

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

Автономная некоммерческая организация науки и образования
«Институт компьютеринга»

УТВЕРЖДАЮ

Директор АНО «Институт компьютеринга»

_____ /А.И. Миков/

«__» _____ 2008 г.

РЕКЛАМНО-ТЕХНИЧЕСКОЕ ОПИСАНИЕ

Система интеграции распределенных приложений Simple BizTalk Server
(SBTServer)

.31569113.00003-02 99 01

Листов 76

Разработчики:

_____ /Лукиных И.А./

_____ / Лядова Л.Н./

_____ / Рыжков С.А./

_____ / Рычков А.Ю./

08.08.2008

1. Функциональное назначение программы, область ее применения, ее ограничения

1.1. Назначение программы

Назначение программной системы Simple BizTalk Server (SBTServer) – *интеграция распределенных приложений и баз данных* на основе использования технологии BizTalk Server.

В задачи сервера интеграции входит реализация взаимодействия приложений в различных режимах, обмен сообщениями между ними в формате BizTalk и других форматах, основанных на XML, с использованием различных протоколов.

Сервер BizTalk – это программное обеспечение, способное считывать и обрабатывать документы BizTalk. Поскольку инфраструктура BizTalk базируется на XML и, как следствие, является самодокументированной, BizTalk-сервер должен уметь загружать документы XML и работать с их содержимым. Это первое требование. В спецификации документов и сообщений BizTalk Framework 2.0 базовые задачи выполняет BizTalk-совместимый сервер (BFC, BizTalk Framework Compatible).

Сервер BizTalk должен быть достаточно гибок для работы в быстро меняющейся бизнес-среде. Он должен обладать гибким интерфейсом, позволяющим сосуществовать и взаимодействовать с традиционными приложениями. Поскольку пользователи работают в разных системах, сервер должен обеспечивать взаимодействие с разными платформами и разными языками программирования [10].

Важнейшими *функциями сервера BizTalk* являются:

- *Надежная пересылка документов по любому протоколу.*

Чтобы использовать документ BizTalk, его надо доставить получателю. Если последний – партнер по бизнесу, требуется система доставки документов. Наиболее распространены протоколы HTTP, FTP и упрощенный протокол передачи почтовых сообщений (SMTP).

- *Защита информации.*

Спецификация BizTalk Framework 2.0 позволяет защитить документ BizTalk с помощью протоколов S/MIME и PKCS. Можно также применить протокол HTTPS, широко применяемый в Интернете и использующий асимметрическое шифрование.

- *Маршрутизация документов.*

Наиболее важная способность сервера BizTalk заключается в передаче документов из одной системы в любую другую. Кроме того,

полученный от партнера ответ обычно нужно автоматически переадресовывать в соответствующее подразделение компании.

– *Документооборот.*

Полученный документ BizTalk должен быть проанализирован, после чего приложение сможет выяснить его тип. Выяснив тип документа и его назначение, можно приступить к его обработке. Для всех входящих документов сервер BizTalk должен иметь несколько вариантов их обработки.

– *Синхронная и асинхронная связь.*

Сервер BizTalk должен иметь возможность синхронной отправки сообщений. Синхронная связь – основной тип обмена сообщениями и обязательно должна поддерживаться сервером. Но если отслеживать подтверждение каждого запроса, системе может не хватить ресурсов. Поэтому сервер должен поддерживать и асинхронный метод передачи.

– *Очередь сообщений.*

Сервер BizTalk должен обеспечивать постановку сообщений в очередь, чтобы иметь возможность переключиться на выполнение другой задачи до окончания обработки сообщения.

– *Пакетный режим.*

BizTalk-сервер должен поддерживать назначение параметров доставки документов BizTalk и их передачу в пакетном режиме в соответствии с требованиями партнера.

– *Отслеживание документов.*

Сервер BizTalk должен иметь развитые возможности отслеживания выполнения транзакций, включая:

- отслеживание отдельных документов и пакетов документов;
- выявление неверно доставленных или не доставленных документов;
- возможность опрашивать БД протоколирования по отдельным документам, по партнерам и по виду деятельности;
- протоколирование времени выполнения всех действий с документами;
- хранение данных, выбранных пользователем.

– *Управление взаимодействием с партнерами.*

Для любого взаимодействия BizTalk-сервер должен уметь отслеживать все аспекты создания и обработки транзакций с конкретным партнером. Кроме того, он должен уметь различать партнеров, с которыми уже налажены деловые отношения, и новых.

– *Масштабируемость.*

Сервер BizTalk должен обеспечивать выполнение необходимого числа транзакций в день с учетом перспектив расширения бизнеса. При этом следует учитывать не количество нынешних партнеров, а число компаний, с которыми потенциально можно наладить связи.

– *Преобразование документов.*

Сервер BizTalk должен уметь различать разные типы входящих документов и автоматически выполнять необходимые преобразования. Должна быть предусмотрена возможность выбора преобразования для каждого партнера, с которым работает сервер BizTalk.

– *Встраивание модулей других производителей.*

Сервер BizTalk должен быть открыт для встраивания расширений, повышающих эффективность взаимодействия с партнерами из конкретных отраслей или между определенными группами партнеров.

– *Прикладные интерфейсы.*

Сервер BizTalk – это просто механизм, который создает и отслеживает документы BizTalk. В его задачи не входит поддержка всего документооборота компании. Сервер BizTalk должен обладать хорошо развитым интерфейсом, обеспечивающим взаимодействие с другими приложениями.

– *Адаптация к изменениям.*

BizTalk-сервер должен быть достаточно гибким, чтобы справиться с проблемами, возникающими при работе с разнообразными партнерами, и с изменением условий и параметров партнеров. Сервер BizTalk должен поддерживать расширение стандартных схем, применяемых в конкретной отрасли.

– *Ориентация на нужды пользователей.*

Сервер должен обладать достаточно развитым интерфейсом пользователя, позволяющим выполнять основные операции: создавать соглашения и изменять их параметры, описывать преобразование входящих и исходящих документов, изменять структуру данных, создавать отчеты о работе с конкретными партнерами и о функционировании системы в целом. А также система должна поддерживать удаленный доступ пользователей [28].

Сервер SBTSerвер – основа реализации систем документооборота, средств интеграции информационных систем бизнес-партнеров, баз данных в распределенной гетерогенной среде.

1.2. Область применения программы

Документооборот является одной из необходимых составляющих любой организации. Именно в наше время, когда все большее предпочтение отдается электронной документации взамен бумажной, появились широкие перспективы автоматизации этой сферы деятельности.

Еще совсем недавно общепризнанного формата для организации документооборота не было (в отдельных областях (например, в электронной коммерции для организации взаимодействия бизнес-партнеров) использовались свои стандарты (EDI и т.п.), но они являются достаточно громоздкими). С появлением XML ситуация стала изменяться и в настоящее время именно он стал таким стандартом [26, 28, 36]. Многие современные системы теперь поддерживают этот формат и используют его для взаимодействия с внешним миром, учитывая универсальность этого формата.

Обычно мысли об интеграции нескольких информационных систем (ИС) разных организаций (или даже подсистем подразделений внутри одной организации) возникают уже после создания собственных ИС, причем по мере роста и развития корпоративной информационной инфраструктуры нередко оказывается, что отдельные ее подсистемы, выросшие из небольших и, как правило, разнородных приложений, «говорят на разных языках», основаны зачастую на разных СУБД, используют разные протоколы и разные форматы хранения информации [32]. Даже если исходно это было готовое решение, а не разработанное внутри компании, все равно высоки шансы, что программные продукты были куплены у разных фирм и потому слабо «понимают друг друга». Группа задач, решающих эти проблемы, объединяется под названием Enterprise Application Integration (EAI) – интеграция систем (приложений) предприятия [14, 23]. В конечном счете, интеграционное решение призвано снизить эксплуатационные расходы на поддержку программных систем, а также обеспечить доступность, целостность и непротиворечивость информации, жизненно важной для бизнеса компании [30].

Общие подходы к решению данной проблемы определяет *инфраструктура BizTalk* [52, 28, 43]. Однако программные продукты, основанные на этих технологиях (например, Microsoft BizTalk Server), требуют для установки значительных ресурсов.

На сегодняшний день очень много компаний предлагает свои услуги по интеграции приложений предприятия. Есть фирмы-лидеры в предоставлении подобных услуг. Их программные продукты обладают обширным спектром возможностей и, по заверениям компаний-разработчиков, обеспечивают уникальный и универсальный в своём роде подход. Но такие программные продукты разрабатываются под потребности мощных корпораций. Фирмы малого бизнеса не могут себе позволить закупить программные комплексы такого масштаба, да они и не нуждаются в полном функциональном наборе

таких приложений. Тем не менее, надобность в интеграции возникает и у таких компаний. Хотелось бы иметь *недорогой, достаточно простой в изучении программный продукт, который бы, тем не менее, содержал в себе необходимую функциональность по интеграции систем.*

Именно такая задача решена при создании программной системы Simple BizTalk Server (SBTServer).

Основные требования, реализованные в программной системе:

- Нетребовательность к программным и аппаратным ресурсам.
- Универсальность представления связей между приложениями.
- Расширяемость, настраиваемость на различные предметные области.
- Мониторинг документов, ошибок системы, надежность обработки документа в системе.
- Поддержка различных протоколов взаимодействия.

1.3. Ограничения использования программы

Программная система Simple BizTalk Server (SBTServer) позволяет настраиваться на различные программные платформы, работать под управлением различных операционных систем Microsoft, настраиваться на различные условия эксплуатации.

Ограничения использования системы SBTServer определяются только требованиями лицензионной чистоты и требованиями, предъявляемыми использованными при ее разработке программными средствами, а также требованиями runtime-компонентов, используемых для организации ее функционирования. Эти средства описываются ниже.

2. Техническое описание программы

2.1. Модель интеграции

На рис. 1 показана схема интеграции, которая описана в BizTalk Framework, и логические структуры, блоки с законченной функциональностью, которые будут необходимы для работы всей системы.

Благодаря этой инфраструктуре можно сразу ввести несколько понятий. В первую очередь понятие *Документа*. Именно *Документ* обрабатывается в системе и именно благодаря *Документу* осуществляется *информационно-ориентированная интеграция*. Всё, что попадает в систему для обработки, является *Документом*. Любая информация, даже если два *Приложения* обмениваются сообщениями через *VFC-сервер*, представляется в системе как *Документ*.

Далее можно ввести такое понятие, как *Приложение*. *Приложение* является отправителем документа, *Приложение* же является и получателем этого *Документа*.

Эти понятия – *внешние* по отношению к системе.

Итак, *Приложение* отправляет *Документ ВФС-серверу* и он должен суметь его принять. Существующие приложения могут отправлять эти документы по совершенно различным протоколам. Таким образом, выделим такое понятие, как *Адаптер*: *Приложение* отправляет *Документ* такому *Адаптеру*, который работает по совместимому *протоколу*. Именно благодаря *Адаптеру* документ попадает внутрь сервера для его последующей обработки. Точно так же необходимо и обратное действие – обработанный *Документ* необходимо доставить *Приложению-получателю*. И в этом случае необходимо воспользоваться *Адаптером*, который передаст сообщение по тому протоколу, по которому общается *Приложение-получатель*.

Адаптеры являются *граничным понятием* между системой и внешним миром. Будем разделять *Адаптеры*, которые служат для получения, *захвата Документов*, назовем их *Функциями Захвата*, и *Адаптеры*, которые обеспечивают отправку *Документа*, будем называть их *Функциями Отправки*.

Перейдем теперь к *внутреннему устройству системы*. Выделим понятие *Канала*. Проходя через *Канал*, *Документ* будет подвергаться всей необходимой обработке внутри системы. С учетом того, что сервер интеграции должен поддерживать преобразование между разными форматами *Документов*, это преобразование будет происходить именно в *Канале*. А если предусмотреть наложение на информационно-ориентированную интеграцию еще и возможность выполнения бизнес-процессов, то начало такого бизнес-процесса должно быть именно в *Канале*.

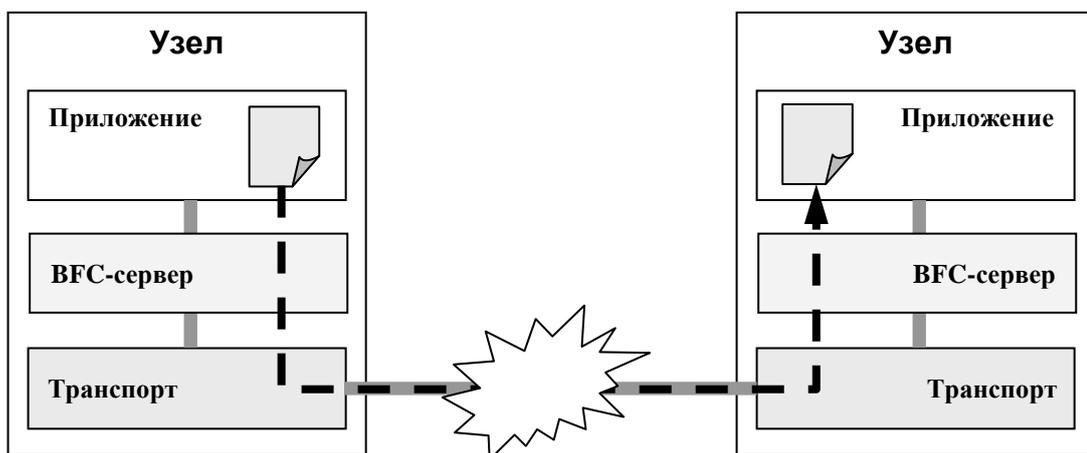


Рис. 1. Логические уровни модели

Необходимо обеспечить возможность получения *Документов* от нескольких *Приложений* с одинаковой логикой внутренней обработки. Другими словами, нельзя, чтобы *Адаптеры* были привязаны непосредственно к *Каналу*.

Выделим функционал по необходимой настройке *Адаптера* в отдельное понятие – *Сервис Захвата*. *Сервис Захвата*, как оболочка над *Адаптером*, содержит все необходимые настройки для корректной работы *Адаптера* и реализует передачу полученных *Документов* в *Канал*. Таким образом, несколько *Сервисов Захвата* могут передавать *Документы* в один *Канал*, используя один и тот же *Адаптер*.

Точно также необходимо реализовать возможность передачи сообщений нескольким *Приложениям-получателям*. В результате выделяется понятие *Сервис Отправки*. *Документ* из *Канала* может попасть в один или несколько *Сервисов Отправки*, который содержит всю необходимую настройку для *Адаптера*, чтобы осуществить передачу *Документа Приложению*.

Таким образом, имеем модель, представленную на рис. 2.

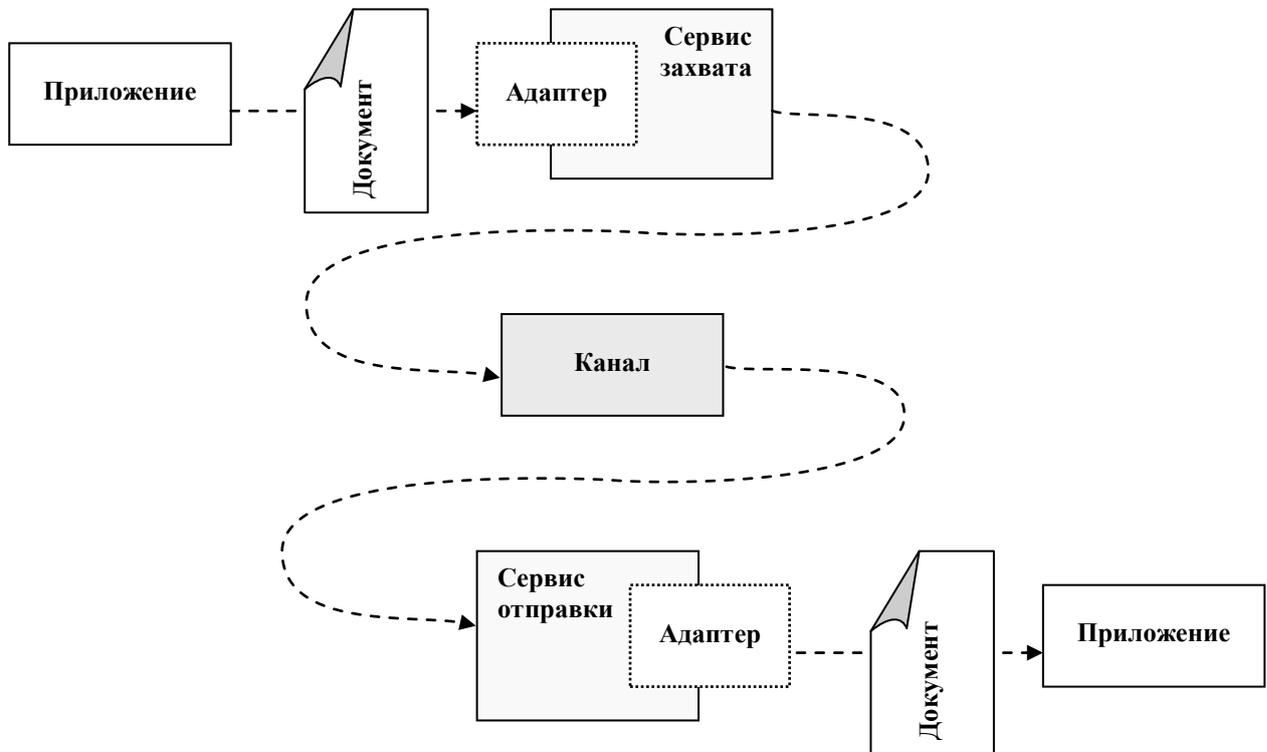


Рис. 2. Модель интеграции

На описанную модель с легкостью накладывается модель Pub-Sub (*Publish-and-Subscribe*). В качестве *Издателя* и *Подписчика* выступает *Приложение*. Для того чтобы *Приложение* стало *Издателем* необходимо создать *Сервис Захвата*. Через этот *Сервис Захвата* *Приложение* будет помещать *Информацию* в *Теме*. Аналогом *Информации* в этой модели является *Документ*. А в качестве *Темы* можно рассматривать *Канал*. Для того чтобы стать *Подписчиком* определенной *Темы* необходимо создать *Сервис Отправки* для этого *Приложения* и соединить его с нужным *Каналом*. Таким образом, можно разослать один и тот же *Документ* нескольким *Приложениям*, что и является основой модели Pub-Sub.

Что касается модели P2P (*Point-to-Point*), то тут реализация немного сложнее. Нельзя напрямую сопоставить *Очередь* и *Канал*: если какой-то *Документ* попал в *Канал* и прошел в нем обработку, то он тиражируется по всем *Сервисам Отправки*. Таким образом, нарушается принцип этой модели в том, что получателем информации будет только одно из *Приложений*. Выхода два: организовать необходимую функциональность внутри системы или предоставить ее реализацию вне системы. В первом случае можно надстроить бизнес-процесс, который бы позволял управлять ситуацией и определять, кто окажется получателем на данном этапе. Во втором случае можно использовать понятие *очереди сообщений Microsoft* (*Microsoft Message Queue, MSMQ*). Итак, можно создать *Сервис Отправки*, который будет отправлять *Документ* в определенную *Очередь сообщений*, тогда *Получатель* должен опросить эту *Очередь* и забрать из нее информацию. Либо наоборот, какое-то *Приложение* отправляет информацию в эту *Очередь*, а посредством *Сервиса Захвата* можно опрашивать эту *Очередь* и получать оттуда *Документы* для последующей обработки в сервере интеграции. Для этой цели необходимо реализовать *Адаптеры* для взаимосвязи с *MSMQ*.

Введем в модель понятие *Очереди документов*. На рис. 2 показан пример идеальной работы сервера интеграции. *Документ* прошел все стадии обработки без ошибок: был получен из вне через *Сервис Захвата*, прошел проверки внутри *Канала*, при необходимости был перекодирован в другой формат и благополучно отправлен через *Сервис Отправки*. Назовем эти стадии обработки *Рабочей очередью*. Если на каком-то из этапов обработки *Документа* происходит ошибка, *Документ* необходимо сохранить в системе и дать возможность проанализировать ошибку, которая не позволила продолжить обработку. Для этого введем понятие *Очереди приостановленных документов*. Наконец, выделим среди ошибок предполагаемые помехи и временную недоступность транспортной среды во время отправки *Документов*. Подобные ошибки необходимо обязательно фиксировать в системе, но переводить *Документ* из-за временных помех из *Рабочей очереди* в *Очередь приостановленных документов* – значит создавать дополнительную нагрузку на администратора сервера и вероятно задержать отправку документов. Для того, чтобы минимизировать подобный риск, введем понятие *Очереди повторов отправки*, что позволит прежде, чем перевести *Документ* в *Очередь приостановленных документов*, предпринять несколько попыток его отправки.

После введения понятия *Очереди приостановленных документов* появилась необходимость введения понятия *Ошибки*. *Ошибка* – это тоже объект системы со своим набором атрибутов. Местом фиксации ошибок служит *Журнал ошибок*. Именно на объекты *Журнала ошибок* должны сохранять ссылку элементы *Очереди приостановленных документов*. Неотъемлемой частью любой ИС

является также *Журнал событий*, необходимый для мониторинга системы и проверки её работоспособности.

2.2. Документ BizTalk

BizTalk предлагает набор правил и исходный набор тегов для создания схем электронного документооборота, а также рекомендаций по созданию собственных наборов тегов для обмена информацией с бизнес-партнерами, определяет единый подход к применению XML. Документы рассматриваются как сообщения, которыми обмениваются бизнес-процессы, включая методы создания документов и их маршрутизацию. BizTalk обеспечивает механизм «почтовой рассылки» XML-документов.

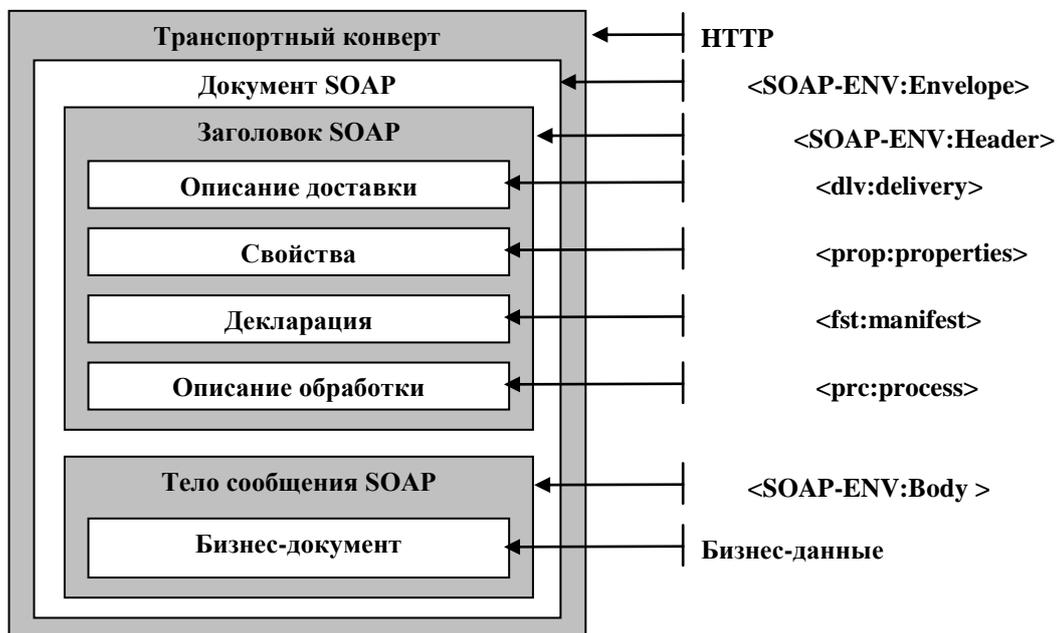


Рис. 3. Документ BizTalk

Документ BizTalk представляет собой сообщение в формате SOAP (рис. 3), в теле которого содержится один или более *бизнес-документов*. Его можно рассматривать как расширение документа SOAP. Спецификация BizTalk содержит несколько *пространств имен*, каждое из которых позволяет решить определенные задачи. Теги этих пространств имен называют *бизнес-тегами* (BizTags). Документ содержится в элементе *SOAP Envelope* – корневом элементе документа. В состав этого документа входит необязательный элемент *Header* и обязательный элемент *Body*. В сообщении BizTalk элемент *SOAP:Header* обязателен, так как он содержит некоторые обязательные бизнес-теги. Элемент *SOAP:Body* содержит бизнес-данные, состоящие из одного или нескольких документов.

Документ BizTalk не выполняет никаких функций, пока не отправлен. Транспортировка документа – задача транспортного конверта, в роли которого может выступать практически любой протокол электронной передачи, включая HTTP, SMTP и FTP.

Подход к разработке документов с использованием пространств имен и схем XML позволяет создавать виртуальные конверты с виртуальными документами внутри.

Синтаксис определения типа документа (DTD) допускает наличие только одного определения, а документ BizTalk фактически содержит два типа документа, для чего используется сокращенный синтаксис схем данных (XML Data Reduced, XDR). Таким образом, обработка транзакций BizTalk на основе синтаксиса определения типа документа невозможна [22].

При получении документа сервер BizTalk приступает к его чтению и анализу. В первую очередь он считывает бизнес-тег *dlv:to*, чтобы выяснить, кому предназначен документ. Затем сервер BizTalk, возможно, займется анализом и обработкой данных, внесением изменений в БД, передачей «вложенного» документа на обработку соответствующему приложению и т.п. Тип обработки целиком определяется бизнес-процессами получателя сообщения.

