки редактирования файлов. Решение вынести его на серверную часть системы было принято для ускорения процесса разработки.

Использование Windows службы для хостинга данного приложения позволит создать пользовательскую сессию один раз при загрузке компьютера и далее работать с контентом без потери производительности.

4.2.3. Технологии для взаимодействия между компонентами

Реализованный прототип планируется использовать для разработки системы с увеличенным размером функциональных возможностей. Поэтому важными нефункциональными требованиями к системе являлись простота поддержки, быстрый порог вхождения в используемые технологии и возможность развития системы (гибкость).

С учетом данных требований в качестве технологий для взаимодействия между компонентами были выбраны: механизм сериализации-десериализации данных Protocol Buffers [7] и средство удаленного вызова процедур gRPC [8].

Эти технологии обладают следующими преимуществами.

- Актуальность. Microsoft после перехода на .NET Core (а далее на .net5) рекомендует разработчикам, которые раньше использовали технологию WCF, переходить на gRPC, ссылаясь на ряд преимуществ более современной технологии [13].

¹³ <https://docs.microsoft.com/en-us/dotnet/architecture/grpc-for-wcf-developers/why-grpc>

- Поддержка языков, используемых в разработке (C++ и C#).
- Достаточно низкий порог вхождения и простота поддержки.
- Благодаря использованию в качестве транспорта HTTP/2, имеется возможность работы как на локальном компьютере, так и по сети.

Такой выбор технологий позволит в дальнейшем развивать систему: имеется возможность серверную часть вынести на отдельный компьютер, тем самым облегчив приложение на компьютере пользователя. При этом архитектура системы может выглядеть следующим образом (рис. 3).

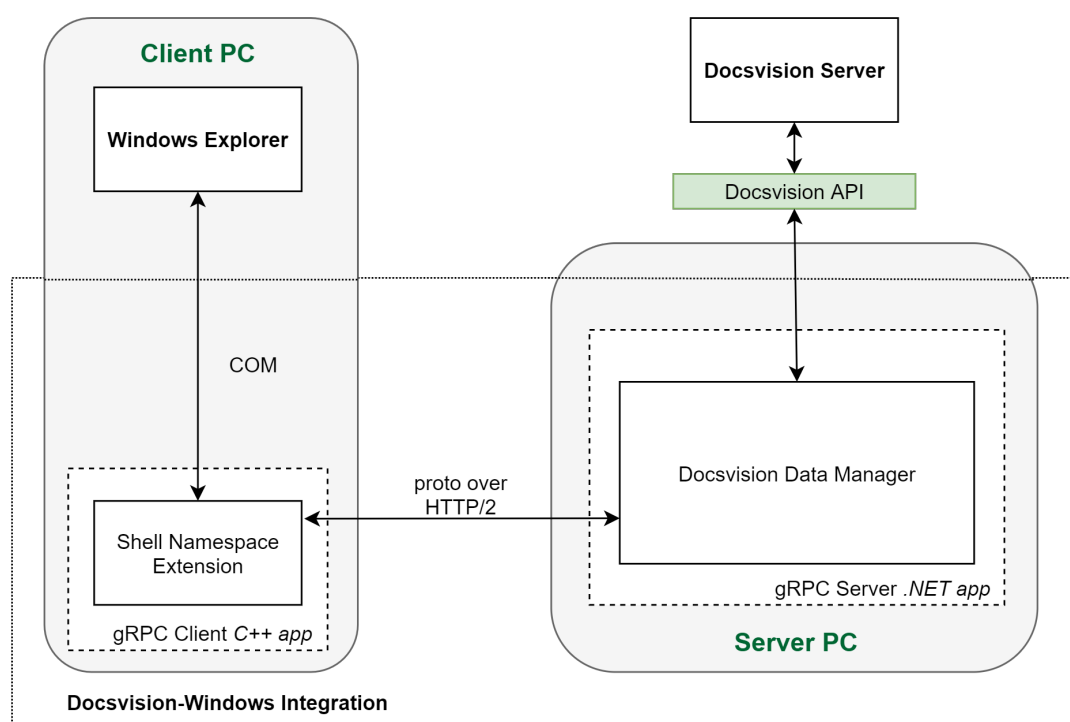


Рис. 3. Архитектура системы при выносе серверной части с клиентского компьютера

Благодаря уровневой архитектуре серверного приложения и независимости верхних слоев от нижних, легко заменить Windows-службу на IIS¹⁴ (Internet Information Services) для хостинга серверной компоненты при ее переносе на удаленный компьютер.