BizTalk предлагает набор правил и исходный набор тегов для создания схем электронного документооборота, а также рекомендаций по созданию собственных наборов тегов для обмена информацией с бизнес-партнерами, определяет единый подход к применению XML. Документы рассматриваются как сообщения, которыми обмениваются бизнес-процессы, включая методы создания документов и их маршрутизацию. BizTalk обеспечивает механизм «почтовой рассылки» XML-документов.

2.3. Структура программного продукта

Сервер был разработан на основе информационно-ориентированной интеграции, в качестве архитектуры была выбрана архитектура с ядром интеграции, созданная система опирается на собственную математическую модель.

Разработка приложения SBTServer состоит из нескольких этапов:

- Разработка структуры программного комплекса. Выделение программных модулей.
- Разработка структуры базы данных сервера и написание SQL-скрипта для создания этой базы максимально независимым от выбора СУБД.
- Разработка приложения, непосредственно выполняющего функции сервера.
- Разработка приложения для администрирования сервера.
- Разработка приложения для работы с объектами документооборота.

Таким образом, все приложение состоит из *следующих программных модулей*:

1. SBTConfig.exe – приложение для работы с настроечным файлом конфигурации сервера по средствам Windows-интерфейса. Настроечный файл можно редактировать и через стандартный редактор. Приложение реализовано для минимизации ошибок в настройке.
2. SBTAdmin.exe – Windows-приложение для администрирования сервера интеграции. Предназначено для регистрации новых адаптеров, просмотра лога ошибок и событий сервера, мониторинга очередей системы.
3. SBTServer.exe – представляет собой Windows-приложение и предназначено для работы с объектами сервера интеграции. Позволяет настраивать Сервисы Захвата, Сервисы Отправки, конфигурировать ход обработки Документа и т.п.
4. SBTM_Scheduler.exe – многопоточное приложение, которое реализует функциональность SBTServer.
5. База данных SBT_DB.mdb – предназначена для хранения непосредственно данных всего программного комплекса в случае выбора СУБД MS Access.
6. База данных SBT_DB.mdf – предназначена для хранения непосредственно данных всего программного комплекса в случае выбора СУБД MS SQL Server.
7. SBTM_Integration.dll – сборка .NET. Содержит открытые интерфейсы для интеграции модулей, расширяющих функциональность приложения.
8. SBTM_Data.dll – сборка .NET, предназначенная для представления информации, хранящейся в БД в объектном виде. Инкапсулирует в себе все методы получения, создания, редактирования и удаления объектов системы через работу с базой данных.

Исходя из структуры предметной области и модели интеграции, можно выделить ряд областей:

- EventLog – регистрация событий и ошибок
- SBTM_Objects – работа с объектами документооборота
- SBTM_Queues – работа с очередями документов

Каждому из модулей, а также каждой из областей предметной области посвящены отдельные разделы с более подробным описанием [17].

На рис. 4 показана схема взаимодействия компонентов приложения SBTServer.

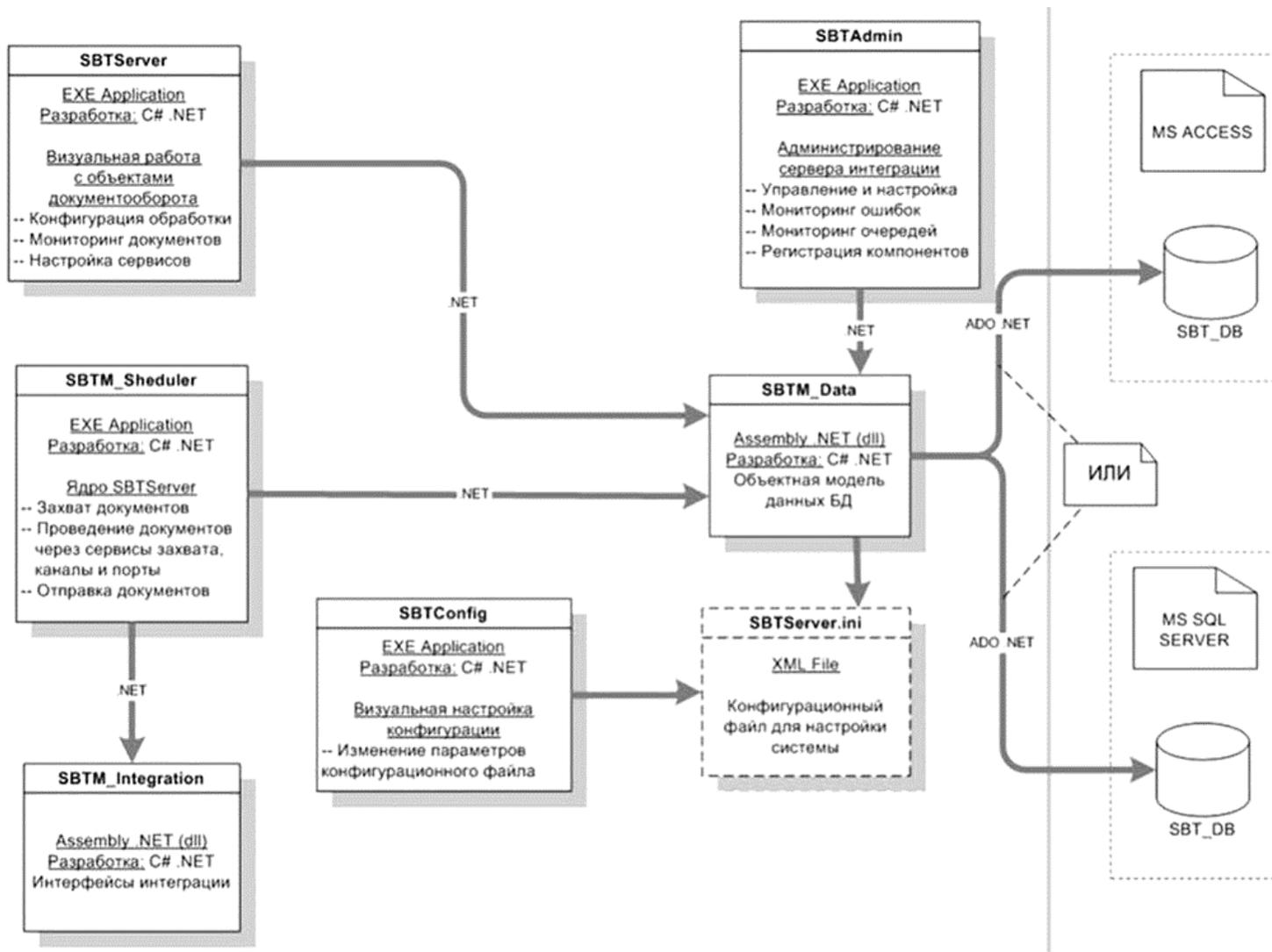


Рис. 4. Схема взаимодействия компонентов

Интерфейсы интеграции

Интерфейсы интеграции должны реализовывать компоненты интеграции. Все интерфейсы собраны в одной сборке .NET – SBTM_Integration.dll. Всего на данный момент предлагается пять основных интерфейсов интеграции:

- *ISBTM_Adjust*: Интерфейс настройки параметров компонента интеграции.
 1. Метод `string Adjust(string OldParams)` – настройка компонента, обычно вызов диалога для настройки параметров, определяемых каждым компонентом отдельно. `OldParams` – строка с предыдущими параметрами
 2. Метод `string SetParams(string Params)` – установка необходимых параметров перед работой компонента. Компонент должен вернуть пустую строку при успешной настройке и описание ошибки при неуспешной настройке. `Params` - строка с параметрами.
- *ISBTM_Send*: Интерфейс для компонентов отправки документов.
 1. Метод `string SendDocument(string Doc, string Name)` – отправка документа, компонент должен вернуть пустую строку при успешной отправке и описание ошибки – при неуспешной. `Doc` – документ для отправки, `Name` – имя документа.
- *ISBTM_Receive*: Интерфейс для компонентов захвата документов.
 1. Метод `string ReceiveDocument(out string Doc, string Name)` – захват документа, компонент должен вернуть через параметр `Doc` полученный документ и пустую строку при успешном получении, и описание ошибки – при неуспешном. `Doc` – полученный документ (`null`, `Nothing` в VB, если документов нет), `Name` – имя документа
- *ArrivedDocumentEventHandler*: Делегат для события прихода документа.
 1. `delegate void ArrivedDocumentEventHandler(string Doc, string Name)` – делегат для события прихода документа. `Doc` - пришедший документ, `Name` – имя документа.
- *ISBTM_ActiveReceive*: Интерфейс компонентов активного захвата документов.
 1. Событие `event ArrivedDocumentEventHandler ArrivedDocument` – событие прихода документа. Компонент сам отслеживает приход документа и при получении документа возбуждает событие.
 2. Метод `void Start()` – начать работу компонента.
 3. Метод `void Stop()` – остановить работу компонента

– *ISBTM_Change*: Интерфейс для дополнительных компонентов (шифровка, архивация, сжатие, преобразование и т.д.).

1. Метод `string ChangeDocument(string Doc, out string ChangedDoc)` – изменение документа, компонент должен вернуть через параметр `ChangedDoc` измененный (преобразованный) документ и пустую строку при успешном выполнении, и описание ошибки – при неуспешном. `Doc` – изменяемый документ, `ChangedDoc` – измененный документ

Компоненты интеграции

Разработано восемь компонентов интеграции: два компонента реализуют прием и отправку документов как файлов в файловой системе, еще два работают с очередями сообщений Microsoft MSMQ, пара компонентов предназначена для отправки и получения фалов по протоколу FTP и два последних принимают и отправляют документы через E-mail.

На рис. 5 изображены компоненты интеграции и реализуемые ими интерфейсы интеграции. Рассмотрим каждый из компонентов более подробно.

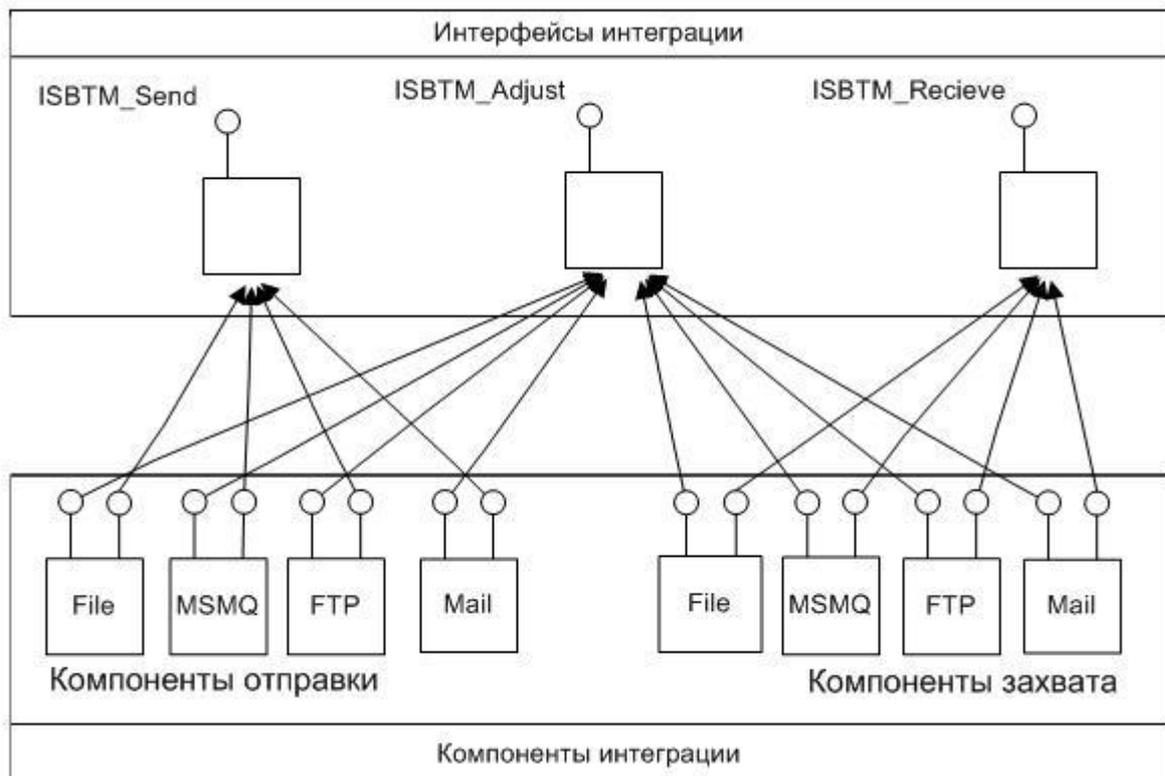


Рис. 5. Компоненты интеграции

Компонент захвата документов по протоколу POP3 MailReceive

Компонент реализует два интерфейса интеграции: ISBTM_Adjust и ISBTM_Receive. При настройке компонента вызывается форма для указания адреса сервера, имени учетной записи и пароля, письма от какого адресата и с какой темой получать.

Во время работы компонент получает письма по протоколу SMTP с указанного почтового сервера и передает их системе. Захваченные файлы удаляются с почтового сервера. Компонент просматривает только тело письма и не работает с вложениями.

Во время работы компонента могут произойти следующие ошибки:

- компонент не настроен;
- указанного почтового сервера не существует или он не доступен;
- неверно указанный пользователь или пароль;
- внутренние ошибки компонента.

Компонент отправки документов по протоколу SMTP MailSend

Компонент реализует два интерфейса интеграции: ISBTM_Adjust и ISBTM_Send. При настройке компонента вызывается форма для указания адреса сервера, письма от какого адресата, какому адресату и с какой темой отправлять.

Во время работы компонент создает письмо с полученным документом в качестве тела сообщения и отправляет его адресату.

Во время работы компонента могут произойти следующие ошибки:

- компонент не настроен;
- указанного почтового сервера не существует или он не доступен;
- внутренние ошибки компонента.

Компонент захвата документов из директории файловой системы FileReceive

Компонент реализует два интерфейса интеграции: ISBTM_Adjust и ISBTM_Receive. При настройке компонента вызывается форма для выбора директории в файловой системе. При этом имеется возможность создать новую директорию. Так же необходимо указать шаблон имени файлов для захвата.

Во время работы компонент извлекает файлы из указанной директории и передает их системе. Захваченные файлы удаляются из директории. Компонент просматривает только файлы и не работает с поддиректориями.

Во время работы компонента могут произойти следующие ошибки:

- компонент не настроен;
- указанной директории не существует;

- невозможно получить монопольный доступ к файлу;
- другие ошибки при работе с объектами файловой системы;
- внутренние ошибки компонента.

Компонент отправки документов в директорию файловой системы FileSend

Компонент реализует два интерфейса интеграции: ISBTM_Adjust и ISBTM_Send. При настройке компонента вызывается форма для выбора директории в файловой системе. При этом имеется возможность создать новую директорию. Так же необходимо указать, с каким расширением необходимо создавать файлы.

Во время работы компонент создает файл с полученным документом в указанной директории. Документ сохраняется под именем, с которым он пришел в систему. В случае, если это имя оказывается не уникальным или имя не присвоено, то генерируются уникальные имена по средствам Visual Studio .NET с попыткой минимальных изменений относительно исходного имени Документа.

Во время работы компонента могут произойти следующие ошибки:

- компонент не настроен;
- указанной директории не существует;
- другие ошибки при работе с объектами файловой системы;
- внутренние ошибки компонента.

Компонент захвата документов из директории файлового сервера по протоколу FTP FTPReceive

Компонент реализует два интерфейса интеграции: ISBTM_Adjust и ISBTM_Receive. При настройке компонента вызывается форма для указания имени сервера, имени учетной записи, пароля и имени директории, которую необходимо просматривать. Так же необходимо указать шаблон имени файлов для захвата.

Во время работы компонент извлекает файлы из указанной директории и передает их системе. Захваченные файлы удаляются из директории. Компонент просматривает только файлы и не работает с поддиректориями.

Во время работы компонента могут произойти следующие ошибки:

- компонент не настроен;
- указанного файлового сервера не существует или он не доступен;
- неверно указанный пользователь или пароль;
- указанной директории не существует;
- невозможно получить монопольный доступ к файлу;
- внутренние ошибки компонента.

Компонент отправки документов в директорию файлового сервера по протоколу FTP FTPSend

Компонент реализует два интерфейса интеграции: ISBTM_Adjust и ISBTM_Send. При настройке компонента вызывается форма для указания имени сервера, имени учетной записи, пароля и имени директории, в которую необходимо отправлять документы. Так же необходимо указать, с каким расширением необходимо создавать файлы.

Во время работы компонент создает файл с полученным документом в указанной директории. Документ сохраняется под именем, с которым он пришел в систему. В случае, если это имя оказывается не уникальным или имя не присвоено, то генерируются уникальные имена по средствам Visual Studio .NET с попыткой минимальных изменений относительно исходного имени Документа.

Во время работы компонента могут произойти следующие ошибки:

- компонент не настроен;
- указанного файлового сервера не существует или он не доступен;
- неверно указанный пользователь или пароль;
- указанной директории не существует;
- другие ошибки при работе с объектами файловой системы;
- внутренние ошибки компонента.

Компонент захвата документов из очереди сообщений Microsoft MSMQReceive

Компонент реализует два интерфейса интеграции: ISBTM_Adjust и ISBTM_Receive. При настройке компонента вызывается форма для выбора или указания имени очереди сообщений Microsoft, которую необходимо просматривать, и время ожидания появления сообщения в этой очереди, если она пуста. Применение компонента возможно только в ОС MS Windows, поддерживающих работу с очередями сообщений Microsoft.

Во время работы компонент читает сообщения из указанной очереди и извлекает их оттуда, тело извлеченного сообщения передает системе.

Во время работы компонента могут произойти следующие ошибки:

- компонент не настроен;
- указанной очереди не существует или она не доступна;
- другие ошибки при работе с объектами очередей сообщений;
- внутренние ошибки компонента.

Компонент отправки документов в очередь сообщений Microsoft MSMQSend

Компонент реализует два интерфейса интеграции: ISBTM_Adjust и ISBTM_Send. При настройке компонента вызывается форма для выбора или

указания имени очереди сообщений Microsoft, в которую необходимо отправлять документы. Применение компонента возможно только в ОС MS Windows, поддерживающих работу с очередями сообщений Microsoft.

Во время работы компонент создает сообщение, телом которого является полученный для отправки документ, и отправляет это сообщение в указанную очередь.

Во время работы компонента могут произойти следующие ошибки:

- компонент не настроен;
- указанной очереди не существует или она не доступна;
- другие ошибки при работе с объектами очередей сообщений;
- внутренние ошибки компонента.

База данных для хранения информации SBTServer

На основе UML-диаграмм (рис. 6-7) разработана концептуальная и физическая модель базы данных системы.

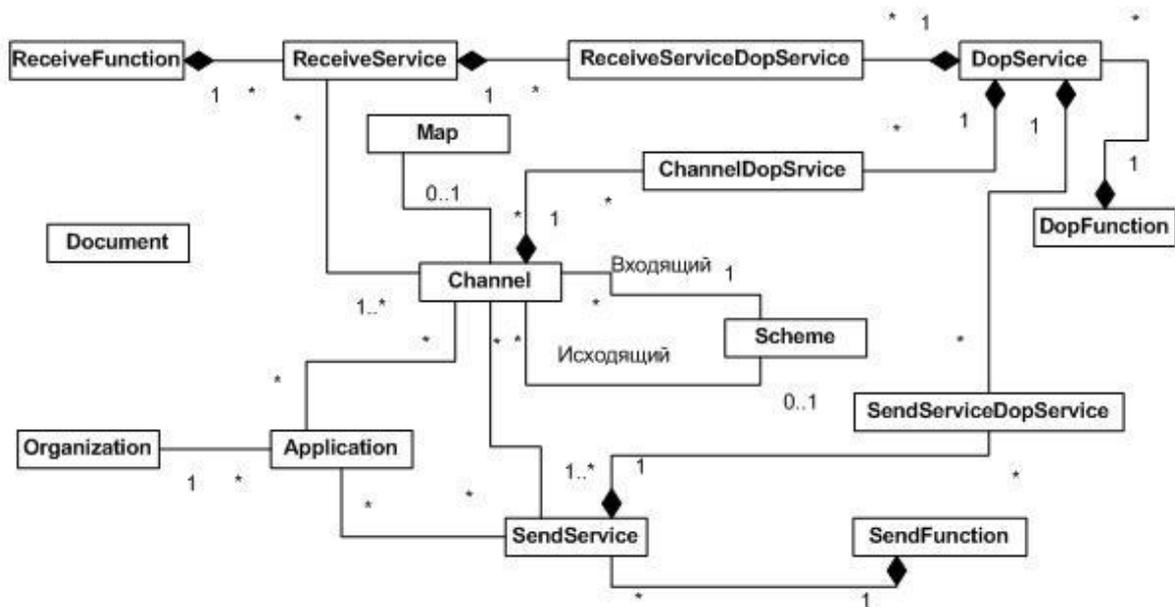


Рис. 6. UML-диаграмма классов объектов документооборота

Написан SQL-скрипт для создания всех таблиц и связей между ними.

Каждая из таблиц может быть отнесена к определенной области, перечисленных в конце раздела. Далее, когда каждая из областей будет рассмотрена более подробно, будет указан список таблиц, который входит в каждую из областей. В целом, уже из названия таблиц можно догадаться, что каждая из них представляет [16].

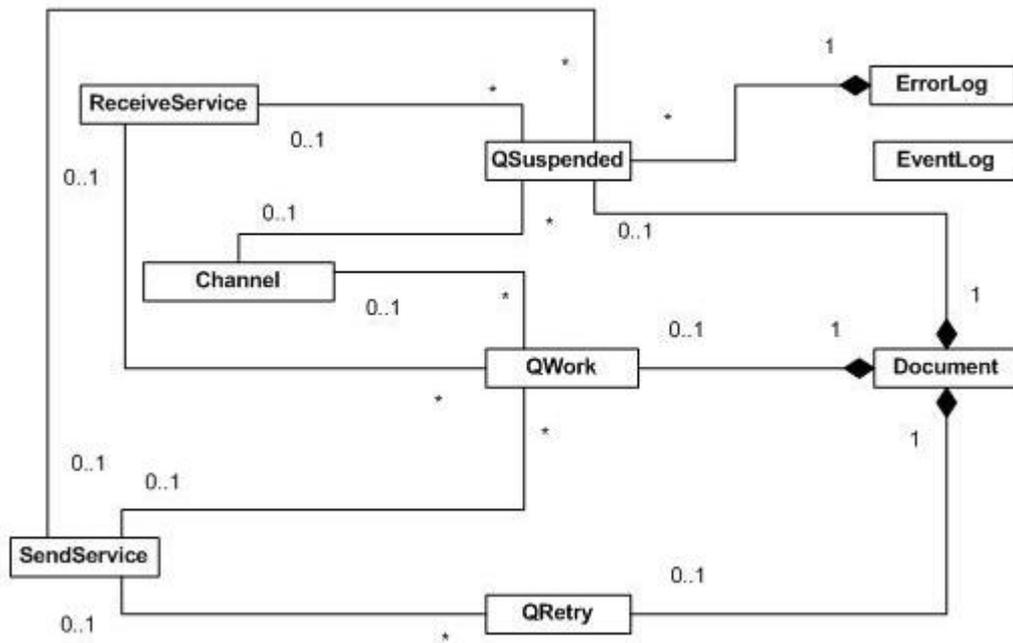


Рис. 7. UML-диаграмма классов очередей, событий и ошибок

Регистрация событий и ошибок SBTM_EventLog

Объекты данной области предоставляют высокоуровневый способ фиксации (занесения в журналы) основных событий и ошибок системы.

Каждый объект любой области обязан заносить ошибки и события, произошедшие во время выполнения его методов в базу данных. Для этого разработан отдельный класс – EventLog. Существует ряд стандартных ошибок, которые необходимо журнализировать. Особое место занимает ошибка с №1000 – внутренняя ошибка, она подразумевает целый класс ошибок, которые могут произойти во время работы с базой данных или в случае попытки неправильно конвертировать типы базы данных в типы .NET.

База данных регистрации событий и ошибок

В базе данных эта область представлена двумя таблицами:

- ErrorLog – таблица, представляющая журнал ошибок,
- EventLog – таблица, представляющая журнал событий.

Объектная модель событий и ошибок

Так как заполнение базы данных этими объектами возложено на приложение SBTM_Scheduler.exe, то в рамках приложения SBTAdmin.exe запрещено создание объектов данного типа. В данной области используется только один класс EventLog, имеющий единственный метод:

- Метод `int SendEvent (TypeEventLog TypeEvent, string Label, string Source, int Number, string Description)` – фиксация события или ошибки. Результат

при успешном завершении равен идентификатору записи в таблице. TypeEvent – тип произошедшего события, Label – краткое описание события – строка с длиной не более 255 символов, Source – источник события – строка с длиной не более 100 символов, Number – номер события\ошибки, Description – подробное описание события – строка с длиной не более 65536 символов.

И перечисление TypeEventLog

```
enum TypeEventLog
```

```
ErrorEvent = 1 – Фиксация ошибки,
```

```
WorkEvent = 2 – Фиксация события
```

Работа с объектами документооборота SBTM_Objects

Объекты данной области предоставляют доступ к основным объектам документооборота, хранящихся в базе данных. Основными объектами системы документооборота являются:

- *Организация* – содержит в себе множество приложений и служит для кластеризации приложений.
- *Приложение* – делит каналы и сервисы отправки по принадлежности какому-либо приложению, которое в свою очередь принадлежит определенной организации. При этом, разработанное приложение для визуальной работы с объектами сервера интеграции SBTServer.exe позволяет узнать, какие приложения общаются между собой.
- *Документ* – непосредственно документ, попавший в систему.
- *Схема документа* – XML-схема описания документа, используемая для проверки действительности документа поступающего и обрабатываемого в системе. Содержит описание полей, типов данных, обязательности, значений по умолчанию и других дополнительных атрибутов [33].
- *Карта преобразования* – XSL-схема преобразования XML-документов, описывает на языке XSL, как из одного документа получить другой в соответствии с XML-схемами документов [34].
- *Канал* – предназначен для проверки, обработки поступающего на вход документа и при необходимости преобразования его в другой XML-формат.
- *Функции захвата* – описывают класс, используя объекты которого можно получить документ по определенному протоколу. Причем функции могут быть активными и пассивными.
- *Сервисы захвата* – описывают периодичность захвата документов через определенную функцию и параметры захвата. Полученный документ

направляется в соответствующие каналы.