¹⁴<https://www.iis.net/>

4.3. Особенности реализации: преобразование моделей данных

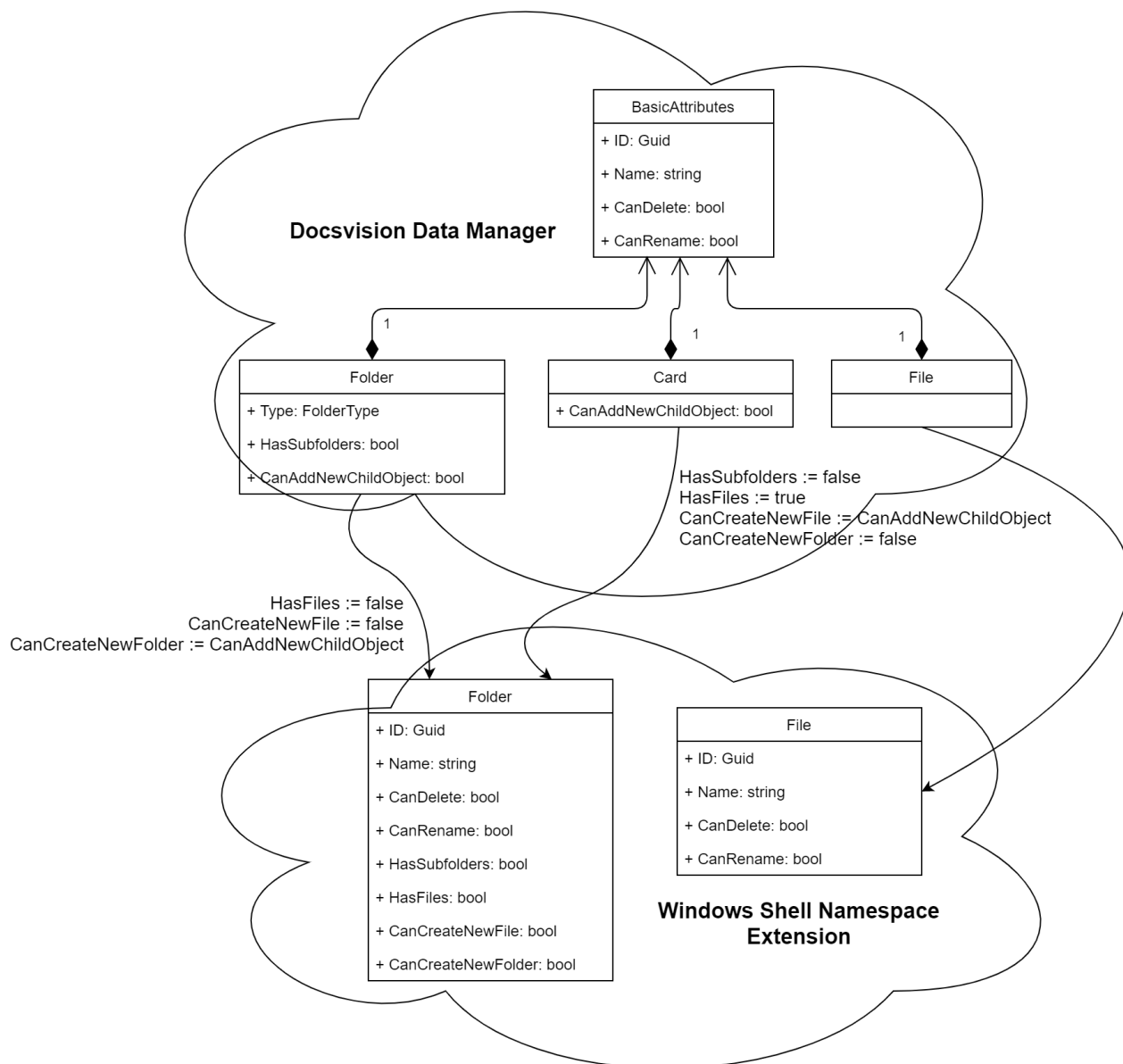


Рис. 4. Преобразование моделей бизнес-уровня в модели расширения

Объекты Docsvision несколько отличаются от объектов, с которыми работает проводник Windows. Транспортный уровень преобразует модели бизнес-уровня в модели, которые используются в расширении пространства имен оболочки. Для корректной трансляции контекстов можно использовать карту, изображенную на рис. [4](#).