- *Функции отправки* – описывают класс, используя объекты которого можно отправить документ по определенному протоколу.
- *Сервис отправки* – предназначен для отправки документа через функции интеграции отправки по определенному данной функцией протоколу, а также параметры отправки.
- *Дополнительные функции* – описывают класс, используя объекты которого можно преобразовать документ. К преобразованиям относятся шифровка, архивация, сжатие и т.д.
- *Дополнительные сервисы* – предназначены для преобразования документов с помощью дополнительных функций интеграции с параметрами для этих функций

База данных

В базе данных эта область представлена следующими таблицами:

- Organization – таблица, представляющая организации,
- Application – таблица, представляющая приложения,
- Document – таблица, представляющая документы,
- Scheme – таблица, представляющая схемы документов,
- Map – таблица, представляющая карты преобразования,
- Channel – таблица, представляющая каналы,
- ReceiveFunction – таблица, представляющая функции захвата (активные и пассивные),
- ReceiveService – таблица, представляющая сервисы захвата,
- SendFunction – таблица, представляющая функции отправки,
- SendService – таблица, представляющая сервисы отправки,
- DopFunction – таблица, представляющая дополнительные функции,
- DopService – таблица, представляющая дополнительные сервисы,
- Channel_Application и SendService_Application – таблицы, представляющие связь между каналами и приложениями и между сервисами отправки и приложениями соответственно,
- ChannelDopService, SendServiceDopService, ReceiveServiceDopService – таблицы, представляющие связь между дополнительными сервисами и каналами, сервисами отправки и сервисами захвата соответственно,
- ReceiveService_Channel, Channel_SendService – таблица, представляющие связь между сервисами захвата и каналами и между каналами и сервисами отправки.

Объектная модель

Для каждого объекта документооборота разработан класс коллекции объектов и класс самого объекта. Для всех классов разработан однотипный набор функций. Так как заполнение базы данных этими объектами возложено на приложения SBTServer.exe и SBTAdmin.exe, то приложения SBTM_Sheduler.exe использует только методы классов для чтения из БД. Ниже перечислены наиболее общие свойства и методы объектов системы.

Объекты:

- *Свойства для чтения и записи* – свойства для доступа к атрибутам объектов. У всех объектов есть свойства:
 - **Id** – Идентификатор (только чтение),
 - **Name** – Наименование,
 - **Description** - Описание.
- *Замещенный метод ToString()* – представление информации об объекте в строковом виде.

Коллекции объектов:

- *Свойство First* – возвращает объект с наименьшим идентификатором,
- *Свойство Next* – возвращает следующий объект, учитывает предыдущие вызовы First, Next и Get
- *Метод Exist* – проверка на существование объекта с заданным идентификатором или именем. Результат True, если объект существует, иначе False, в случае ошибки так же возвращается False
- *Метод Get* – возвращает объект с заданным идентификатором или именем, если объект существует, иначе null, в случае ошибки так же возвращается null.
- *Метод Add* – вставка нового объекта в коллекцию объектов, результат при успешном выполнении равен идентификатору объекта, иначе -1.
- *Метод Read* – чтение объекта из базы данных с заданным идентификатором, результат при успешном выполнении равен ссылке на объект, иначе null.
- *Метод Remove* – удаление объекта с заданным идентификатором из коллекции объектов и из базы данных, результат при успешном выполнении равен ссылке на объект, иначе null.

Документ:

- *Метод ChangeDocument* – изменение значения тела документа. Результат при успешном выполнении равен 0, при неуспешном – № ошибки.

Сервис захвата, канал и сервис отправки:

- *Метод TurnOff* – делает заданный объект неактивным. Теперь документы не будут проходить через этот объект.
- *Метод TurnOn* – делает заданный объект активным. Если до этого объект был неактивным, то теперь документы будут вновь проходить обработку в этом объекте.

***Работа с очередями документов системы
SBTM_Queues***

Объекты данной области предназначены для хранения документов, проходящих через сервер, при этом необходимо фиксировать расположение документа. Также необходимо хранение документов, приостановленных по разным причинам: вследствие ошибки, несоответствия схеме документа, невозможности преобразования, невозможности отправки по конкретному протоколу и т.д.

Всего в системе существует три типа очередей и, соответственно, три типа элементов этих очередей:

1. *Рабочая очередь* – очередь обрабатываемых документов. В очередь документы ставятся сразу, как только попадают в систему, через компонент захвата. В рабочей очереди документ может находиться в трех состояниях: 1) Захват; 2) Канал; 3) Отправка.

2. *Очередь повторов отправки*. Эта очередь предназначена для документов, находящихся в сервисах отправки. Для сервиса отправки можно настроить количество попыток отправки и временной интервал между повторами попытки. Необходимость данной возможности повторов обуславливается тем, что некоторые узлы сети могут оказаться недоступными или может произойти сбой в сети при отправке документа. Кроме того, предусмотрены вторичные параметры для настройки функции отправки, если не удастся отправить документ в случае настройки первичными параметрами.

3. *Очередь приостановленных документов*. В данную очередь попадают документы по следующим причинам:

- Внутренняя ошибка при обработке документа на сервере.
- Внутренняя ошибка при обработке документа в компонентах интеграции.
- Документ не является правильным или действительным (не соответствует указанной схеме документа).
- Схемы документов или схема их преобразования не являются правильными или не доступны.
- Истекло количество повторов отправки.

- Ошибки при работе с базой данных.
- И другие ошибки возможные при неправильной настройке объектов документооборота.

В очереди приостановленных документов документ может находиться в трех состояниях: 1) Захват; 2) Канал; 3) Отправка.

База данных

В базе данных эта область представлена следующими таблицами:

- QWork_ReceiveService, QWork_Channel, QWork_SendService – таблицы, представляющие документы, находящиеся в рабочей очереди с состояниями захват, канал и отправка соответственно.
- QRetry_SendService – таблица, представляющая документы, находящиеся в очереди повторов отправки.
- QSuspended_ReceiveService, QSuspended_Channel, QSuspended_SendService – таблицы, представляющие документы, находящиеся в очереди приостановленных документов с состояниями захват, канал и отправка соответственно.

Объектная модель

Для каждой очереди и элемента очереди разработаны классы, содержащие однотипный набор функций. Так как заполнение базы данных этими объектами возложено на приложение SBTM_Sheduler.exe, то в рамках приложений SBTServer.exe и SBTAdmin.exe запрещено создание объектов данного типа:

Элемент очереди:

- *Свойства для чтения и записи* – свойства для доступа к атрибутам элемента очереди. У всех элементов очереди есть свойства:
 - **Id** – Идентификатор (только чтение),
 - **DateTime** – Время постановки в очередь (только чтение),
- *Свойство Next* – возвращает следующий за ним элемент очереди.
- *Замещенный метод ToString()* – представление информации об элементе очереди в строковом виде.
- *Метод ChangeState()* – Изменяет состояние документа в очереди (только для элементов рабочей очереди).
- *Метод ChangeLastRetry()* – Изменяет значения последней попытки отправки документа (только для элементов очереди повторов).

Очереди:

- *Метод Push()* – постановка нового элемента в очередь, результат при успешном выполнении равен идентификатору объекта, иначе -1.

- *Метод Peek()* – чтение первого элемента очереди или элемента очереди с заданным идентификатором, результат при успешном выполнении равен элементу очереди, иначе null.
- *Метод Pop()* – извлечение первого элемента из очереди или элемента с заданным идентификатором, результат при успешном выполнении равен элементу, иначе null.

Перечисление TypeStateDoc - тип состояния документа в очереди

enum TypeStateDoc

stdReceive = 1 – в сервисе захвата,

stdChannel = 2 – в канале,

stdSend = 3 – в сервисе отправки.

Приложение для визуальной работы с объектами документооборота SBTServer.exe

Приложение создано с использованием Visual Studio .NET , главная форма которого представлена на рис. 8.

Для добавления, удаления и редактирования объектов документооборота используются однотипные формы-Мастера, похожие на ранее сгенерированные формы.

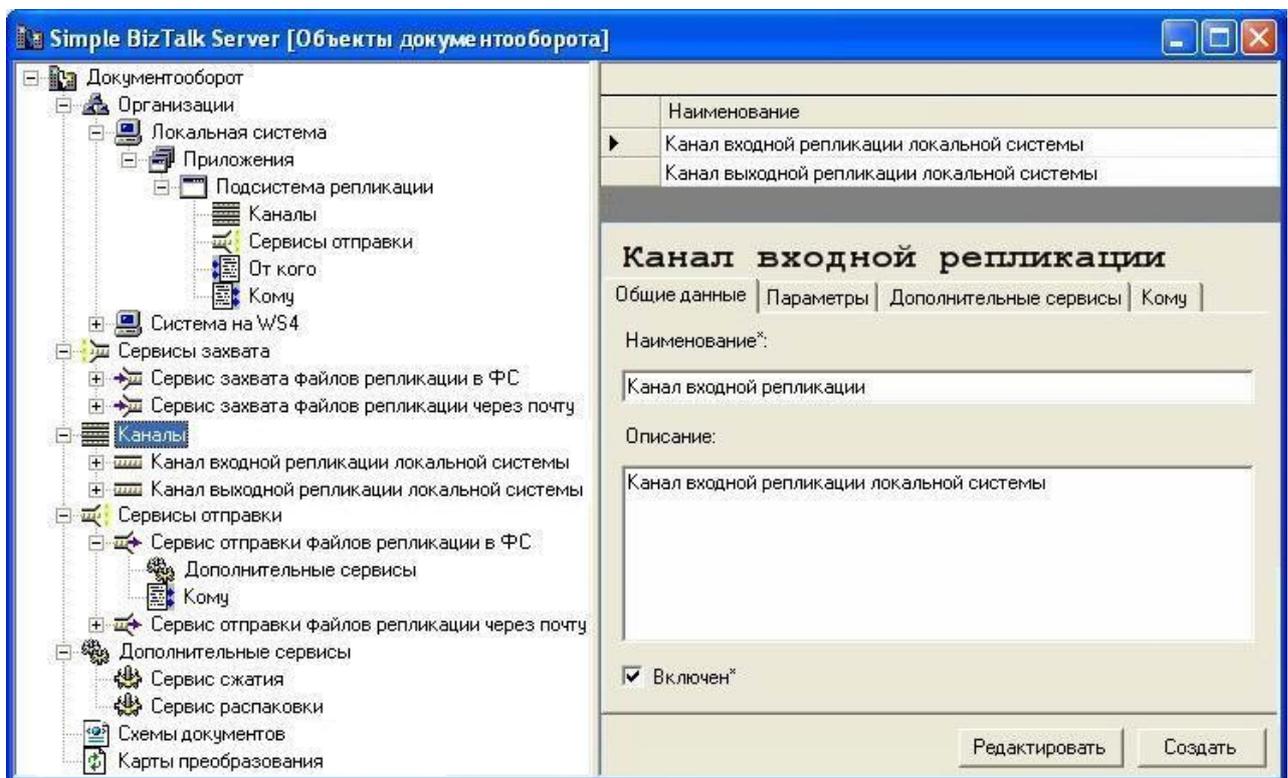


Рис. 8. Объекты документооборота

В левой части расположено дерево объектов документооборота, в правой части вверху – краткая информация об объекте, выделенном в дереве объектов, а в правой нижней части в виде вкладок помещается детальная информация об объекте, выделенном в верхней части. Информация на вкладках доступна только для чтения, если нужно отредактировать значения каких-либо параметров, то необходимо нажать на кнопку «Редактировать». Если необходимо создать выделенный объект системы, то для этого существует кнопка «Создать». Как в первом, так и во втором случае открывается отдельная форма.

Редактирование и добавление осуществляется при помощи мастера. Мастер реализован как набор вкладок (рис. 9).



Рис. 9. Редактирование канала

Приложение для администрирования сервера SBTAdmin.exe

На каждой вкладке сгруппированы поля для заполнения свойств объекта. Обязательные поля помечены звездочкой (*).

Изначально при проектировании интерфейса при помощи CASE-инструментария METAS 2.0. приложение для администрирования было совмещено с приложением для визуальной работы с объектами документооборота. После того, как приложения были переписаны полностью с использованием Visual Studio .NET, они были разделены на два приложения, работающих с одной базой данных.

Приложение для администрирования сервера (рис. 10) предназначено для мониторинга очередей системы, регистрации новых компонентов интеграции,

просмотра журналов событий и ошибок и отслеживания обрабатываемых и архивных документов.

Консоль сервера содержит четыре основные группы:

- *Очереди документов.* Содержит три очереди, подробное описание которых было приведено выше. Так же, как и для объектов, по центру представлена краткая информация обо всех объектах, представленных выделенной вершиной, в правой части – детальная информация об объекте, выделенном в центральной части.
- *Компоненты интеграции.* Здесь можно зарегистрировать, изменить или удалить нужный компонент.
- *Журналы.* Содержит журнал ошибок и журнал сообщений.
- *Документы.* Документы, прошедшие обработку в системе или находящиеся в одной из очередей.

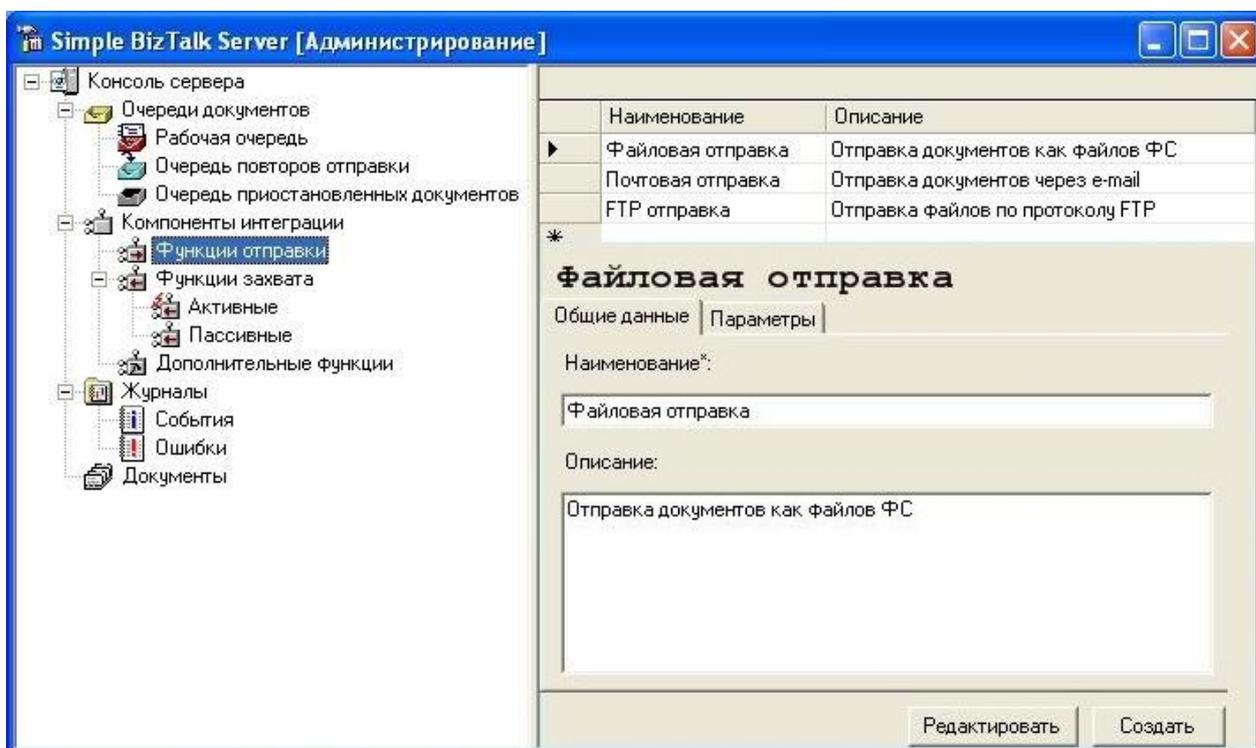


Рис. 10. Администрирование сервера

В левой части расположено дерево консоли сервера, в правой части вверху – краткая информация об объекте, выделенном в дереве слева, а в правой нижней части в виде закладок помещается детальная информация об объекте, выделенном в верхней части. Информация на закладках доступна только для чтения, если нужно отредактировать значения каких-либо параметров, то необходимо нажать на кнопку «Редактировать». Если необходимо создать выделенный объект системы, то для этого существует кнопка «Создать». Как в

первом, так и во втором случае открывается отдельная форма. Для редактирования и добавления доступны только компоненты интеграции. Элементы очереди приостановленных документов так же можно отредактировать. Остальные объекты консоли сервера доступны только для просмотра.

Ядро сервера SBTM_Scheduler.exe

В предыдущих главах было рассмотрено множество компонентов для обеспечения работы сервера и его расширяемости, а теперь рассмотрим само ядро сервера, которое управляет обработкой документов.

Ядро является *фоновым многопоточным приложением*, созданным в системе программирования Visual Studio .NET на языке С#. При запуске приложения на панели задач в области системных значков можно наблюдать значок ядра сервера. Мигающая красная точка в левом верхнем углу значка говорит о нормальном функционировании сервера:



– сервер запущен,



– сервер приостановлен,



– сервер остановлен.

Для предотвращения конфликтных ситуаций и для того, чтобы избежать утери документов, необходимо запретить запуск второй копии данного приложения. Общий подход решения этой проблемы такой: в приложении создаем любой именованный объект ядра (например, Mutex), доступ к которому можно получить из любого процесса в Windows по его имени, например. Причем при создании такого объекта можно определить действительно ли он был создан только что вызванной функцией или мы уже имеем дело с ранее созданным экземпляром. Кроме того, при попытке запустить вторую копию приложения будет активизирована главная форма уже запущенного приложения, что будет вызывать меньше вопросов, чем если бы приложение просто не запускалось или выдавалось сообщение об уже запущенной копии приложения.

Сервер построен как *многопоточное приложение*. Потоки захвата (активного и пассивного) постоянно запущены в системе, по мере поступления документов запускаются потоки, проводящие документ через сервис захвата, канал, сервис отправки, и потоки отправки документов. На рис. 11 представлена структура ядра сервера.

Поток главного окна отвечает за создание всех диспетчеров во время загрузки приложения, прием оконных сообщений и распространение сообщений старта, паузы, возобновления и остановки всем диспетчерам.

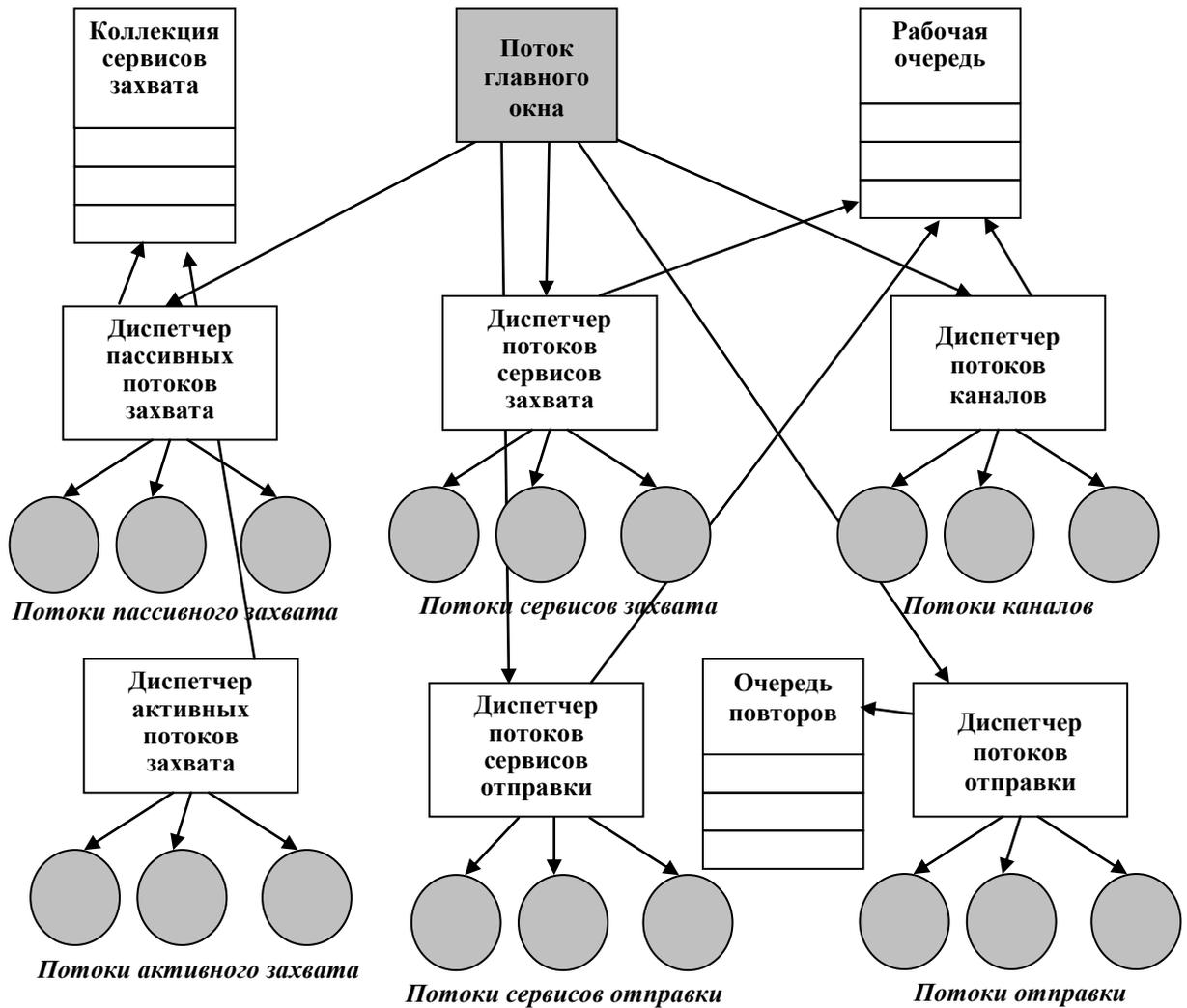


Рис. 11. Структура сервера

Диспетчеры отвечают за создание потоков, уничтожение выполнившихся потоков и распространение сообщений старта, паузы, возобновления и остановки всем потокам. Сканируют очереди и коллекции, отмеченные на рисунке.

Потоки активного захвата создают объекты интеграции функций активного захвата, передают параметры для настройки и вызывают метод Start, предварительно подписавшись на событие приема документа. При возникновении такого события запускается отдельный поток Windows и полученный документ помещается в рабочую очередь документов с состоянием «Захват».

Потоки пассивного захвата выполняются с заданной периодичностью, вызывая соответствующий компонент интеграции для захвата документа.

Полученный документ помещается в рабочую очередь документов с состоянием «Захват».

Потоки сервисов захвата выполняются по заданию диспетчера, получив от него элемент рабочей очереди с состоянием «Захват». Проводят документ через все дополнительные сервисы сервиса захвата и изменяют состояние элемента на «Канал».

Потоки каналов выполняются по заданию диспетчера, получив от него элемент рабочей очереди с состоянием «канал». Проводят документ через все дополнительные сервисы канала. Документ проверяется, преобразуется и в соответствии с количеством портов для канала располагается в рабочей очереди с состояниями «Отправка».

Потоки сервисов отправки выполняются по заданию диспетчера, получив от него элемент рабочей очереди с состоянием «Отправка». Проводят документ через все дополнительные сервисы сервиса отправки. Помещают документ в очередь повторов со значением попыток отправки равным 0.

Потоки отправки выполняются по заданию диспетчера, получив от него элемент очереди повторов. Пытаются отправить документ, используя объект интеграции – функцию отправки. Если количество попыток исчерпано, элемент помещается в очередь приостановленных документов, если же отправка прошла успешно, то элемент очереди удаляется из системы, а Документ попадает в архив прошедших обработку документов.

Все классы потоков и диспетчеров имеют абстрактный *класс-предок*, который обеспечивает базовую функциональность дочерних классов. Для синхронизации потоков используется критическая секция. Таким образом, не возникает ошибок и конфликтов, если два или более потоков готовы ввести в систему документ или зарегистрировать ошибку. Кроме того, в силу ограничений СУБД Access на транзакции, а именно: невозможен одновременный запуск нескольких транзакций, критическая секция используется для запрещения запуска транзакций, если в системе уже запущена одна из них.

Таким образом, при обработке документа на разных стадиях, различные потоки между собой непосредственно не взаимодействуют, а только через очереди документов. Данный подход увеличивает время обработки документа, но в то же время значительно снижается риск потери документа при его обработке. Кроме того, в любое время можно просмотреть элементы очереди, используя приложение для администрирования сервера.

Работа ядра сервера организована таким образом, что документ при обработке его на сервере потеряться в случае непредвиденных сбоев или принудительного останова сервера не может. Так как при обработке документов

используются транзакции, то в случае сбоя система откатится назад до последнего установленного состояния.

Все ошибки при работе ядра сервера фиксируются в журнале ошибок. Если ошибки происходят при обработке документа, то документ перемещается в очередь приостановленных документов со ссылкой на эту ошибку.

2.3. Применяемые программные средства

Инструментальным средством разработки является среда программирования Visual Studio .NET 2003. Информационное взаимодействие, интеграция ИС осуществляется на основе технологии XML [26, 36]. Основой для разработки послужили открытые инициативы SOAP[28], BizTalk.

Выбор операционной системы

Для обеспечения работы сервера были выбраны ОС MS Windows (начиная с версии 98 и NT 4.0) в силу их распространенности и широкой используемости, а также, потому что .NET на данный момент поддерживает только ОС данного семейства.

Выбор компонентной технологии

На сегодняшний день реализация открытой системы немыслима без использования компонентных технологий. В компоненте инкапсулируется его функциональность. Через предоставление внешнего набора интерфейсов обеспечивается сборка компонентов в один программный комплекс, состоящий из многих из функциональных блоков.

Сейчас наиболее законченными и распространенными являются следующие технологии:

- COM – компонентная модель объектов, является детищем Microsoft и поддерживается только на ОС класса MS Windows.
- CORBA – открытая спецификация, разработанная группой поддержки объектной технологии OMG (Object Management Group). Для ее поддержки необходимо установить соответствующее программное обеспечение. Поддерживается большинством распространенных ОС.
- JB – Java Beans – компонентная технология от Java, является открытой спецификацией. Для работы необходима виртуальная Java-машина. Спектр поддерживаемых ОС также широк. Java Beans ориентирован в основном на графические компоненты.
- EJB – Enterprise Java Beans, продолжение развития Java Beans, отличие в том, что это уже корпоративные компоненты, в которые

инкапсулирована логика приложений и данные, а не только графический интерфейс

- .NET Assemblies – сборки .NET – одно из последних достижений Microsoft. Поддерживается только семейством ОС Windows, начиная с версии 98, с установленным .NET Framework.

По следующим причинам выбор остановлен на технологии .NET:

- Для CORBA и JB необходимо соответствующее программное обеспечение, которое требует дополнительных расходов и ресурсов.
- Java – интерпретирующая система, следовательно, дополнительные издержки ресурсов.
- Метаданные .NET являются значительным усовершенствованием по сравнению с классической информацией о типах COM. Метаданные описывают не только типы, используемые в сборке, но и саму сборку.
- Усовершенствованная работа с версиями одного и того же модуля
- Повышенная надежность и безопасность.

Выбор системы программирования

Единственной системой программирования, поддерживающей технологию .NET является Microsoft Visual Studio .NET. Для разработки выбран язык C#. C# – это фактически гибрид разных языков. При этом C# синтаксически не менее (если не более) чист, чем Java, так же прост, как Visual Basic, и обладает практически той же мощностью и гибкостью, что и C++ [27]. Все компоненты: непосредственно сам сервер, приложение для его администрирования и конфигурации реализованы по средствам C#. Изначально структура БД была разработана при помощи инструментария MDK Suite 2.0. Однако, в силу того, что этот инструментарий продолжает разрабатываться, в нем не хватает функционала, необходимой для полной реализации возможностей сервера интеграции. Таким образом, расширив существующую структуру БД и переписав Приложения по визуальной работе с сервером интеграции, были решены большинство проблем, возникших при разработке этих приложений при помощи CASE-средства METAS [21].

Выбор СУБД и метода доступа к данным

Для работы приложения необходимо хранить различную информацию в БД. Это очереди входящих, исходящих, обрабатываемых, отвергнутых документов; также журналы и описание всех объектов BizTalk Server.

По той причине, что планируется работа на простых ОС класса рабочих станций, а главным образом обеспечения настраиваемости и расширяемости, в

частности, за счет смены СУБД, желательно обеспечить возможность выбора между СУБД.

Под ОС MS Windows распространены следующие технологии:

- ODBC (Open Database Connectivity) – открытый низкоуровневый интерфейс для работы с реляционными данными.
- OLE DB – открытый низкоуровневый интерфейс, построенный в соответствии со спецификациями ODBC. OLE DB определяет открытый стандарт доступа к любым данным, причем как к реляционным, так и к не реляционным источникам информации (в отличие от ODBC).
- DAO – предоставляет модель объектов для доступа к локальным базам данных или базам данных SQL через Jet. DAO позволяет обращаться к базам данных Microsoft Jet (напрямую или через ODBC) и к таким источникам данных ISAM, как FoxPro, Paradox или Lotus. В сравнении с более новыми технологиями DAO работает довольно медленно.
- ADO (ActiveX Data Objects) – объектная модель данных на основе технологии COM, предоставляет модель объектов для доступа к данным любых типов через OLE DB.
- ADO .NET – это не просто перенос классической модели ADO на платформу .NET, но ее коренная переработка. Одним из важнейших отличий этой модели является полная поддержка представления данных в XML-совместимом формате.

Выбор сделан в пользу ADO .NET.

Изначально для работы выбрана СУБД MS Access, драйверы которой присутствуют в ОС, даже если сама Access не установлена. При выбранном подходе закладывается возможность смены СУБД. Смена СУБД заключается в смене провайдера, т.е. смене строки подключения, ну, и конечно, переносе схем данных на другую БД. На данном этапе разработки есть возможность выбора СУБД: помимо MS Access поддерживается работа с MS SQL Server [3].

2.4. Используемые технические средства и требования к аппаратуре

Требования к аппаратуре определяются требованиями используемого программного обеспечения.

Минимальные требования следующие: процессор Pentium, 64 Мб ОЗУ.

Для работы по протоколам электронной почты, FTP необходимо подключение к сети.

При передаче файлов на различных носителях требуются соответствующие устройства для чтения и записи файлов.

3. Специальные условия применения и требования организационного, технического и технологического характера

Для работы с программами нет необходимости создания специальных условий применения и выполнения особых требований организационного, технического и технологического характера. Условия применения и требования определяются требованиями к применяемому программному и аппаратному обеспечению, перечисленными выше, а также выполнением лицензионных соглашений при использовании необходимого для работы приложений программного обеспечения.

Администратор системы должен иметь опыт установки и администрирования ОС Microsoft Windows, реляционных СУБД.

4. Условия передачи программной документации или ее продажи

Продажа и передача программ и программной документации возможна на основе договора с АНО науки и образования «Институт компьютеринга». Условия передачи и/или продажи полностью определяются договором, заключаемым заказчиками/покупателями с АНО «Институт компьютеринга».

Представителем АНО «Институт компьютеринга», имеющим право заключать договоры на передачу и/или продажу программного обеспечения, на разработки программ с использованием представленных в данном документе средств, является заместитель директора Лядова Л.Н. (e-mail: lnlyadova@mail.ru; тел.: +7 (342) 222-37-95).

Библиографический список

1. *Агibalов Г.П., Скутин А.А.* Математическая модель и технология разработки безопасных корпоративных информационных систем // Электронный журнал «Исследовано в России». 2001. [<http://www.russiantechology.ru/151.htm>].
2. Анализ рынка решений интеграции среди фирм-разработчиков. [<http://www.tibco.com/software/applicationintegration/default.jsp?m=c1>].
3. *Артемов Д.В.* Microsoft SQL Server 2000. Новейшие технологии. М: «Русская редакция», 2001.
4. *Букавнев А.* Введение в JMS. Модели обмена сообщениями. [http://www.javable.com/columns/serv_side/workshop/14/].
5. *Галкин Г.* Интеграция приложений: история подходов // Сетевой журнал №6.2002. [<http://www.setevoi.ru/cgi-bin/text.pl/magazines/2002/6/20>].
6. *Галкин Г.* Мифы и парадигмы интеграции приложений // Электронная версия журнала «Intelligent Enterprise», №12-13(101)/2004. [<http://www.iemag.ru/?ID=495937>].
7. Интеграция корпоративных приложений: основные понятия. Перевод: Intersoft Lab. [<http://citcity.ru/11132/>].
8. Интеграция процессов и приложений. [http://www.i-teco.ru/pi_solutions/pi_ipa_ipp.html].
9. Как обмен сообщениями решает проблемы современного бизнеса. [<http://www.getinfo.ru/article207.html>].
10. *Колесов А.* Технологии BizTalk для управления XML-документами // Электронная версия статьи еженедельника «PC Week/RE», № 21/2000. [<http://www.visual.2000.ru/kolesov/pcweek/2000/00607bzt.htm>].
11. *Ланин В.В., Лукиных И.А., Лядова Л.Н.* Многоуровневая система подготовки, сбора и обработки отчетов для учреждений сферы образования // Материалы IV всероссийской научно-практической конференции «Образовательная среда сегодня и завтра» / М.: 2005. С.332-334.
12. *Ложечкин А.* Microsoft BizTalk Server 2000 как средство интеграции приложений и бизнес-процессов // Электронная версия журнала «ВУТЕ/Россия» №02/2001. [<http://www.bytemag.ru/Article.asp?ID=141>].
13. *Ложечкин А.* Microsoft BizTalk Server 2002 – второе пришествие // Электронная версия журнала «ВУТЕ/Россия» №05/2002. [<http://www.bytemag.ru/?ID=600873>].
14. *Ложечкин А.* Интеграция приложений электронной коммерции с использованием BizTalk Server. М.: Издательский дом «Русская редакция», 2001.
15. *Лукиных И.А.* BizTalk Server как средство интеграции распределенных баз данных // Материалы XLIII Международной научной студенческой конференции «Студент и научно-технический прогресс»: Информационные технологии / Новосибирск: НГУ, 2005. С. 322-323.
16. *Лукиных И.А.* Интеграция информационных систем учреждений образования на основе технологии BizTalk Server в информационной системе «Образование Пермской области» // Информатика в школе: Тезисы докладов X юбилейной областной научно-методической конференции 9-10 января 2006 г. «Рождественские чтения» / Пермь: Перм. ун-т. 2006. С.61-63.

17. *Лукиных И.А.* Информационное взаимодействие в распределенных системах на основе технологии BizTalk Server // Сборник научных статей участников конференции «Молодежь. Образование. Экономика». Часть 4 / Ярославль: «Ремдер», 2004 С.120-124.
18. *Лукиных И.А.* Интеграция информационных систем и распределенных баз данных на основе технологии BizTalk Framework // Спец. выпуск журнала «Известия Белорусской инженерной академии». Доклады IX Международной школы-семинара аспирантов, магистрантов и студентов «Современные информационные технологии». Республика Беларусь, 2006.
19. *Лукиных И.А.* Интеграция информационных систем на основе технологии BizTalk Server // Тезисы докладов конференции-конкурса работ студентов, аспирантов и молодых ученых «Технологии Microsoft в информатике и программировании» / Новосибирск: НГУ, 2004. С.25-27.
20. *Лукиных И.А.* Интеграция информационных систем с распределенными базами данных на основе BizTalk Server // Тезисы докладов конференции-конкурса работ студентов, аспирантов и молодых ученых «Технологии Microsoft в информатике и программировании» / Новосибирск: НГУ, 2006. С.27-28.
21. *Лядова Л.Н., Рыжков С.А.* CASE-технология METAS // Математика программных систем: Межвуз. сб. науч. тр. / Перм. ун-т. – Пермь, 2003. С.4-18.
22. *Лядова Л.Н.* Информационные системы в экономике и управлении. Лекции по теме «Корпоративные информационные системы». Пермь: Пермский ун-т, 2003.
23. *Мусаев Э.* Microsoft BizTalk Server 2004 – автоматизация документооборота и интеграция систем предприятия // Электронная версия журнала «ВУТЕ/Россия» №6/2004. [<http://www.bytemag.ru/Article.asp?ID=2773>].
24. *Новакова Н.* Microsoft BizTalk Server 2000 и интеграция бизнес-процессов. Особенности подсистем // Электронная версия журнала «Компьютер-Информ». [http://www.ci.ru/inform04_02/p_17micros.htm].
25. *Рыжков С.А.* Программное обеспечение интеграции приложений на основе технологии BizTalk Framework // Межвузовский сб. науч. трудов / Математика программных систем. Пермь: Перм. ун-т. 2002. С.45-59.
26. *Сандра С. Э.* XML: справочник. СПб: Питер, 2000.
27. *Троелсен Э.* С# и платформа .NET. Библиотека программиста. СПб.: Питер, 2002.
28. *Трэвис Б.* XML и SOAP: программирование для серверов BizTalk. М.: Издательский дом «Русская редакция», 2001.
29. *Черняк Л.* Общая шина предприятия. Электронная версия журнала «Открытые системы» №4/2003. [<http://www.osp.ru/text/302/182897>].
30. *Шульгин Д.* Microsoft BizTalk Server 2004 для интеграции приложений. Электронная версия журнала «ВУТЕ/Россия» №6/2005. [<http://www.bytemag.ru/?ID=603941>].
31. *Эйприл К., Мур К.* Пирог интеграции. Электронная версия еженедельника «Computerworld», №35/2002. [http://www.osp.ru/cw/2002/35/021_2.htm].
32. Computerage: улучшить систему, а не бороться с ней // Сетевой журнал: галерея ИТ-проектов. [<http://www.setevoj.ru/ITS/comprage78`2002.html>].

33. *Eric van der Vlist*. Using W3C XML Schema.
[<http://www.xml.com/pub/a/2000/11/29/schemas/part1.html>].
34. *Holman G. Ken*. What Is XSLT.
[<http://www.xml.com/pub/a/2000/08/holman/index.html>].

Список источников в Internet

35. Узел компании ИВК. [<http://www.ivk.ru>].
36. Узел консорциума W3C, посвященный языку XML. [<http://www.xml.com>].
37. Узел корпорации Axway, посвященный платформе интеграции Axway Integration Platform
[http://www.axway.com/templates/page.php?lang_code=EN&menu_mnemo=SOLU4].
38. Узел корпорации BEA-Systems. [<http://www.bea.com>].
39. Узел корпорации GXS, посвященный продуктам интеграции.
[<http://www.gxs.com/products/gateways>].
40. Узел корпорации IBM, посвященный продукту интеграции WebSphere.
[http://www.ibm.com/ru/software/websphere/bus_integr/].
41. Узел корпорации InterSystems, посвященный продукту интеграции Ensemble.
[<http://www.intersystems.ru/ensemble/technology/imagine/brochure-print.html>].
42. Узел корпорации Mercator Software. [<http://www.mercator.com>].
43. Узел корпорации Microsoft, посвященный BizTalk Server (русифицированный).
[<http://www.microsoft.com/rus/biztalk/>].
44. Узел корпорации Oracle, посвященный продуктам интеграции.
[<http://www.oracle.com/global/ru/products/middleware/index.html>].
45. Узел корпорации SAP, посвященный продукту NetWeaver.
[<http://www.sap.com/cis/solutions/netweaver/index.epx>].
46. Узел корпорации SeeBeyond, посвященный решениям в области интеграции.
[<http://www.seebeyond.com/software/ican.asp>].
47. Узел корпорации Sterling Commerce, посвященный решениям в области интеграции.
[<http://www.sterlingcommerce.com/Products/AllProducts/Gentran/integrationsuite.htm>].
48. Узел корпорации Sybase, посвященный решениям в области интеграции.
[<http://www.sybase.ru/Syb/products/middleware/index.htm>].
49. Узел корпорации Vitria Technology, посвященный решениям в области интеграции.
[<http://www.vitria.com/Product/smartgateway.php>].
50. Узел корпорации webMethods, посвященный решениям в области интеграции.
[<http://www.webmethods.com/meta/default/folder/0000008313>].
51. Узел справочной системы MSDN, посвященный технологии BizTalk.
[<http://msdn.microsoft.com/biztalk>].
52. Спецификация «BizTalk Framework 2.0: Document and Message Specification».
[<http://www.bixtalk.org>].

Приложение 1. Основные обозначения и сокращения

- CASE – Computer Aided Software Engineering, автоматизация разработки программного обеспечения.
- EAI – Enterprise Application Integration, интеграция систем (приложений) предприятия.
- IIS – Internet Information Server.
- METAS – METAdata System, система, основанная на метаданных.
- SOAP – Simple Object Access Protocol, простой протокол доступа к объектам.
- UML – Unified Modeling Language, унифицированный язык моделирования.
- XML – eXtensible Markup Language, расширенный язык разметки.
- XSLT – eXtensible Stylesheet Language Transformations, язык преобразований расширенной таблицы стилей.
- БД – база данных.
- БМД – база метаданных.
- ИС – информационная система.
- ПО – программное обеспечение.
- СУБД – системами управления базами данных.

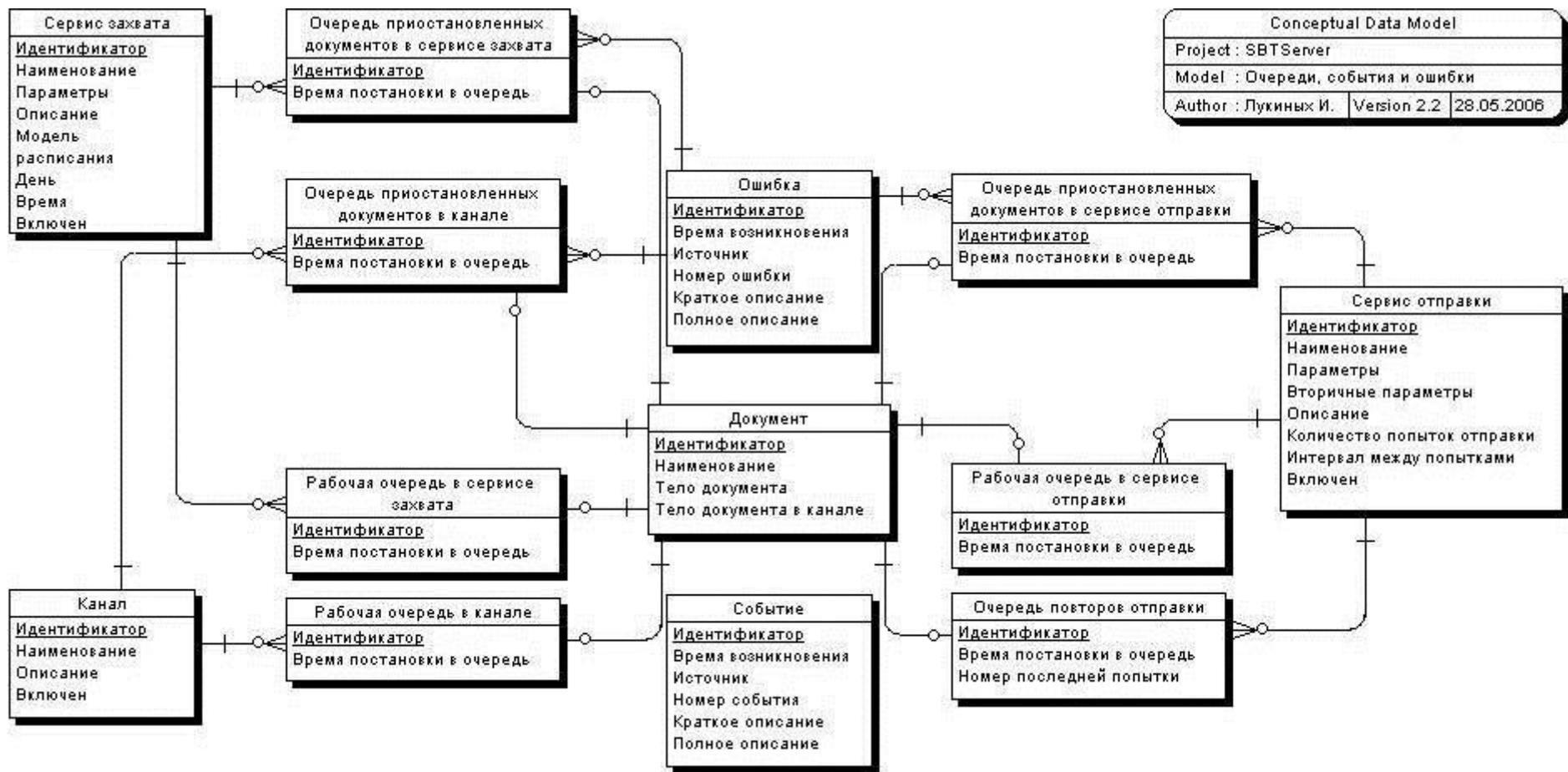


Рис. 2.2. Концептуальная модель таблиц очередей, событий и ошибок

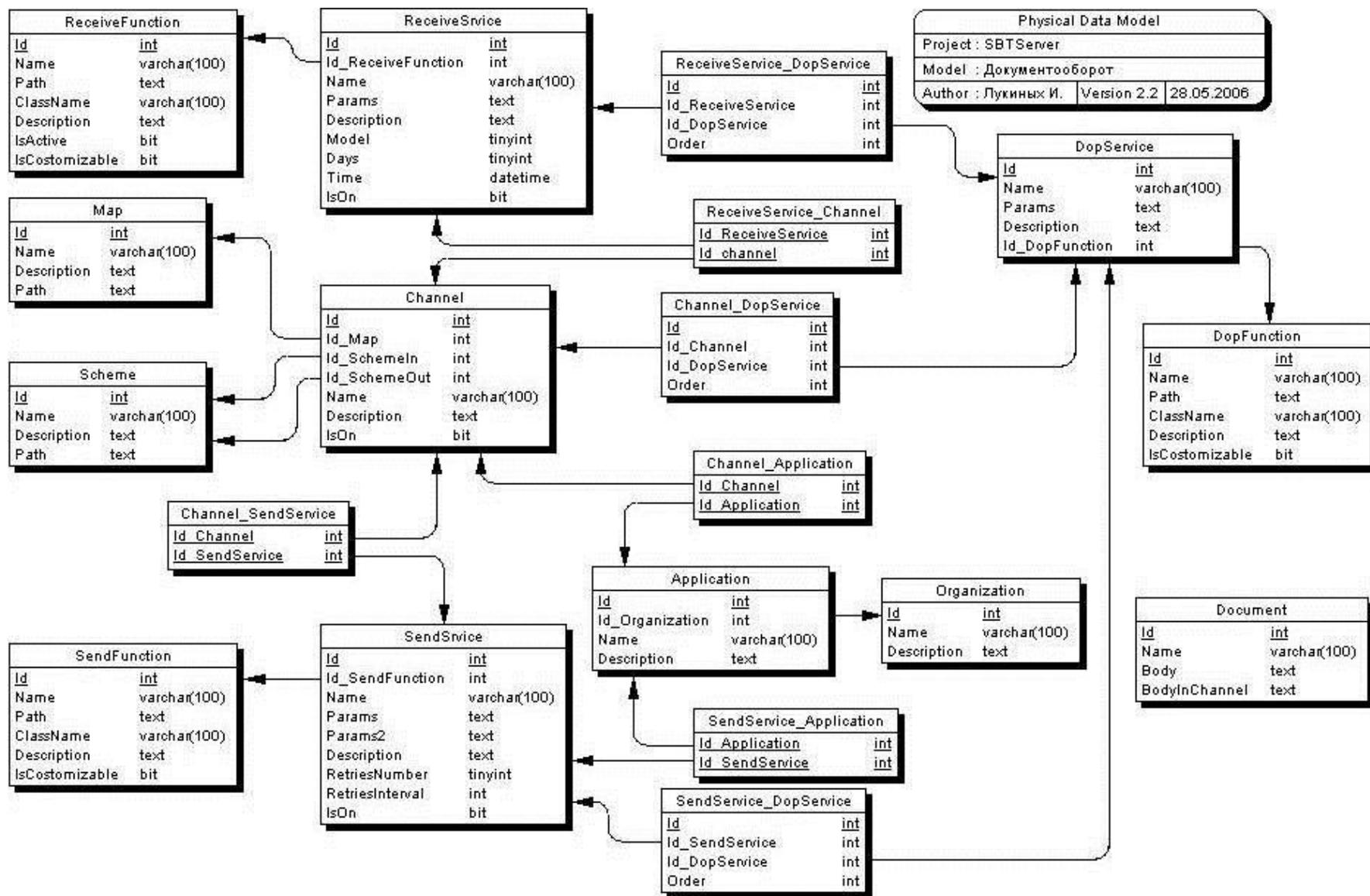


Рис. 2.3. Физическая модель таблиц объектов документооборота

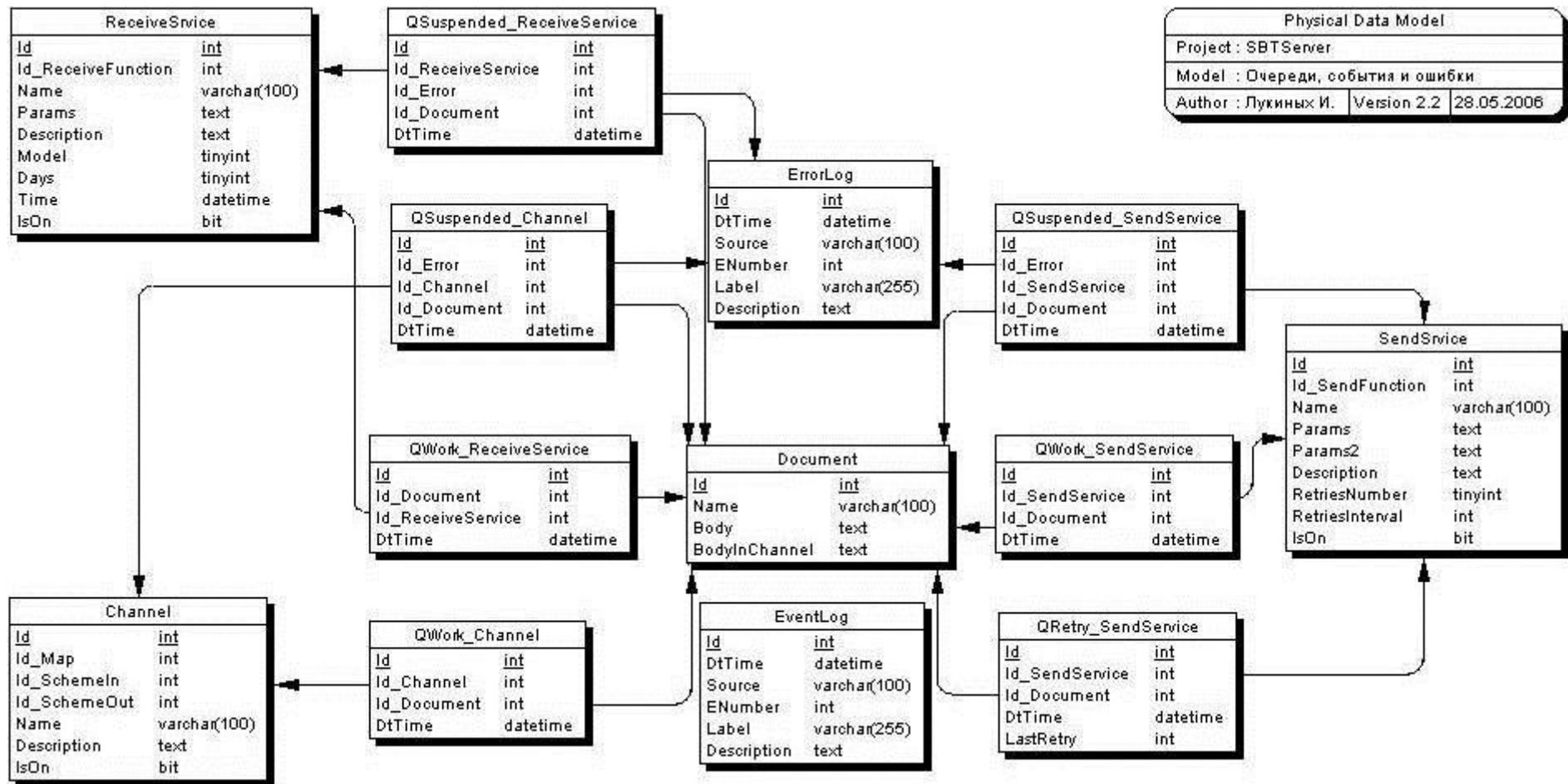


Рис. 2.4. Физическая модель таблиц очередей, событий и ошибок

Приложение 3. Руководство администратора

Установка системы

Установка приложения

Для установки приложения необходимо запустить файл Setup.exe, который представляет собой файл установки приложения. Для установки системы надо пройти несколько шагов. На первом шаге Мастер установки приглашает начать установку SBTServer (рис. 3.1). Мастер установки информирует о том, что проведет вас через все шаги, необходимые для установки SBTServer на компьютер и выдает предупреждение и защите программного продукта законом. Можно либо отказаться от установки, нажав на кнопку «Cancel», либо перейти к следующему шагу установки, нажав на кнопку «Next >>».