4.4. Проводник Windows как графическая оболочка для контента Docsvision ECM

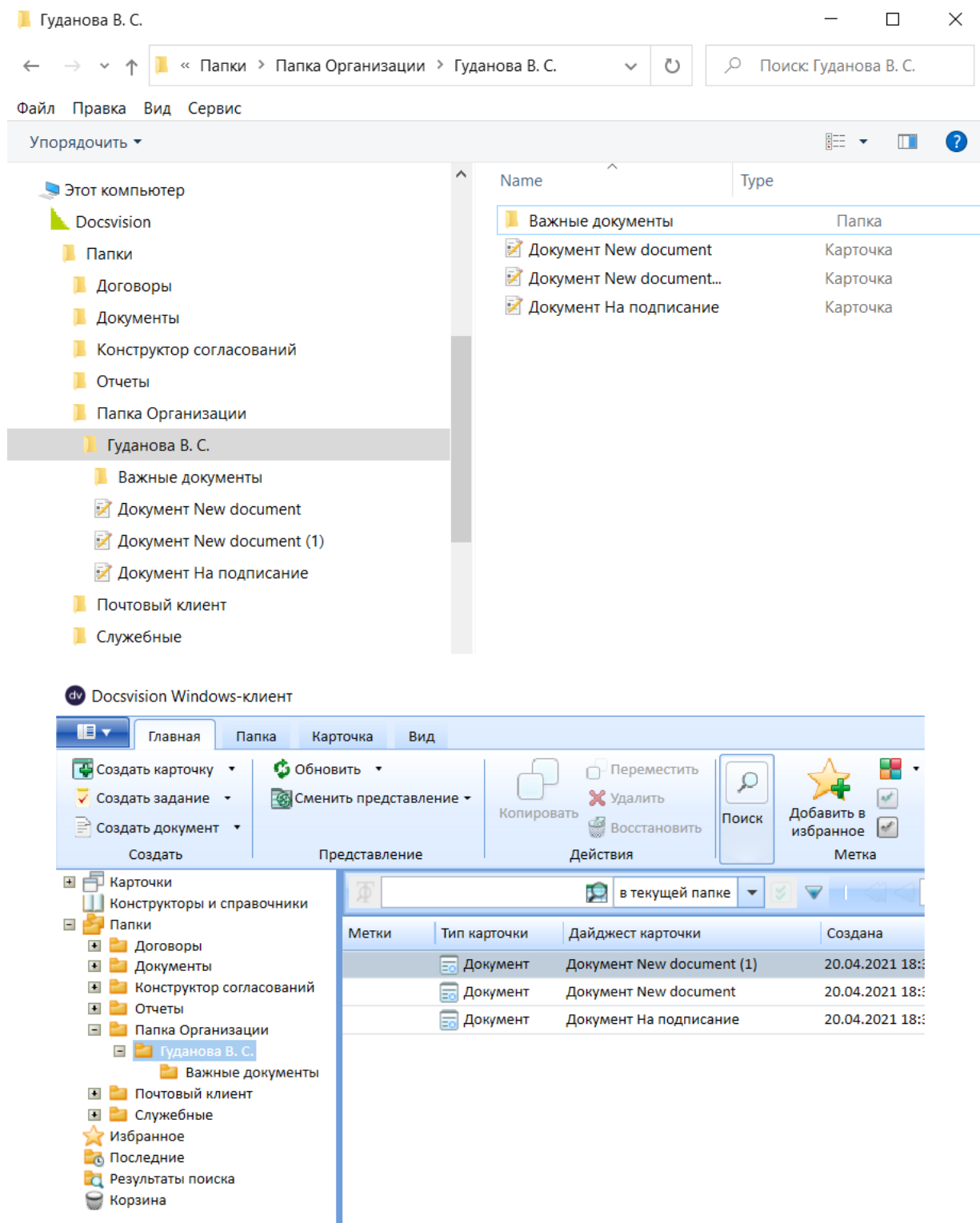


Рис. 5. Интеграция Docsvision в Проводник

Полученный результат интеграции Docsvision в Проводник можно увидеть на рис. 5. На верхнем изображении представлено дерево папок и объекты, отображаемые в проводнике Windows. Снизу показан интерфейс десктопного клиентского приложения Docsvision с отображением дерева элементов того же пользователя.

5. Тестирование

Расширение пространства имён оболочки обладает большой сложностью. Поэтому было решено провести ее тестирование в два этапа: без взаимодействия с Docsvision API и после реализации взаимодействия для проверки соответствия функциональным требованиям.