Рис. 3.1. Установка системы. Шаг 1

На следующем шаге необходимо выбрать каталог для установки системы. Рекомендуемый путь установки приложения – C:\Program Files\PSU\BizTalk. Если этот путь не устраивает, есть возможность прописать путь для установки файлов приложения, либо, воспользовавшись кнопкой «Browse...» выбрать каталог через диалоговое окно. С помощью кнопки «Disk Cost...» можно узнать количество свободного места на диске. Так же необходимо выбрать, устанавливается приложение для всех пользователей на этом компьютере или только для того, под которым происходит установка. После чего необходимо нажать кнопку «Next >>» (рис. 3.2). Можно вернуться на предыдущий шаг, нажав на кнопку «< Back» или вообще отказаться от установки, нажав на кнопку «Cancel».

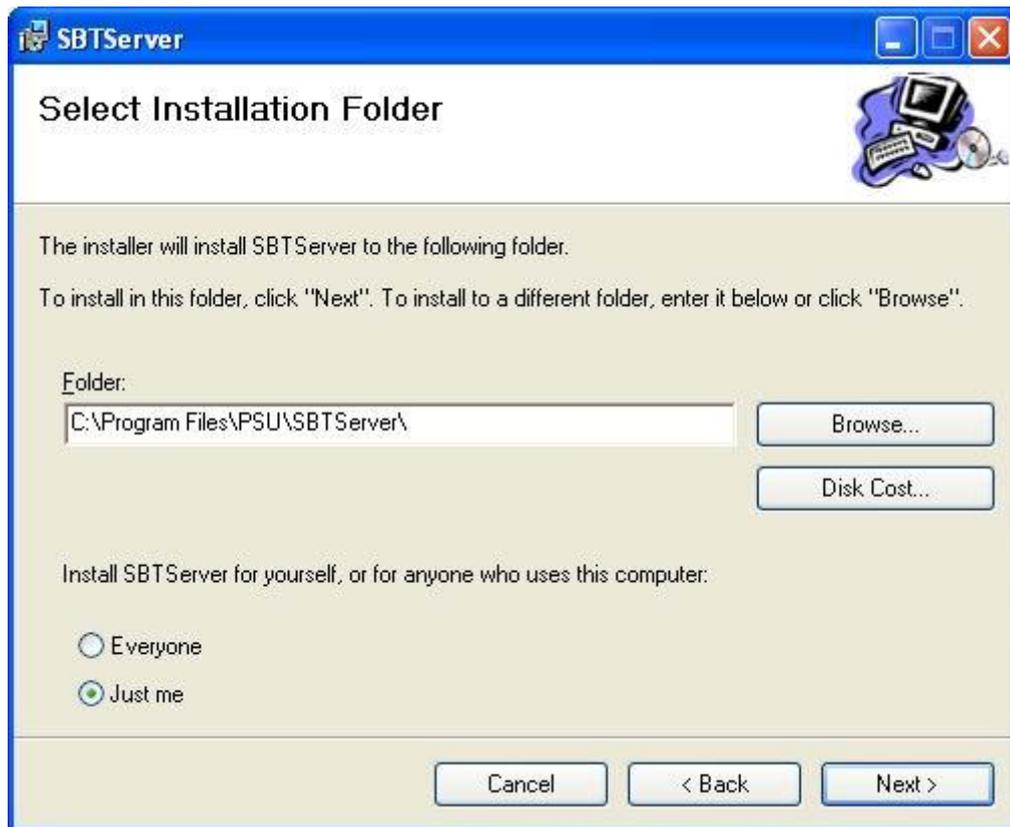


Рис. 3.2. Установка системы. Шаг 2

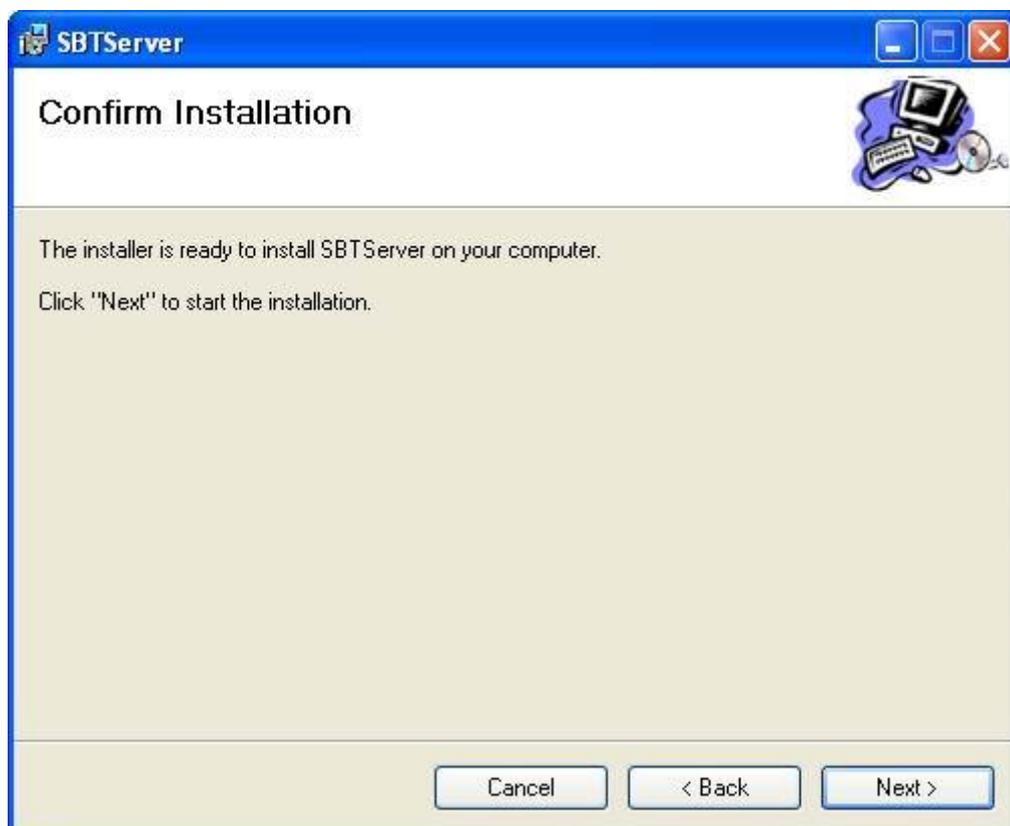


Рис. 3.3. Установка системы. Шаг 3

На следующем шаге Мастер попросит подтвердить установку системы. И всё, что остается сделать – нажать на кнопку «Next >», чтобы установить SBTServer на компьютер. На этом шаге можно отказаться от установки или вернуться на предыдущий шаг. Для этого служат кнопки «Cancel» и «< Back» соответственно (рис. 3.3). После нажатия на кнопку «Next >» на предыдущем шаге начинается непосредственно установка SBTServer на компьютер и форма примет вид, как на рис. 3.4.

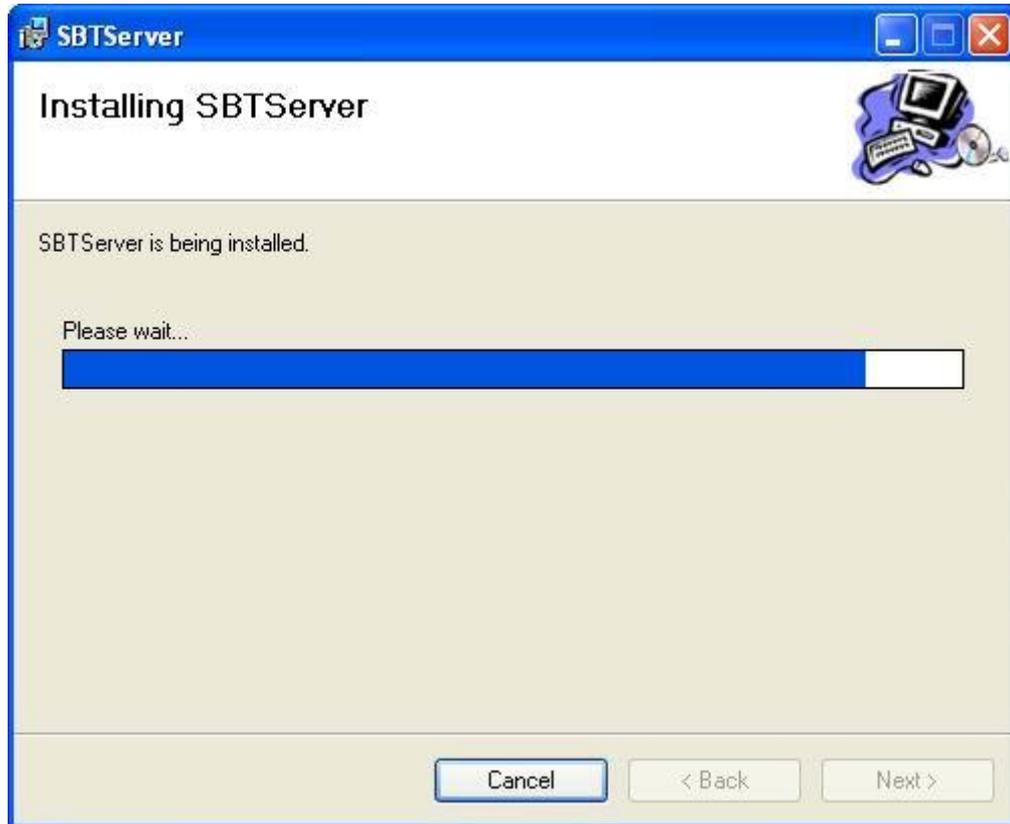


Рис. 3.4. Установка системы. Шаг 4

После окончания установки всех файлов, форма примет вид, как на рис. 3.5. На этом шаге сообщается, что установка системы успешно завершена. Для выхода из Мастера установки необходимо нажать «Close».

В результате выполнения установки должны появиться пункт меню «Пуск» - PSU->SBTServer и каталог PSU\SBTServer со следующей структурой (рис. 3.6):

- *Каталог SBTServer* – корневой каталог, в котором располагаются все остальные каталоги системы.
- *Каталог bin* – каталог, в котором расположены все исполняемые файлы:
 - файл SBTM_Scheduler.exe для запуска сервера интеграции Simple BizTalk Server,
 - файл SBTServer.exe для запуска приложения работы с объектами документооборота системы,
 - файл SBTAdmin.exe для администрирования сервера,
 - файл SBTConfig.exe для конфигурирования сервера,
 - файл SBTServer.ini – файл конфигурации запуска сервера

интеграции, его содержимое будет рассмотрено далее,

- файл SBTM_Data.dll – библиотека .NET, предназначенная для работы с БД,
- файл DB_Create.exe для создания базы данных системы.

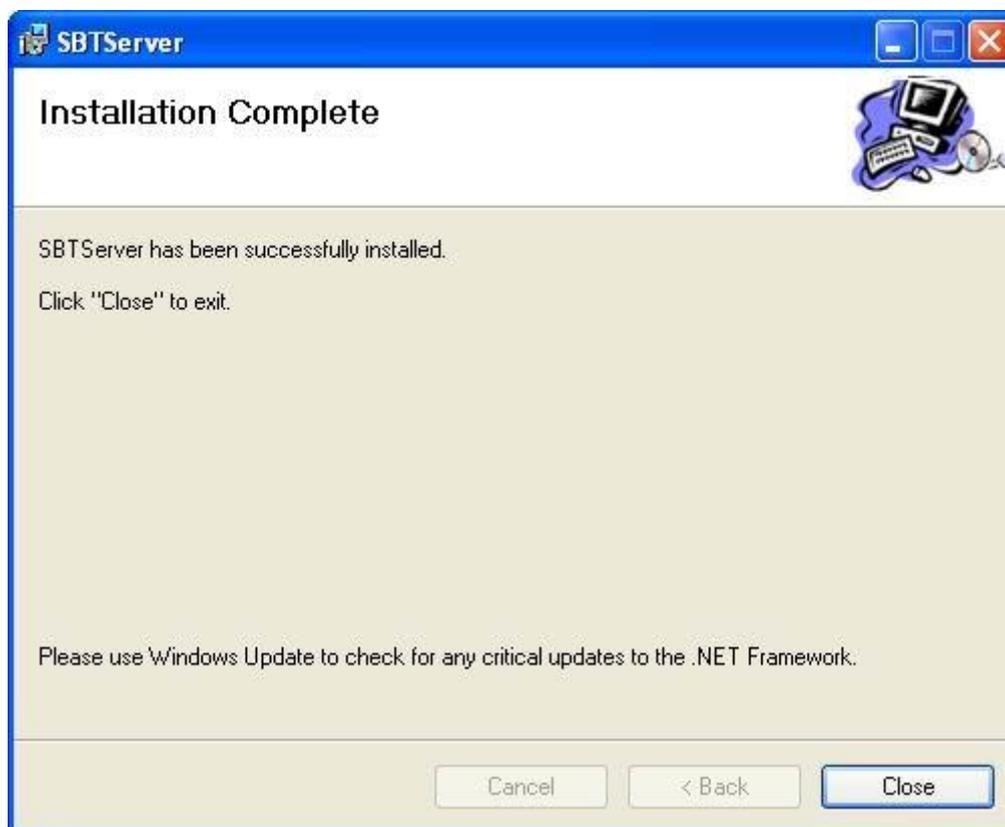


Рис. 3.5. Установка системы. Шаг 5

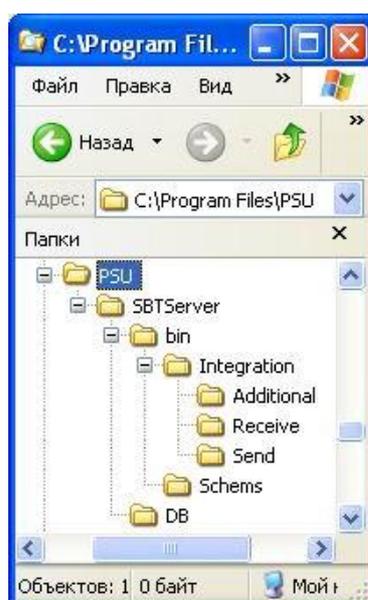


Рис. 3.6. Структура каталога BizTalk

- Каталог *bin\Integration\Additional* предназначен для хранения библиотек, реализующих дополнительные функции системы.
- Каталог *bin\Integration\Receive* предназначен для хранения библиотек, реализующих функции захвата документов.
- Каталог *bin\Integration\Send* предназначен для хранения библиотек, реализующих функции отправки документов.
- Каталог *bin\Schems* предназначен для хранения схем документов и схем преобразования (файлов с расширением *xsd* и *xslt*)
- Каталог *DB* – предназначен для хранения файла базы данных при работе приложения под управлением СУБД MS Access.

Для удаления системы необходимо зайти в Пуск->Панель управления->Установка и удаление программ, выбрать в списке SBTServer и нажать «Удалить». Появится окно удаления системы (рис. 3.7).

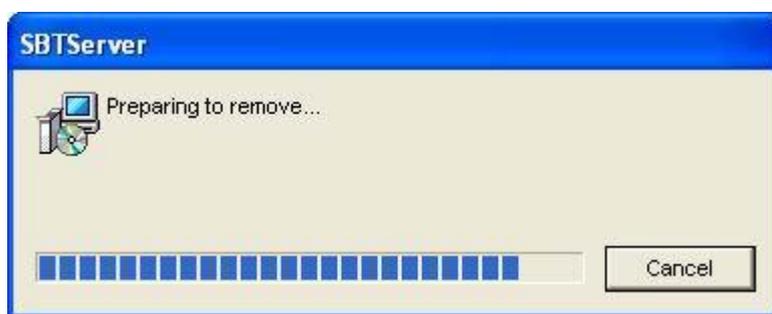


Рис. 3.7. Удаление системы

Установка базы данных

На данной стадии разработки система может функционировать под управлением двух СУБД: MS Access и MS SQL Server. Если выбрана установка версии под управлением MS Access, то в Windows по умолчанию уже должны быть установлены драйверы для работы с базами данных MS Access. Если же выбрана установка версии под управлением MS SQL Server, то на компьютер, на который производится установка системы, необходимо предварительно установить эту СУБД.

Для создания базы данных системы необходимо запустить файл *SBTServer\bin\DB_Create.exe*. Установка базы данных состоит из нескольких шагов, а само приложение работает подобно Мастеру установки.

На первом шаге необходимо выбрать СУБД (рис. 3.8).

Если на первом шаге выбрана СУБД MS Access, то на следующем шаге необходимо указать каталог и имя файла базы данных (рис. 3.9).

Если на первом шаге выбрана СУБД MS SQL Server, то на втором шаге необходимо указать имя сервера СУБД MS SQL Server и имя создаваемой базы данных (рис. 3.10), а на третьем ввести пароль к базе данных master (рис. 3.11).

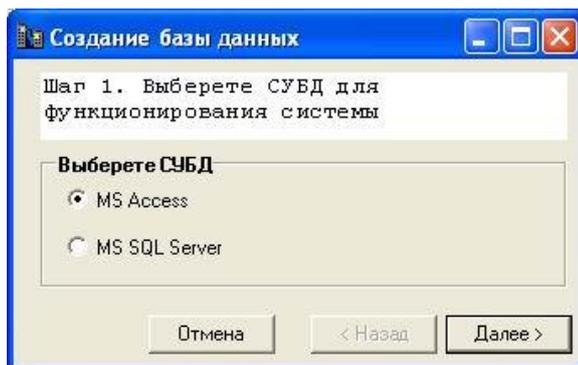


Рис. 3.8. Создание базы данных. Шаг 1

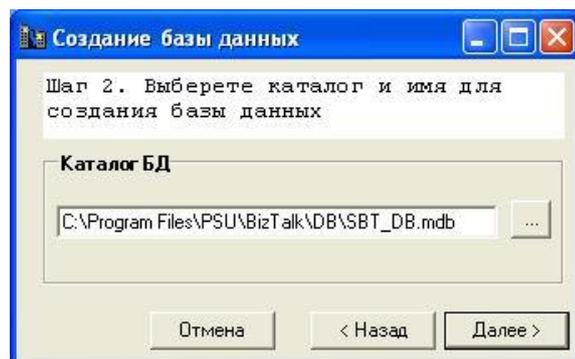


Рис. 3.9. Создание базы данных. Шаг 2а

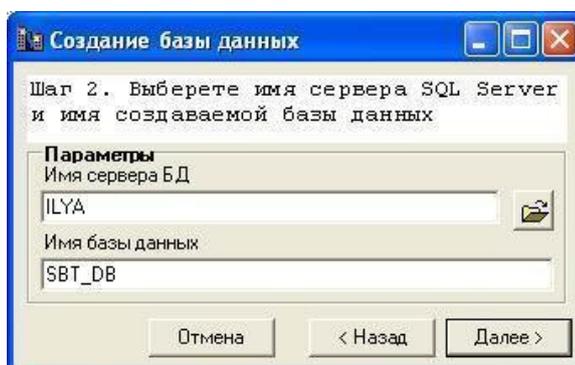


Рис. 3.10. Создание базы данных. Шаг 2б

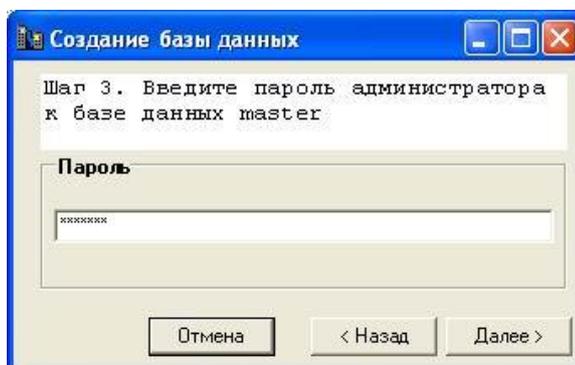


Рис. 3.11. Создание базы данных. Шаг 3б

Последующие шаги не зависят от выбора СУБД и отличаются только номером шага. На этом этапе необходимо ввести пароль для доступа к новой базе данных (рис. 3.12).

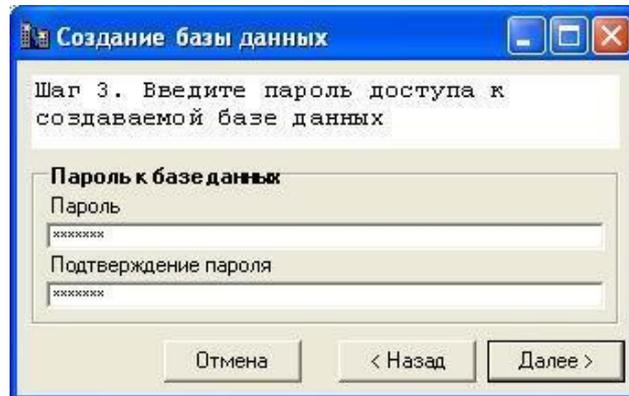


Рис. 3.12. Создание базы данных. Шаг 3 (4)

На следующем шаге остается только подтвердить создание базы данных (рис. 3.13).

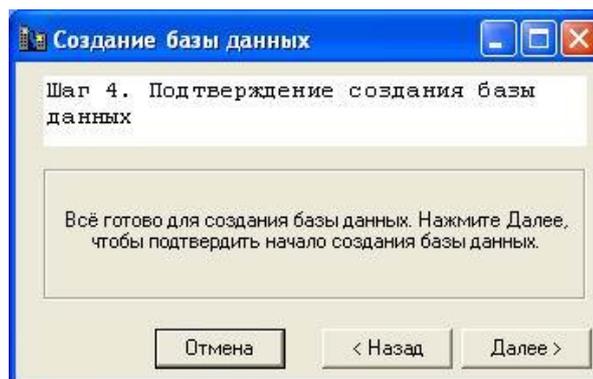


Рис. 3.13. Создание базы данных. Шаг 4 (5)

При нажатии на кнопку «Далее >» начнется создание базы данных и форма примет вид как на рис. 3.14.

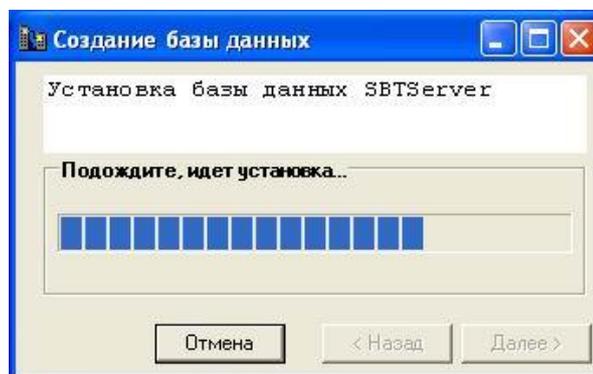


Рис. 3.14. Создание базы данных. Процесс установки

И если все проходит удачно, то установка завершается следующим сообщением (рис. 3.15).

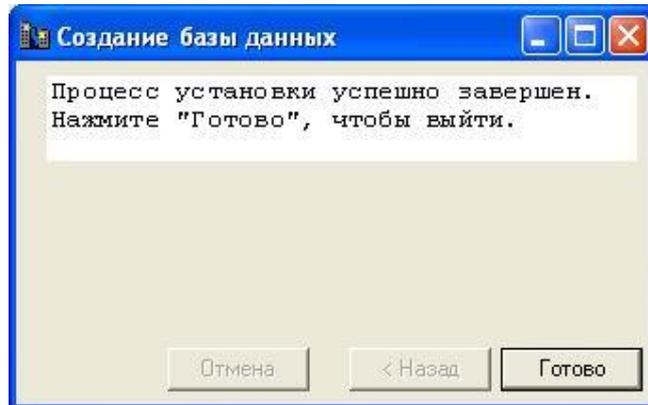


Рис. 3.15. Создание базы данных. Завершение

Настройка конфигурационных файлов

В системе имеется конфигурационный файл SBTServer.ini, который используют все приложения системы, работающие с базой данных. Файл представляет собой файл в формате xml, что позволяет достаточно прозрачно реализовать процесс настройки.

Если открыть на редактирование файл конфигурации, то можно обнаружить следующие параметры для настройки:

- `<ConnectionString>Provider = Microsoft.Jet.OLEDB.4.0; Data Source=%AppPath%\..\DB\ SBT_DB.mdb</ConnectionString>` – строка подключения к базе данных приложения в случае работы приложения под управлением СУБД MS Access. Provider=Microsoft.Jet.OLEDB.4.0 – указывает, что работа идет с базой данных MS Access, Data Source – путь до файла базы данных.
- `<ConnectionString>Provider=SQLOLEDB.1;Persist Security Info=True;User ID=SA;Initial Catalog=SBT_DB;Data Source=ILYA;Use Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=ILYA;Use Encryption for Data=False;Tag with column collation when possible=False</ConnectionString>` – строка подключения к базе данных приложения в случае работы приложения под управлением СУБД MS SQL Server. Provider=SQLOLEDB – указывает, что работа идет с базой данных MS SQL Server, Catalog=SBT_DB – имя базы данных, Data Source=ILYA – имя персонального компьютера, на котором располагается БД, Workstation ID=ILYA – имя персонального компьютера, на котором будет происходить работа с БД.

- <Password>□□□□□□□□□□</Password> – хранит в зашифрованном виде пароль к базе данных. Для установки пароля необходимо воспользоваться приложением конфигурации SBTConfig.exe.
- <ConnectionLimit>60</ConnectionLimit> - лимит подключений к базе данных. Сервер интеграции является многопоточковым приложением Windows и для каждого потока выделяется своё подключение к базе данных, однако некоторые СУБД накладывают ограничение на количество одновременно открытых подключений к базе данных, одной из таких СУБД является MS Access. Помимо этого параметр предназначен для управления загрузкой процессора, ведь чем больше разрешено подключений, тем больше может запускаться потоков приложения и наоборот.
- <ThreadLimit>100</ThreadLimit> – лимит запускаемых потоков приложения. В зависимости от характеристик компьютера необходимо изменять этот параметр. Чем больше лимит потоков – тем быстрее идет обработка документов в системе, но тем больше нагрузки идет на компьютер.

Лучше всего для настройки системы воспользоваться приложением SBTConfig.exe, которое позволяет изменять конфигурационный файл средствами Windows-интерфейса. В случае выбора СУБД MS Access это приложение будет выглядеть следующим образом (рис. 3.16).

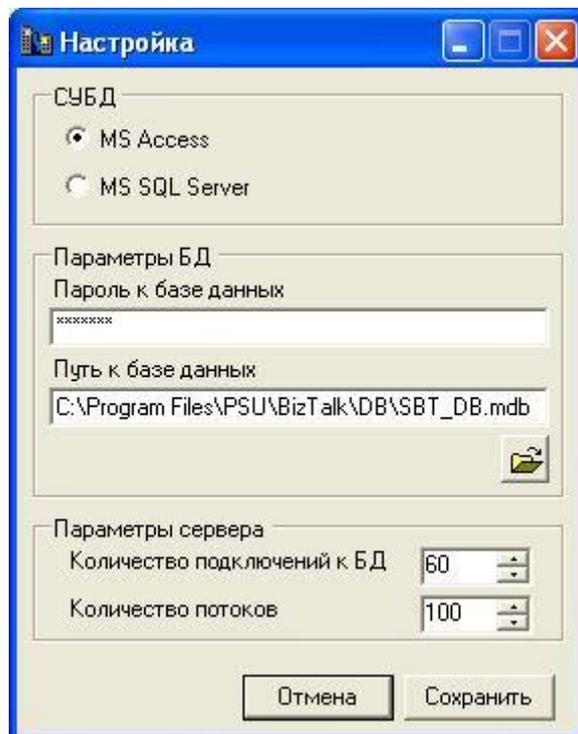


Рис. 3.16. Настройка системы. Access

Все параметры, которые описаны выше, можно задать, используя эту форму настройки. Первый параметр говорит о том, что используется драйвер СУБД Access для доступа к файлу базы данных системы. Во второй группе параметров указывается пароль доступа и путь к файлу БД. С помощью третьей группы параметров указывается лимит подключений к БД и лимит на количество потоков приложения.

Если используется СУБД SQL Server, то форма настройки приложения принимает вид как на рис. 3.17.

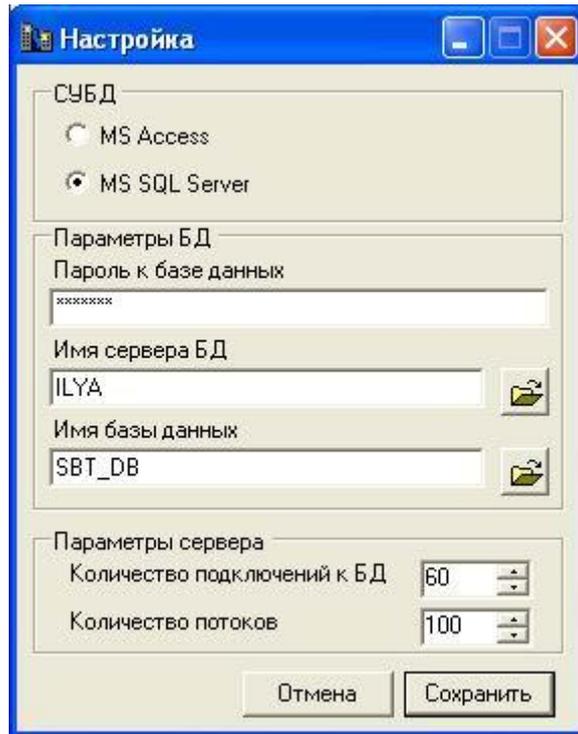


Рис. 3.17. Настройка системы. SQL Server

Отличием от предыдущей формы является указание в качестве драйвера доступа к БД драйвера СУБД MS SQL Server. Во второй группе теперь вместо пути к файлу БД нужно указать имя сервера SQL Server и имя базы данных. Остальные параметры аналогичны.

Документация по работе с SBTServer.exe - приложением для визуальной работы с объектами документооборота

Для организации документооборота используются следующие объекты:

- **Организация** – содержит в себе множество приложений и служит для кластеризации приложений.
- **Приложение** – делит каналы и сервисы отправки по принадлежности какому-либо приложению, которое в свою очередь принадлежит определенной организации.

- **Документ** – непосредственно документ, попавший в систему.
- **Схема документа** – XML-схема описания документа, используемая для проверки действительности документа поступающего и обрабатываемого в системе. Содержит описание полей, типов данных, обязательности, значений по умолчанию и других дополнительных атрибутов.
- **Карта преобразования** – XSL-схема преобразования XML-документов, описывает на языке XSL, как из одного документа получить другой в соответствии с XML-схемами документов.
- **Канал** – предназначен для проверки, обработки поступающего на вход документа и при необходимости преобразования его в другой XML-формат.
- **Функции захвата** – описывают класс, используя объекты которого можно получить документ по определенному протоколу. Причем функции могут быть активными и пассивными.
- **Сервисы захвата** – описывают периодичность захвата документов через определенную функцию и параметры захвата. Полученный документ направляется в соответствующие каналы.
- **Функции отправки** – описывают класс, используя объекты которого можно отправить документ по определенному протоколу.
- **Сервис отправки** – предназначен для отправки документа через функции интеграции отправки по определенному данной функцией протоколу, а также параметры отправки.
- **Дополнительные функции** – описывают класс, используя объекты которого можно преобразовать документ. К преобразованиям относятся шифровка, архивация, сжатие и т.д.
- **Дополнительные сервисы** – предназначены для преобразования документов с помощью дополнительных функций интеграции с параметрами для этих функций

На рис. 3.18 представлено главное окно приложения SBTServer. В левой части расположено дерево объектов документооборота, в правой части сверху – краткая информация об объекте, выделенном в дереве объектов, а в правой нижней части в виде закладок помещается детальная информация об объекте, выделенном в верхней части. Информация на закладках доступна только для чтения, если нужно отредактировать значения каких-либо параметров, то необходимо нажать на кнопку «Редактировать». Если необходимо создать выделенный объект системы, то для этого существует кнопка «Создать». Как в первом, так и во втором случае открывается отдельная форма.

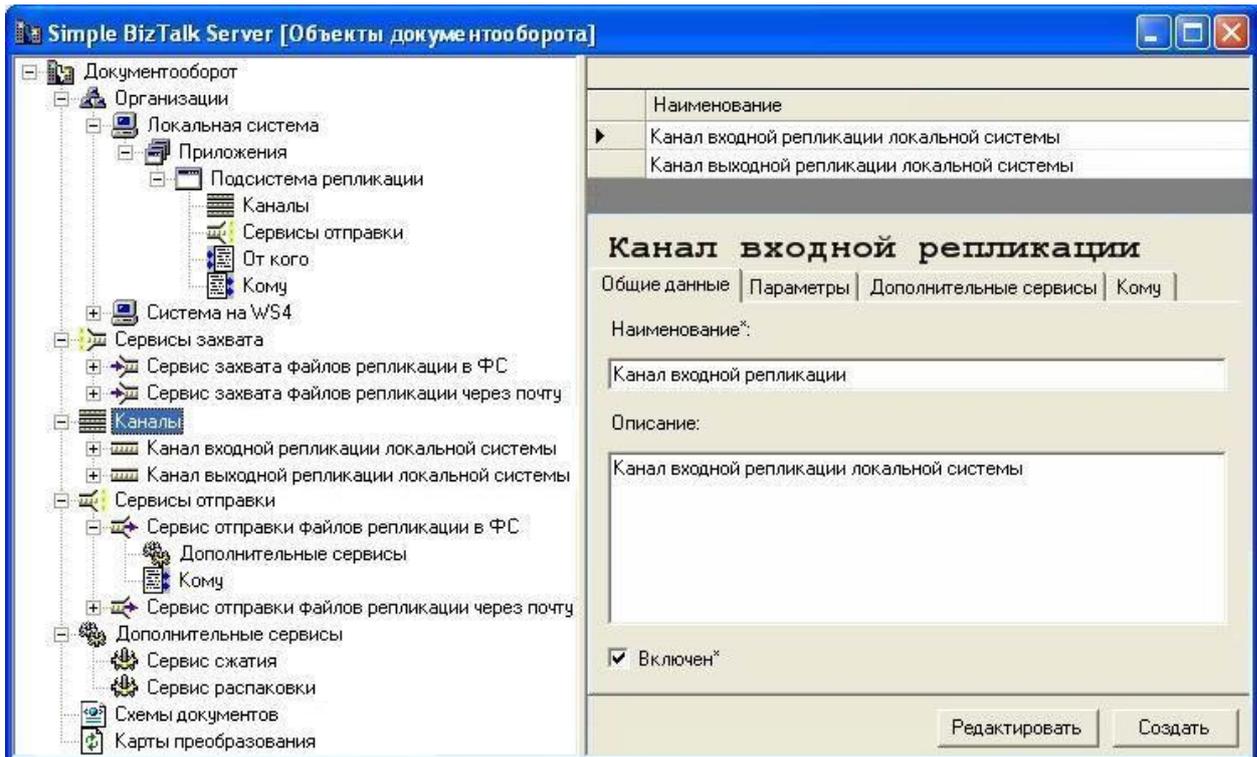


Рис. 3.18.. Объекты документооборота

Редактирование и добавление осуществляется при помощи мастера. Мастер реализован как набор вкладок. На каждой вкладке сгруппированы поля для заполнения свойств объекта. Обязательные поля помечены звездочкой (*). Рассмотрим подробно свойства каждого объекта документооборота.

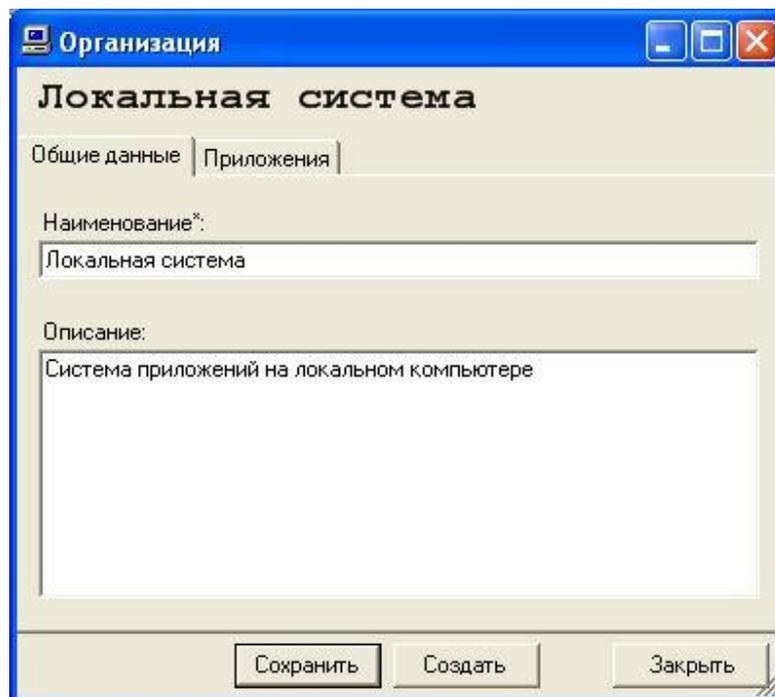


Рис. 3.19. Организация. Общие данные

На форме редактирования организации в верхней части отображается наименование организации и имеется две вкладки.

На рис. 3.19 показаны поля вкладки «Общие данные»:

- Наименование – обязательное поле, содержит уникальное в системе наименование объекта организации. Если попытаться сохранить объект с наименованием, которое присутствует в системе, то выдается сообщение об этой ошибке, данные не сохраняются и окно редактирования не закрывается.
- Описание – необязательное поле, содержит полное описание организации. Длина описания не должна превышать 2000 символов. Если попытаться сохранить объект с описанием, длина которого превышает 2000 символов, то выдается сообщение об этой ошибке, данные не сохраняются и окно редактирования не закрывается.

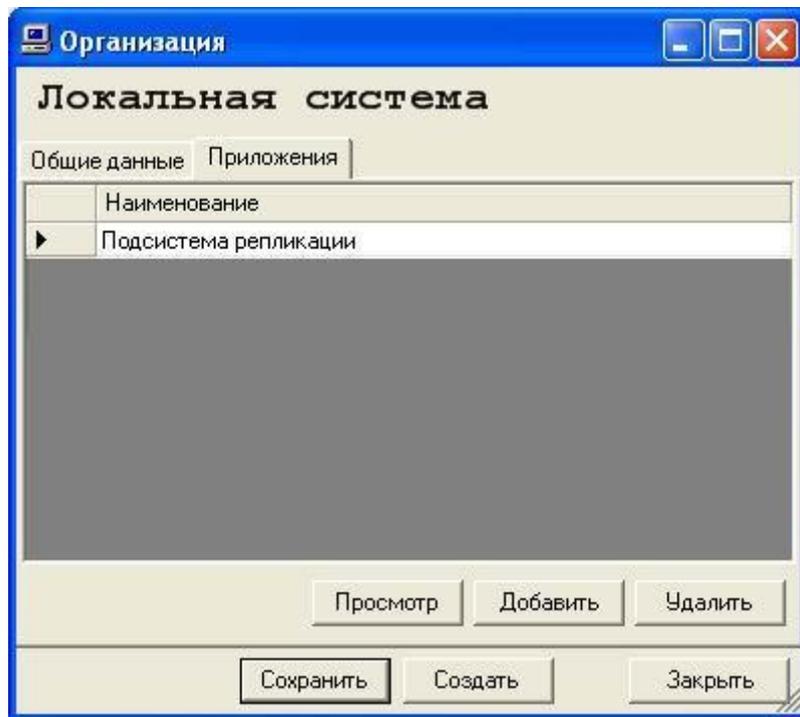


Рис. 3.20. Организация. Приложения

На рис. 3.20 показано содержимое вкладки «Приложения»:

- Таблица – таблица с наименованиями приложений, которые принадлежат данной организации.
- Кнопка «Просмотр» – при нажатии на эту кнопку открывается форма редактирования выделенного в таблице приложения.
- Кнопка «Добавить» – при нажатии на эту кнопку открывается форма создания нового приложения, принадлежащего данной организации.
- Кнопка «Удалить» – при нажатии на эту кнопку задается вопрос, действительно ли Вы хотите удалить данное приложение и все связи с

ним из системы. Если ответ отрицательный, то приложение не удаляется, а если положительный, то удаляется приложение и все ссылки на него.

На форме редактирования приложения в верхней части отображается наименование приложения и имеется четыре вкладки.

На рис. 3.21 показаны поля вкладки «Общие данные»:

- Организация – обязательное поле, указывающее, какой организации системы принадлежит данное приложение. Есть возможность выбрать существующую организацию нажатием кнопки  или создать новую организацию, которой будет принадлежать данное приложение, нажатием кнопки .

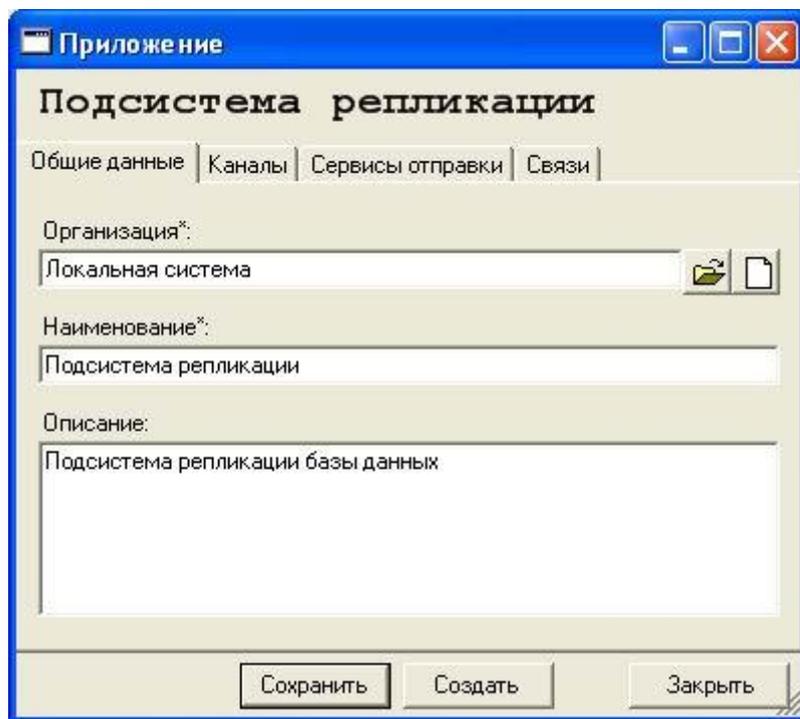


Рис. 3.21. Приложение. Общие данные

- Наименование – обязательное поле, содержит уникальное в системе наименование объекта приложения. Если попытаться сохранить объект с наименованием, которое присутствует в системе, то выдается сообщение об этой ошибке, данные не сохраняются и окно редактирования не закрывается.
- Описание – необязательное поле, содержит полное описание приложения. Длина описания не должна превышать 2000 символов. Если попытаться сохранить объект с описанием, длина которого превышает 2000 символов, то выдается сообщение об этой ошибке, данные не сохраняются и окно редактирования не закрывается.

На рис. 3.22 показано содержимое вкладки «Каналы»:

- Таблица – таблица с наименованиями каналов, в которые отправляет документы данное приложение.
- Кнопка «Просмотр» – при нажатии на эту кнопку открывается форма редактирования выделенного в таблице канала.
- Кнопка «Добавить» – при нажатии на эту кнопку открывается форма со списком существующих каналов системы, можно выбрать каналы из этого списка или создать новые.
- Кнопка «Удалить» – при нажатии на эту кнопку задается вопрос, действительно ли Вы хотите удалить данный канал из списка этого приложения. Если ответ отрицательный, то канал остается в списке, а если положительный, то удаляется связь между приложением и выбранным каналом, при этом сам канал остается в системе.

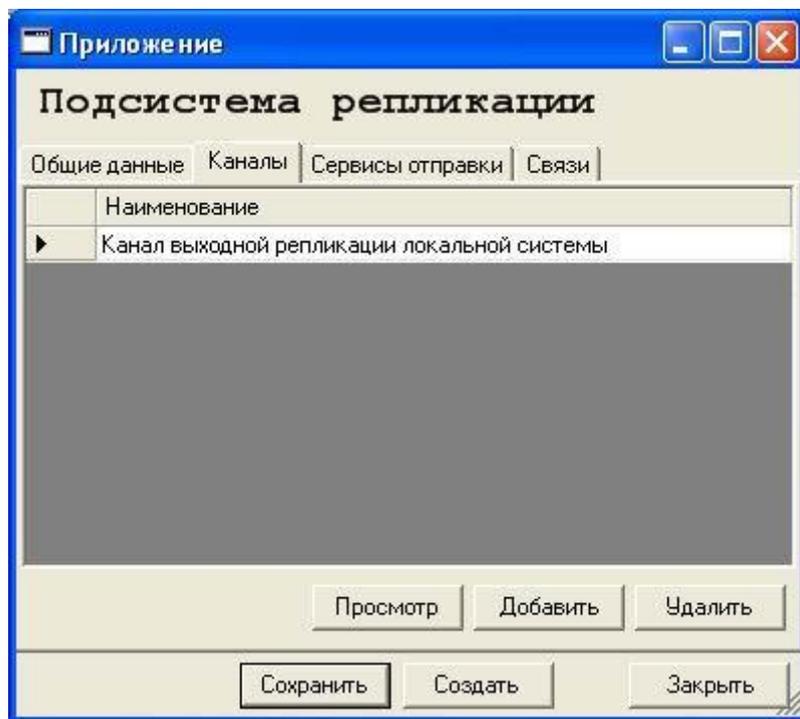


Рис. 3.22. Приложение. Каналы

На рис. 3.23. показано содержимое вкладки «Сервисы отправки»:

- Таблица – таблица с наименованиями сервисов отправки, от которых данное приложение получает документы.
- Кнопка «Просмотр» – при нажатии на эту кнопку открывается форма редактирования выделенного в таблице сервиса отправки.
- Кнопка «Добавить» – при нажатии на эту кнопку открывается форма со списком существующих сервисов отправки системы, можно выбрать сервисы отправки из этого списка или создать новые.

- Кнопка «Удалить» – при нажатии на эту кнопку задается вопрос, действительно ли Вы хотите удалить данный сервис отправки из списка этого приложения. Если ответ отрицательный, то сервис отправки остается в списке, а если положительный, то удаляется связь между приложением и выбранным сервисом отправки, при этом сам сервис отправки остается в системе.

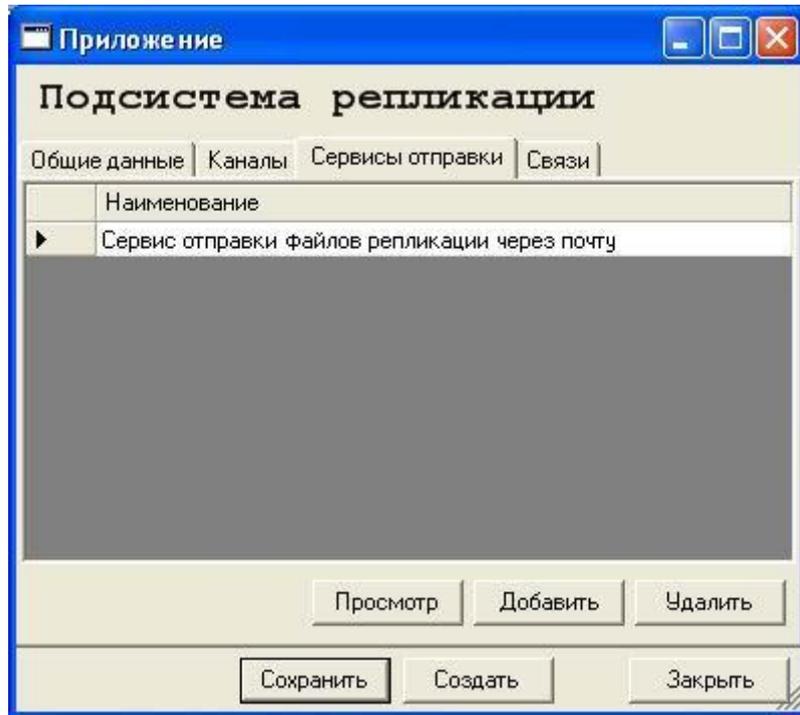


Рис. 3.23. Приложение. Сервисы отправки

На рис. 3.24 показано содержимое вкладки «Связи»:

- Таблица «От кого» – таблица с наименованиями приложений, от которых данное приложение получает документы. Данный список собирается из приложений-владельцев каналов, которые отправляют документы в сервисы отправки, принадлежащих данному приложению.
- Таблица «Кому» – таблица с наименованиями приложений, которым данное приложение отправляет документы. Данный список собирается из приложений-владельцев сервисов отправки, в которые отправляют документы каналы, принадлежащие данному приложению.