Для поддержки промежуточного тестирования сам процесс реализации был разбит на два больших этапа.

1. Реализация расширения, внешнего сервиса и тестовой заглушки функциональности сервиса, позволяющей протестировать всю возможную функциональность расширения.
2. Реализация функциональности сервиса по работе с Docsvision.
 - 2.1 Поддержка отображения дерева папок и карточек документов; поддержка операций над деревом папок.
 - 2.2 Поддержка работы поисковых папок.
 - 2.3 Поддержка операций над документом и отображения содержимого документа в виде набора файлов.
 - 2.4 Поддержка операций над файлами и редактирования файлов.

Совместно с представителями ООО «ДоксВижн» были разработаны тестовые сценарии, покрывающее функциональность, заявленную в требованиях. На основе данных сценариев было проведено ручное тестирование.

Промежуточное тестирование после первого этапа показало работоспособность C++ компоненты системы и ее готовность к интеграции с Docsvision ЕСМ. Расширение корректно выполняет основные задачи, такие как: навигация по папкам, создание новых папок и файлов, копирование-вставка файлов из расширения в файловую систему и наоборот, удаление объектов.

В ходе тестирования после реализации второго этапа были выявлены дефекты, связанные с редактированием файлов и автоматической загрузкой изменений на сервер Docsvision. Причиной возникновения дефектов является особенность разделения файлов в карточках документов Docsvision на главные и дополнительные, что не было учтено в первый раз при реализации C++ расширения. Однако данные дефекты были оперативно исправлены, и окончательное тестирование после второго этапа показало соответствие системы заявленным функциональным требованиям.

Помимо ручного тестирования, была проведена демо-презентация реализованного прототипа системы специалистам ООО «ДоксВижн». Прототип собрал положительные отзывы, по оценкам аналитиков показал работоспособность бизнес-идеи интеграции Docsvision ЕСМ в Проводник Windows и возможность дальнейшего развития.

Заключение

В ходе проделанной работы были получены следующие результаты.

- Объекты и пользовательские сценарии Docsvision адаптированы к функциональным возможностям Проводника Windows.
- Разработана архитектура приложения.
- Разработан прототип системы (с использованием технологий C#, C++, COM, gRPC, Protocol Buffers, Docsvision API), предоставляющий пользователю следующие возможности:
 - получение доступа к документам, хранящимся в ЕСМ-системе Docsvision, через интерфейс проводника Windows;
 - создание и удаление объектов Docsvision (папок, файлов, карточек и т.д.) через проводник Windows;
 - копирование-вставка файлов из файловой системы в расширение и наоборот;
 - редактирование файлов.
- Разработаны тестовые сценарии, покрывающие заявленную базовую функциональность. На основе созданных сценариев проведено ручное тестирование, результаты которого показали, что реализованный прототип соответствует заявленным функциональным требованиям.

Список используемой литературы

- [1] Docsvision. — URL: <https://docsvision.com/> (дата обращения: 24.04.2021)
- [2] Microsoft Documentation. Windows Shell. — URL: <https://docs.microsoft.com/en-us/windows/win32/shell/shell-entry> (дата обращения: 24.04.2021)
- [3] Microsoft Documentation. Shell Namespace Extension. — URL: <https://docs.microsoft.com/en-us/windows/win32/shell/nse-works> (дата обращения: 24.04.2020)
- [4] Esposito D. Visual C++ Windows Shell Programming. — Wrox Press Ltd., 1998
- [5] Microsoft Documentation. The Component Object Model. — URL: <https://docs.microsoft.com/en-us/windows/win32/com/the-component-object-model> (дата обращения: 24.04.2020)
- [6] Stanley Lippman, Josée Lajoie, and Barbara E. Moo. C++ Primer. 5 edition. — Addison-Wesley Professional, 2012
- [7] Protocol Buffers Documentation. — URL: <https://developers.google.com/protocol-buffers/docs/proto3> (дата обращения: 24.04.2021)
- [8] gRPC Documentation. — URL: <https://grpc.io/docs/> (дата обращения: 24.04.2021)
- [9] Microsoft Documentation. Introduction To Windows Service Application. — URL: <https://docs.microsoft.com/en-us/dotnet/framework/windows-services/introduction-to-windows-service-applications> (дата обращения: 24.04.2021)
- [10] Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е изд. — СПб.: Питер, 2013

- [11] Харт Джонсон М. Системное программирование в среде Windows, Глава 11. Взаимодействие между процессами. 3-е изд. — Издательский дом «Вильямс», 2005