На форме редактирования сервиса захвата в верхней части отображается наименование сервиса захвата и имеется четыре вкладки.

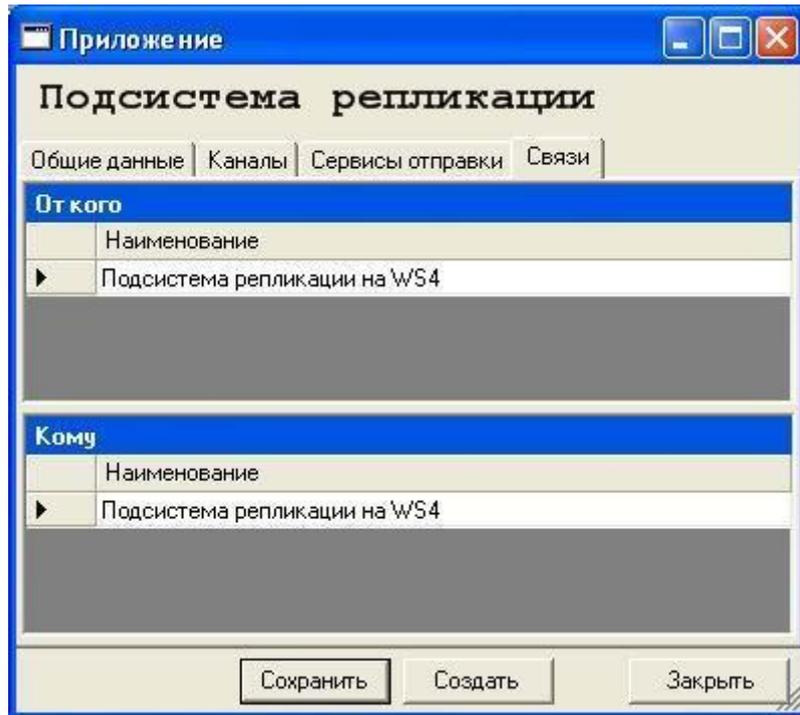


Рис. 3.24. Приложение. Связи

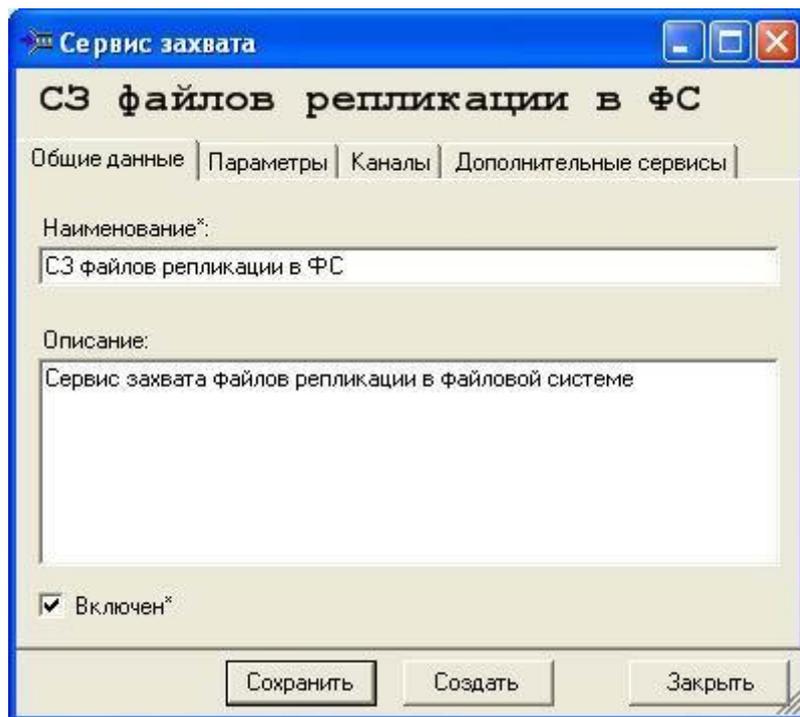


Рис. 3.25. Сервис захвата. Общие данные

На рис. 3.25 показаны поля вкладки «Общие данные»:

- Наименование – обязательное поле, содержит уникальное в системе наименование объекта сервиса захвата. Если попытаться сохранить объект с наименованием, которое присутствует в системе, то выдается

сообщение об этой ошибке, данные не сохраняются и окно редактирования не закрывается.

- Описание – необязательное поле, содержит полное описание сервиса захвата. Длина описания не должна превышать 2000 символов. Если попытаться сохранить объект с описанием, длина которого превышает 2000 символов, то выдается сообщение об этой ошибке, данные не сохраняются и окно редактирования не закрывается.
- Включен – обязательное поле, указывает состояние сервиса захвата. Если галка стоит, то сервис захвата работает и документы проходят через него обработку, в противном случае, документы не поступают в систему через данный сервис захвата.

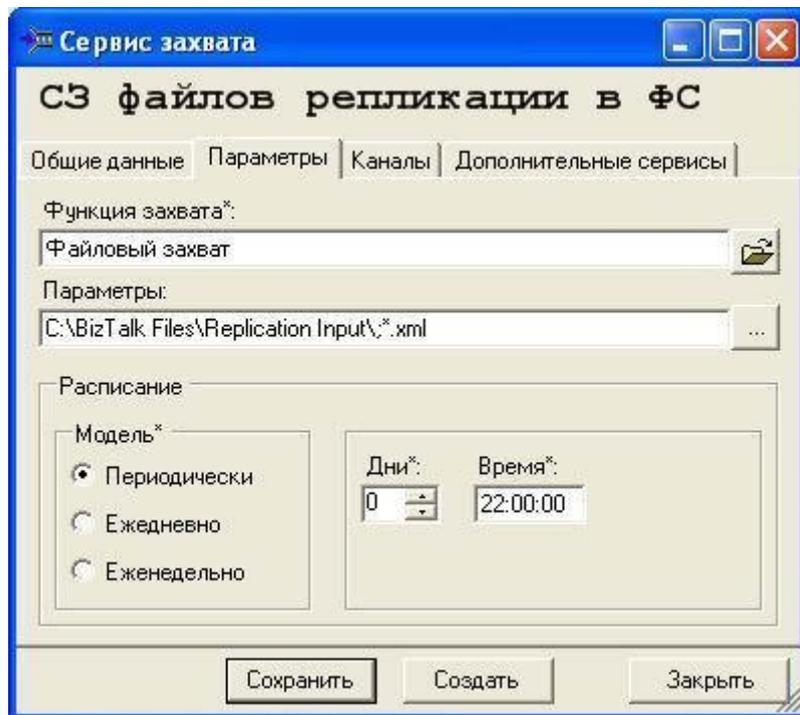


Рис. 3.26. Сервис захвата. Параметры

На рис. 3.26 показаны поля вкладки «Параметры»:

- Функция захвата – обязательное поле, указывающее, с помощью какой функции захвата данный сервис захвата получает документы из внешнего мира. Есть возможность выбрать другую функцию захвата нажатием кнопки .
- Параметры – необязательно поле с параметрами для настройки функции захвата. Есть возможность вызвать форму для настройки выбранной функции захвата нажатием кнопки .

- Расписание – обязательная группа параметров, которая задает расписание запуска данного сервиса:
 - Периодически – запуск осуществляется через каждый указанный промежуток времени (рис. 3.26).
 - Ежедневно – запуск осуществляется ежедневно в указанное время (рис. 3.27).

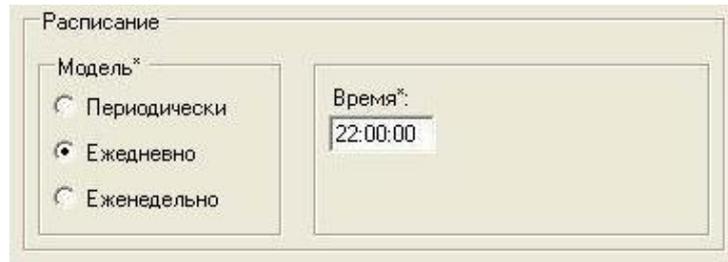


Рис. 3.27. Расписание сервиса захвата. Ежедневно

- Еженедельно – запуск осуществляется в указанное время в отмеченные дни недели (рис. 3.28).

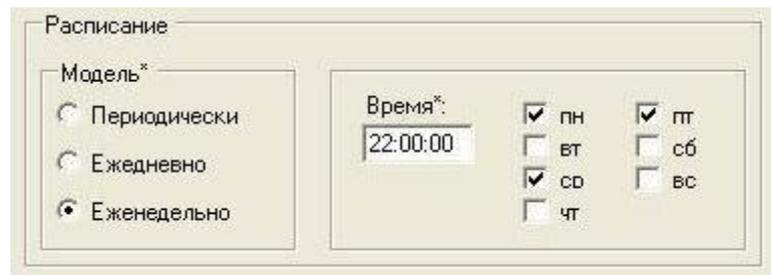


Рис. 3.28. Расписание сервиса захвата. Еженедельно

На рис. 3.29 показано содержимое вкладки «Каналы»:

- Таблица – таблица с наименованиями каналов, в которые отправляет документы данный сервис захвата.
- Кнопка «Просмотр» – при нажатии на эту кнопку открывается форма редактирования выделенного в таблице канала.
- Кнопка «Добавить» – при нажатии на эту кнопку открывается форма со списком существующих каналов системы, можно выбрать каналы из этого списка или создать новые.
- Кнопка «Удалить» – при нажатии на эту кнопку задается вопрос, действительно ли Вы хотите удалить данный канал из списка этого сервиса захвата. Если ответ отрицательный, то канал остается в списке, а если положительный, то удаляется связь между данным сервисом захвата и выбранным каналом, при этом сам канал остается в системе.

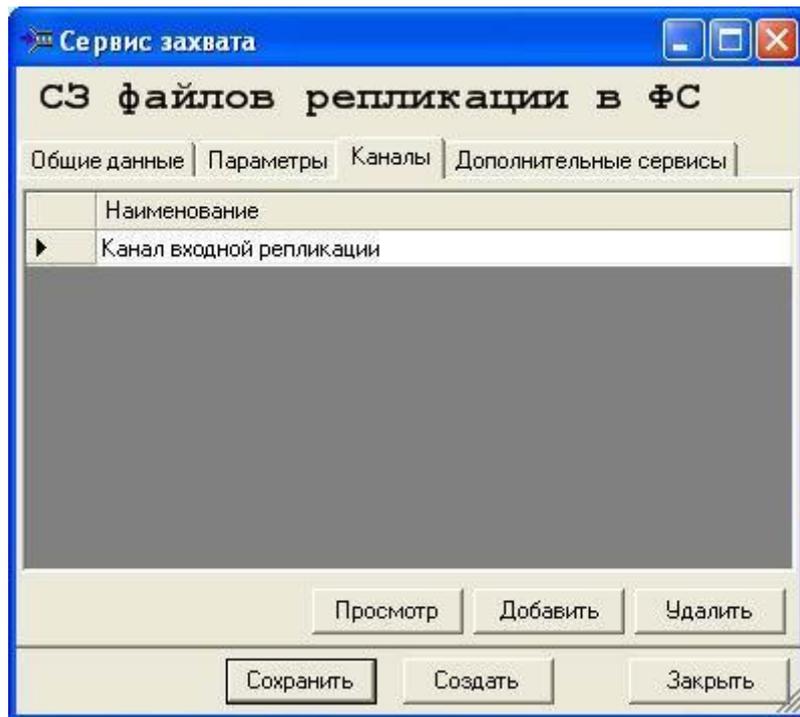


Рис. 3.29. Сервис захвата. Каналы

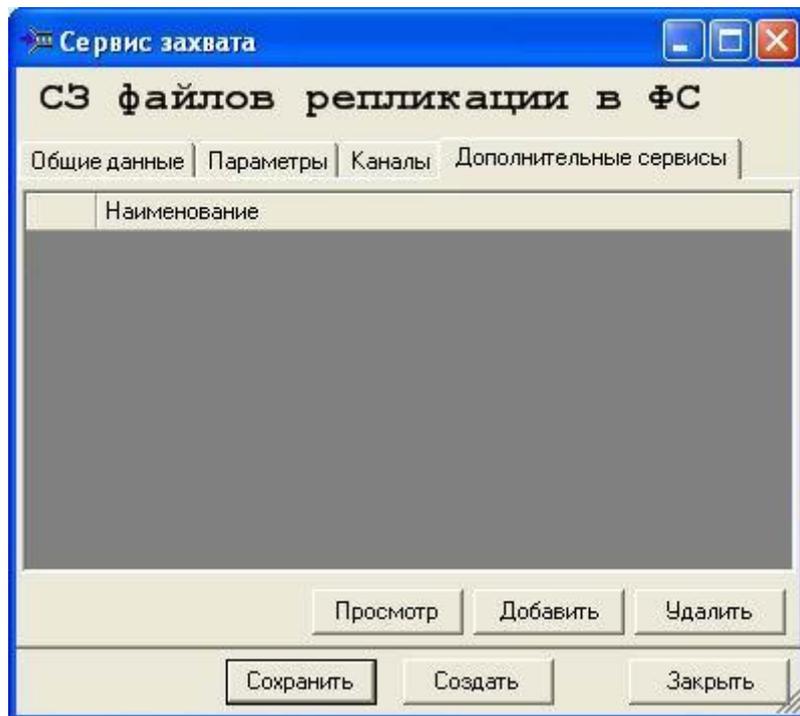


Рис. 3.30. Сервис захвата. Дополнительные сервисы

На рис. 3.30 показано содержимое вкладки «Дополнительные сервисы»:

- Таблица – таблица с наименованиями дополнительных сервисов, которые вызываются во время обработки документа в данном сервисе захвата.
- Кнопка «Просмотр» – при нажатии на эту кнопку открывается форма редактирования выделенного в таблице дополнительного сервиса.
- Кнопка «Добавить» – при нажатии на эту кнопку открывается форма со списком существующих дополнительных сервисов системы, можно выбрать сервисы из этого списка или создать новые.
- Кнопка «Удалить» – при нажатии на эту кнопку задается вопрос, действительно ли Вы хотите удалить данный дополнительный сервис из списка этого сервиса захвата. Если ответ отрицательный, то дополнительный сервис остается в списке, а если положительный, то удаляется связь между данным сервисом захвата и выбранным дополнительным сервисом, при этом сам дополнительный сервис остается в системе.

На форме редактирования канала в верхней части отображается наименование канала и имеется шесть вкладок.

На рис. 3.31 показаны поля вкладки «Общие данные»:

- Наименование – обязательное поле, содержит уникальное в системе наименование объекта канала. Если попытаться сохранить объект с наименованием, которое присутствует в системе, то выдается сообщение об этой ошибке, данные не сохраняются и окно редактирования не закрывается.
- Описание – необязательное поле, содержит полное описание канала. Длина описания не должна превышать 2000 символов. Если попытаться сохранить объект с описанием, длина которого превышает 2000 символов, то выдается сообщение об этой ошибке, данные не сохраняются и окно редактирования не закрывается.
- Включен – обязательное поле, указывает состояние канала. Если галка стоит, то канал работает и документы проходят через него обработку, в противном случае, документы не отправляются в данный канал.

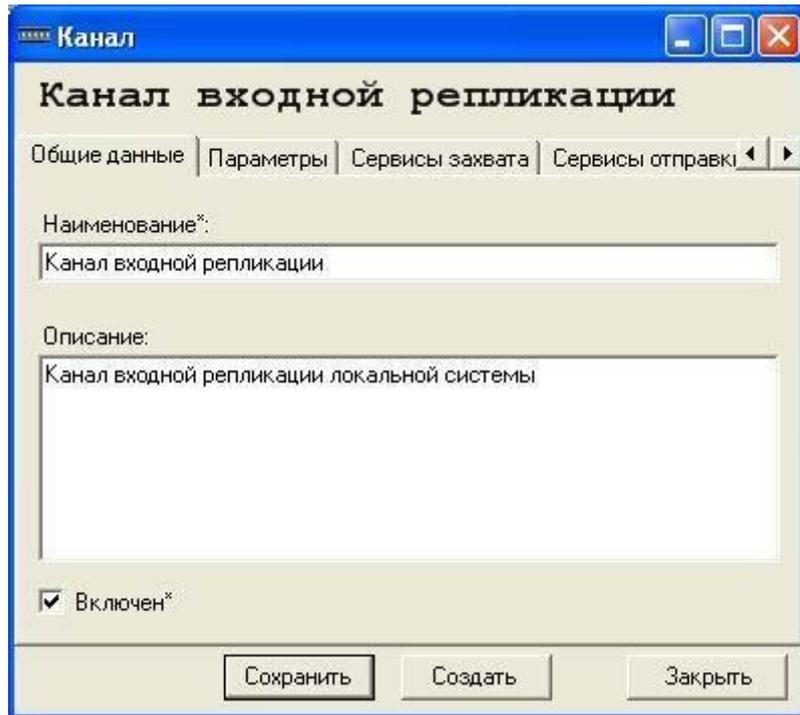


Рис. 3.31. Канал. Общие данные

На рис. 3.32 показаны поля вкладки «Параметры»:

- Схема входящего документа – обязательное поле, указывающее, какая схема документа используется для проверки документа при поступлении в канал. Есть возможность выбрать существующую схему документа нажатием кнопки  или создать новую схему, которая будет использоваться для проверки, нажатием кнопки .
- Карта преобразования – необязательное поле, указывающее, какая карта преобразования документа используется для изменения формата документа. Есть возможность выбрать существующую карту преобразования документа нажатием кнопки  или создать новую карту преобразования, которая будет использоваться для перевода в другой формат, нажатием кнопки .
- Схема исходящего документа – необязательное поле, указывающее, какая схема документа используется для проверки документа после применения карты преобразования в канале. Есть возможность выбрать существующую схему документа нажатием кнопки  или создать новую схему, которая будет использоваться для проверки, нажатием кнопки .

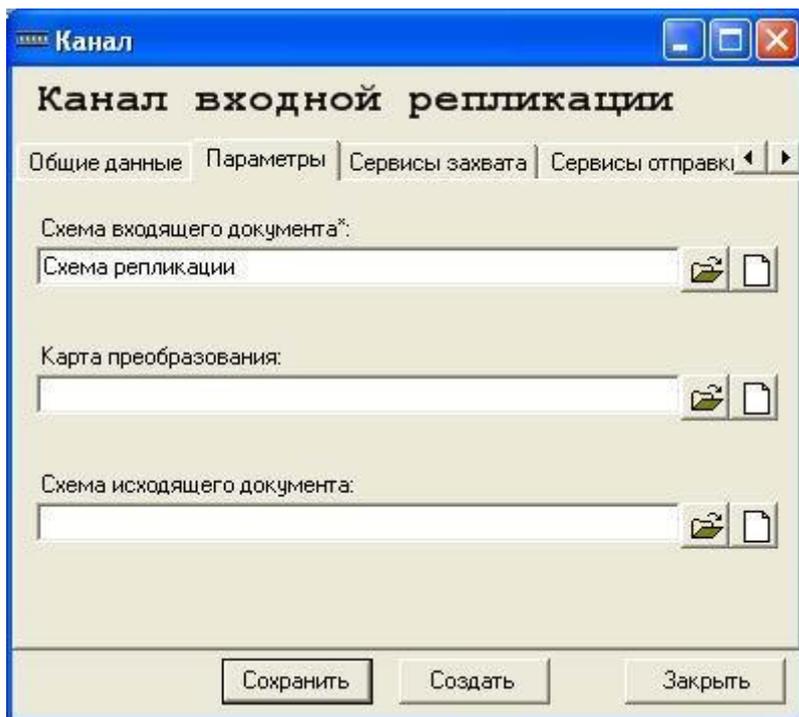


Рис. 3.32. Канал. Параметры

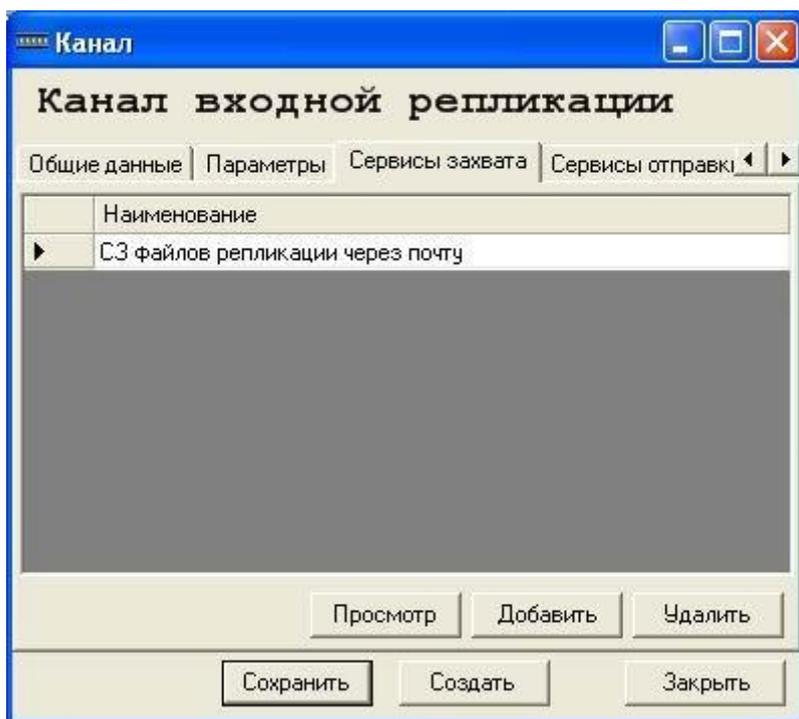


Рис. 3.33. Канал. Сервисы захвата

На рис. 3.33 показано содержимое вкладки «Сервисы захвата»:

- Таблица – таблица с наименованиями сервисов захвата, которые отправляют документы в данный канал.
- Кнопка «Просмотр» – при нажатии на эту кнопку открывается форма

редактирования выделенного в таблице сервиса захвата.

- Кнопка «Добавить» – при нажатии на эту кнопку открывается форма со списком существующих сервисов захвата системы, можно выбрать сервисы захвата из этого списка или создать новые.
- Кнопка «Удалить» – при нажатии на эту кнопку задается вопрос, действительно ли Вы хотите удалить данный сервис захвата из списка этого канала. Если ответ отрицательный, то сервиса захвата остается в списке, а если положительный, то удаляется связь между данным каналом и выбранным сервисом захвата, при этом сам сервис захвата остается в системе.

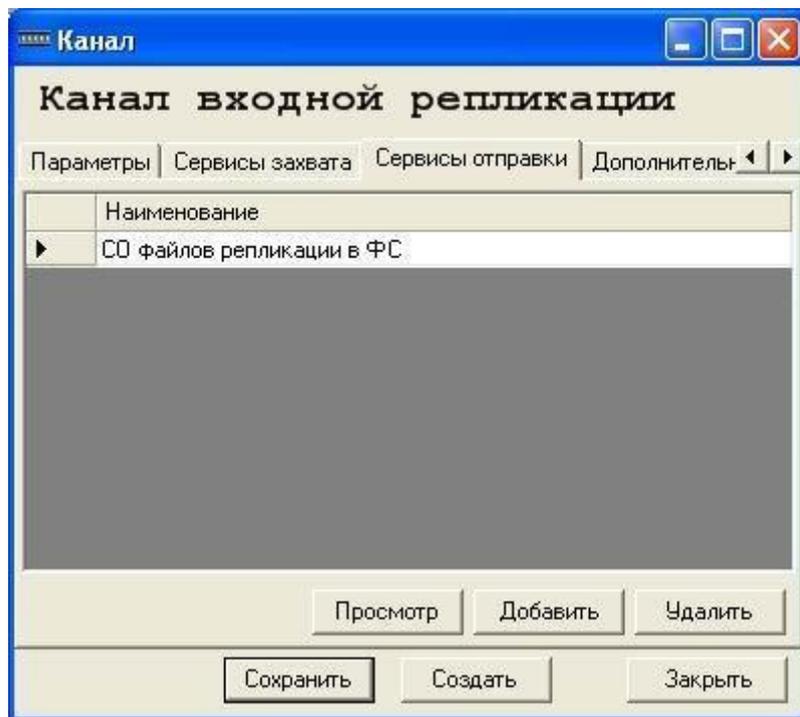


Рис. 3.34. Канал. Сервисы отправки

На рис. 3.34 показано содержимое вкладки «Сервисы отправки»:

- Таблица – таблица с наименованиями сервисов отправки, в которые данный канал отправляет документы.
- Кнопка «Просмотр» – при нажатии на эту кнопку открывается форма редактирования выделенного в таблице сервиса отправки.
- Кнопка «Добавить» – при нажатии на эту кнопку открывается форма со списком существующих сервисов отправки системы, можно выбрать сервисы отправки из этого списка или создать новые.
- Кнопка «Удалить» – при нажатии на эту кнопку задается вопрос, действительно ли Вы хотите удалить данный сервис отправки из списка этого канала. Если ответ отрицательный, то сервис отправки остается в

списке, а если положительный, то удаляется связь между каналом и выбранным сервисом отправки, при этом сам сервис отправки остается в системе.

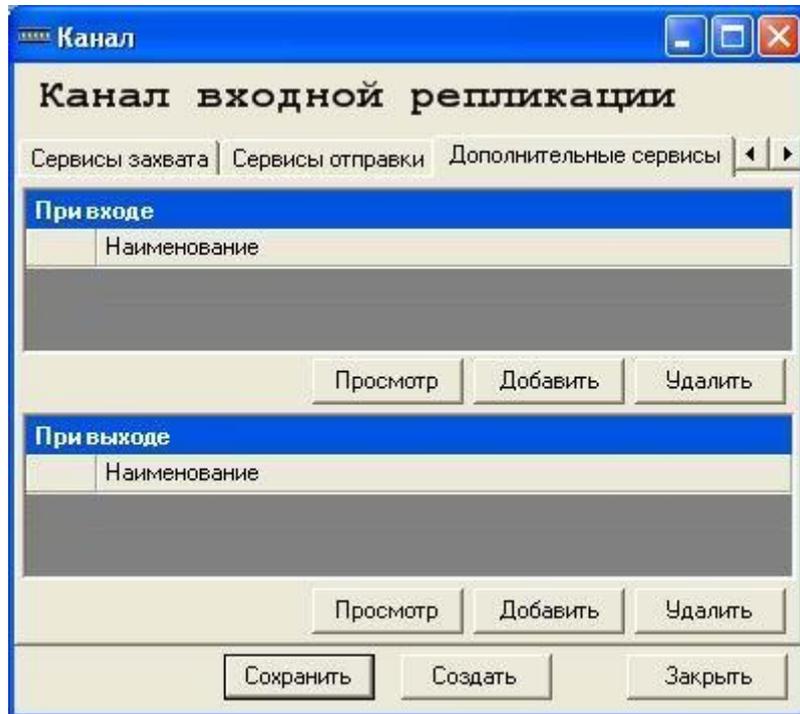


Рис. 3.35. Канал. Дополнительные сервисы

На рис. 3.35 показано содержимое вкладки «Дополнительные сервисы»:

- Таблица «При входе» – таблица с наименованиями дополнительных сервисов, которые вызываются до проверки документа на соответствие схеме входящего документа.
- Таблица «При выходе» – таблица с наименованиями дополнительных сервисов, которые вызываются до отправки документа в сервисы отправки.
- Кнопки «Просмотр» – при нажатии на эту кнопку открывается форма редактирования выделенного в таблице дополнительного сервиса.
- Кнопки «Добавить» – при нажатии на эту кнопку открывается форма со списком существующих дополнительных сервисов системы, можно выбрать сервисы из этого списка или создать новые.
- Кнопки «Удалить» – при нажатии на эту кнопку задается вопрос, действительно ли Вы хотите удалить данный дополнительный сервис из списка этого канала. Если ответ отрицательный, то дополнительный сервис остается в списке, а если положительный, то удаляется связь между данным каналом и выбранным дополнительным сервисом, при этом сам дополнительный сервис остается в системе.

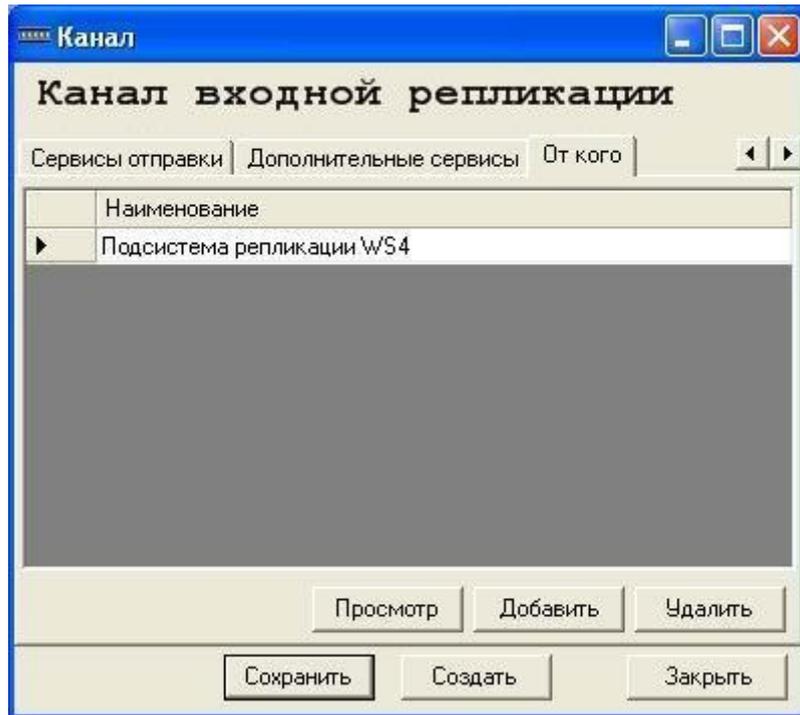


Рис. 3.36. Канал. От кого

На рис. 3.36 показано содержимое вкладки «От кого»:

- Таблица – таблица с наименованиями приложений, от которых данный канал получает документы.
- Кнопка «Просмотр» – при нажатии на эту кнопку открывается форма редактирования выделенного в таблице приложения.
- Кнопка «Добавить» – при нажатии на эту кнопку открывается форма со списком существующих приложений системы, можно выбрать приложения из этого списка или создать новые.
- Кнопка «Удалить» – при нажатии на эту кнопку задается вопрос, действительно ли Вы хотите удалить данное приложение из списка этого канала. Если ответ отрицательный, то приложение остается в списке, а если положительный, то удаляется связь между данным каналом и выбранным приложением, при этом само приложение остается в системе.

На форме редактирования сервиса отправки в верхней части отображается наименование сервиса отправки и имеется пять вкладок.

На рис. 3.37 показаны поля вкладки «Общие данные»:

- Наименование – обязательное поле, содержит уникальное в системе наименование объекта сервиса отправки. Если попытаться сохранить объект с наименованием, которое присутствует в системе, то выдается сообщение об этой ошибке, данные не сохраняются и окно редактирования не закрывается.

- Описание – необязательное поле, содержит полное описание сервиса отправки. Длина описания не должна превышать 2000 символов. Если попытаться сохранить объект с описанием, длина которого превышает 2000 символов, то выдается сообщение об этой ошибке, данные не сохраняются и окно редактирования не закрывается.
- Включен – обязательное поле, указывает состояние сервиса отправки. Если галка стоит, то сервис отправки работает и документы проходят через него обработку, в противном случае, документы не отправляются в данный сервис отправки.

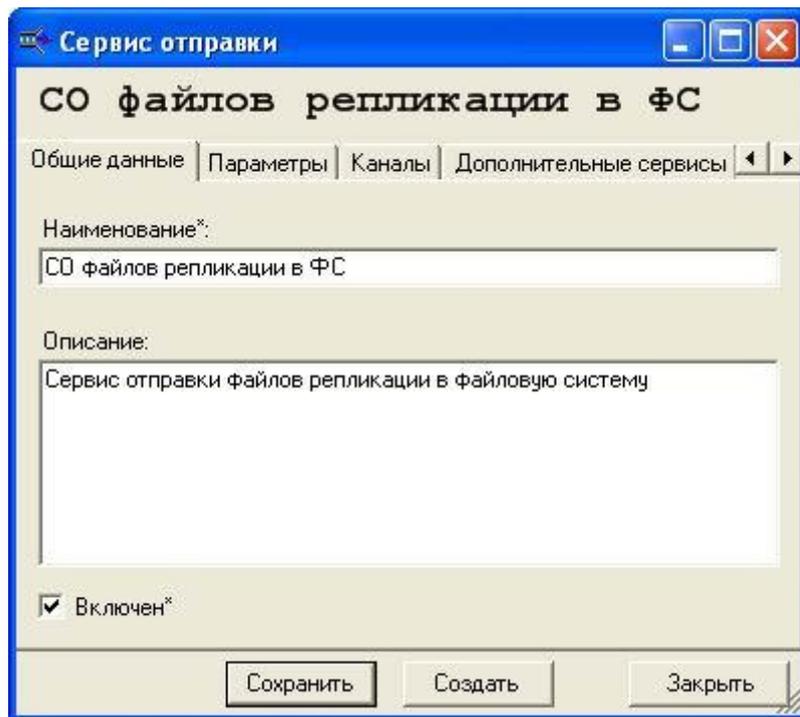


Рис. 3.37. Сервис отправки. Общие данные

На рис. 3.38 показаны поля вкладки «Параметры»:

- Функция отправки – обязательное поле, указывающее, с помощью какой функции отправки данный сервис отправки передает документы во внешний мир. Есть возможность выбрать другую функцию отправки нажатием кнопки .
- Параметры – необязательно поле с параметрами для настройки функции отправки. Есть возможность вызвать форму для настройки выбранной функции отправки нажатием кнопки .
- Кол-во повторов – обязательное поле, указывающее количество попыток отправки документа перед перемещением его в очередь приостановленных документов.
- Интервал – обязательное поле, указывающее интервал между попытками

отправки документа.

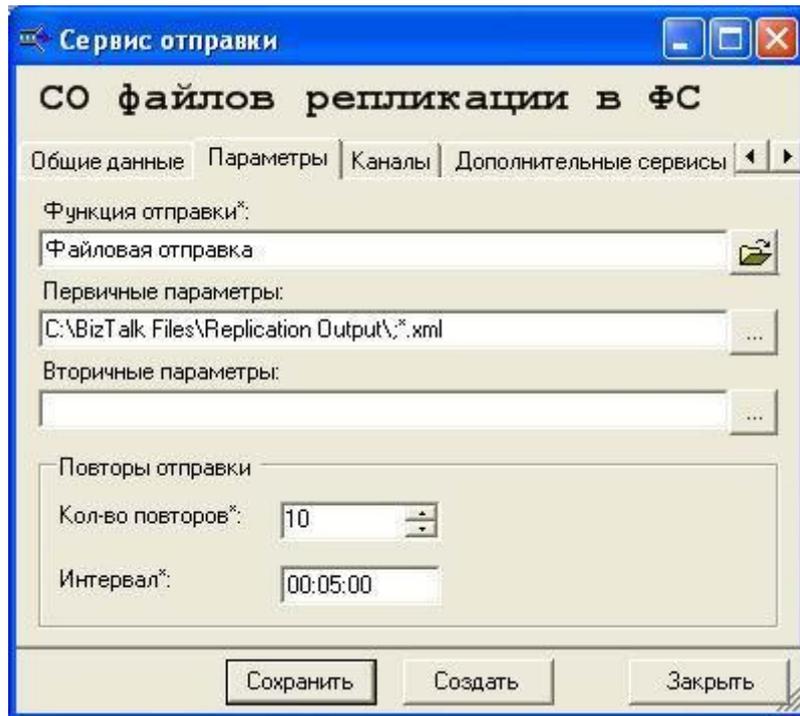


Рис. 3.38. Сервис отправки. Параметры

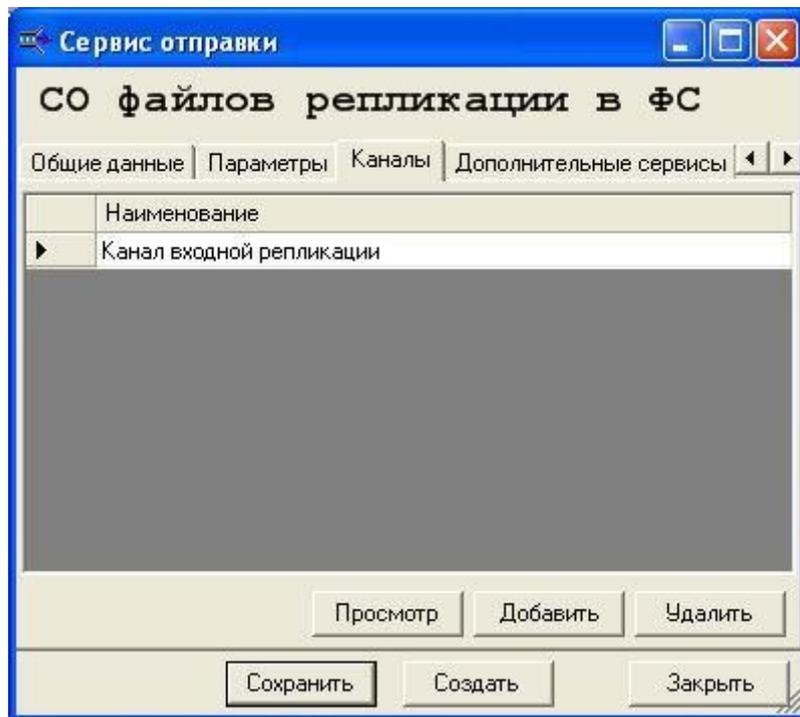


Рис. 3.39. Сервис отправки. Каналы

На рис. 3.39 показано содержимое вкладки «Каналы»:

- Таблица – таблица с наименованиями каналов, от которых получает документы данный сервис отправки.

- Кнопка «Просмотр» – при нажатии на эту кнопку открывается форма редактирования выделенного в таблице канала.
- Кнопка «Добавить» – при нажатии на эту кнопку открывается форма со списком существующих каналов системы, можно выбрать каналы из этого списка или создать новые.
- Кнопка «Удалить» – при нажатии на эту кнопку задается вопрос, действительно ли Вы хотите удалить данный канал из списка этого сервиса отправки. Если ответ отрицательный, то канал остается в списке, а если положительный, то удаляется связь между данным сервисом отправки и выбранным каналом, при этом сам канал остается в системе.

На рис. 3.40 показано содержимое вкладки «Дополнительные сервисы»:

- Таблица – таблица с наименованиями дополнительных сервисов, которые вызываются во время обработки документа в данном сервисе отправки.
- Кнопка «Просмотр» – при нажатии на эту кнопку открывается форма редактирования выделенного в таблице дополнительного сервиса.
- Кнопка «Добавить» – при нажатии на эту кнопку открывается форма со списком существующих дополнительных сервисов системы, можно выбрать сервисы из этого списка или создать новые.
- Кнопка «Удалить» – при нажатии на эту кнопку задается вопрос, действительно ли Вы хотите удалить данный дополнительный сервис из списка этого сервиса отправки. Если ответ отрицательный, то дополнительный сервис остается в списке, а если положительный, то удаляется связь между данным сервисом отправки и выбранным дополнительным сервисом, при этом сам дополнительный сервис остается в системе.

На рис. 3.41 показано содержимое вкладки «Кому»:

- Таблица – таблица с наименованиями приложений, которым данный сервис отправки отправляет документы.
- Кнопка «Просмотр» – при нажатии на эту кнопку открывается форма редактирования выделенного в таблице приложения.
- Кнопка «Добавить» – при нажатии на эту кнопку открывается форма со списком существующих приложений системы, можно выбрать приложения из этого списка или создать новые.
- Кнопка «Удалить» – при нажатии на эту кнопку задается вопрос, действительно ли Вы хотите удалить данное приложение из списка этого сервиса отправки. Если ответ отрицательный, то приложение остается в списке, а если положительный, то удаляется связь между данным сервисом отправки и выбранным приложением, при этом само приложение остается в системе.

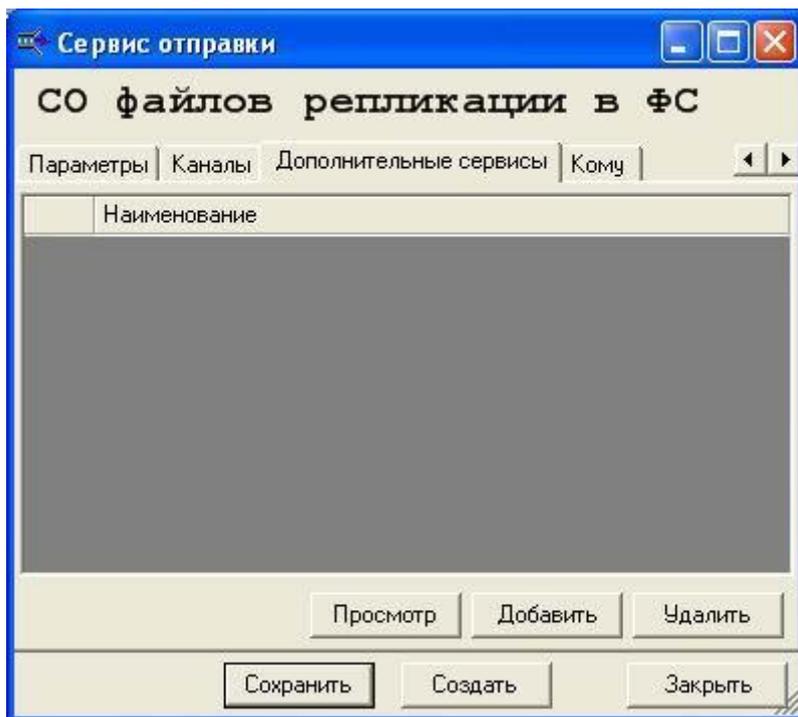


Рис. 3.40. Сервис отправки. Дополнительные сервисы

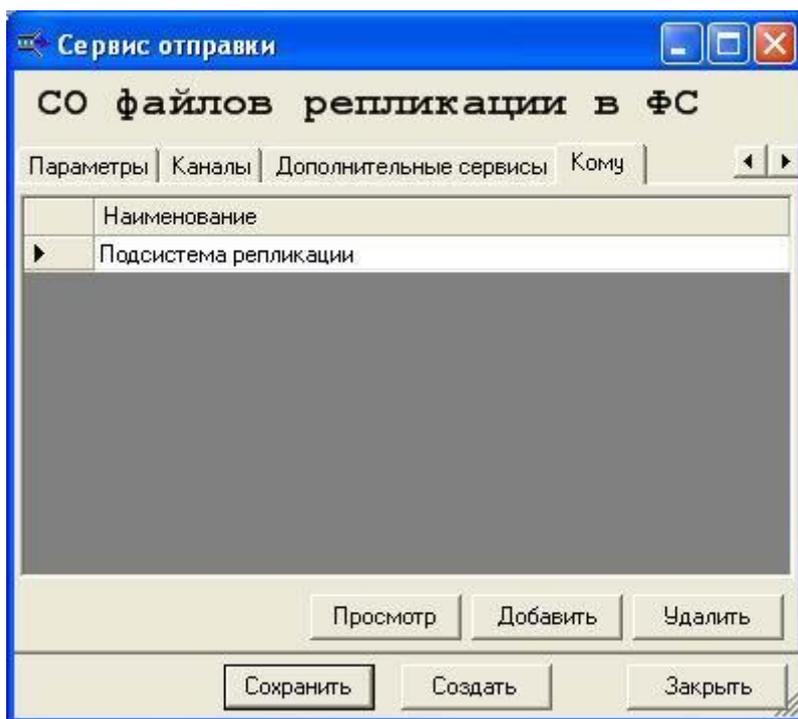


Рис. 3.41. Сервис отправки. Кому

На форме редактирования схемы документа в верхней части отображается наименование схемы документа и имеется единственная вкладка.

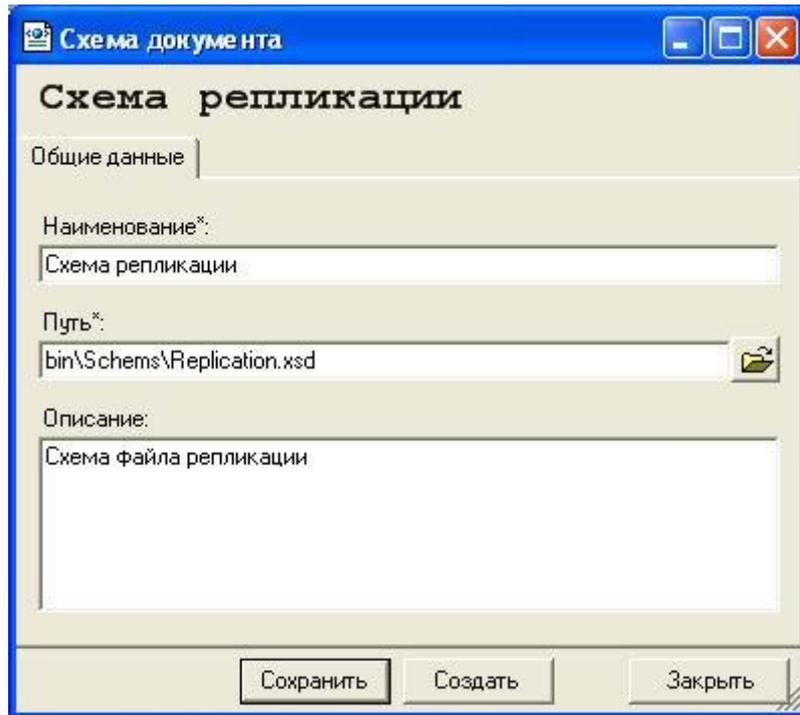


Рис. 3.42. Схема документа. Общие данные

На рис. 3.42 показаны поля вкладки «Общие данные»:

- Наименование – обязательное поле, содержит уникальное в системе наименование объекта схемы документа. Если попытаться сохранить объект с наименованием, которое присутствует в системе, то выдается сообщение об этой ошибке, данные не сохраняются и окно редактирования не закрывается.
- Путь – обязательное поле, указывающее путь до файла, представляющего данную схему. Можно выбрать путь с помощью диалога выбора файла, нажав на кнопку .
- Описание – необязательное поле, содержит полное описание схемы документа. Длина описания не должна превышать 2000 символов. Если попытаться сохранить объект с описанием, длина которого превышает 2000 символов, то выдается сообщение об этой ошибке, данные не сохраняются и окно редактирования не закрывается.

На форме редактирования карты преобразования документа в верхней части отображается наименование карты преобразования и имеется единственная вкладка.

На рис. 3.43 показаны поля вкладки «Общие данные»:

- Наименование – обязательное поле, содержит уникальное в системе наименование объекта карты преобразования. Если попытаться сохранить объект с наименованием, которое присутствует в системе, то

выдается сообщение об этой ошибке, данные не сохраняются и окно редактирования не закрывается.

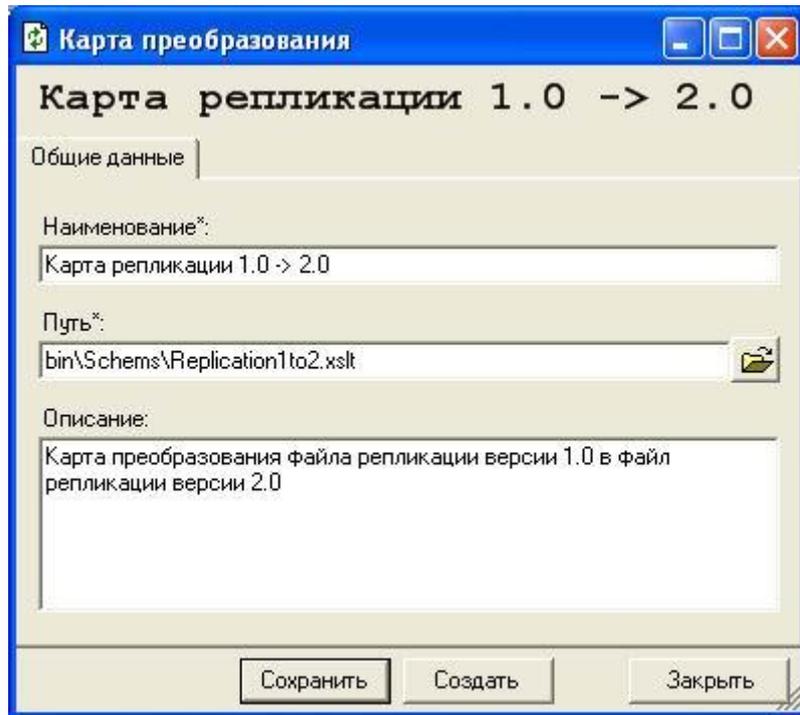


Рис. 3.43. Карта преобразования. Общие данные

- Путь – обязательное поле, указывающее путь до файла, представляющего данную карту преобразования. Можно выбрать путь с помощью диалога выбора файла, нажав на кнопку .
- Описание – необязательное поле, содержит полное описание карты преобразования. Длина описания не должна превышать 2000 символов. Если попытаться сохранить объект с описанием, длина которого превышает 2000 символов, то выдается сообщение об этой ошибке, данные не сохраняются и окно редактирования не закрывается.

СОДЕРЖАНИЕ

1. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПРОГРАММЫ, ОБЛАСТЬ ЕЕ ПРИМЕНЕНИЯ, ЕЕ ОГРАНИЧЕНИЯ	2
1.1. Назначение программы	2
1.2. Область применения программы	5
1.3. Ограничения использования программы.....	6
2. ТЕХНИЧЕСКОЕ ОПИСАНИЕ ПРОГРАММЫ.....	6
2.1. Модель интеграции.....	6
2.2. Документ BizTalk	10
2.3. Структура программного продукта.....	11
Интерфейсы интеграции	14
Компоненты интеграции.....	15
База данных для хранения информации SBTServer.....	19
Работа с объектами документооборота SBTM_Objects	21
Работа с очередями документов системы SBTM_Queues	24
Приложение для визуальной работы с объектами документооборота SBTServer.exe	26
Приложение для администрирования сервера SBTAdmin.exe	27
Ядро сервера SBTM_Scheduler.exe	29
2.3. Применяемые программные средства	32
Выбор операционной системы	32
Выбор компонентной технологии	32
Выбор системы программирования.....	33
Выбор СУБД и метода доступа к данным.....	33
2.4. Используемые технические средства и требования к аппаратуре.....	34
3. СПЕЦИАЛЬНЫЕ УСЛОВИЯ ПРИМЕНЕНИЯ И ТРЕБОВАНИЯ ОРГАНИЗАЦИОННОГО, ТЕХНИЧЕСКОГО И ТЕХНОЛОГИЧЕСКОГО ХАРАКТЕРА	35
4. УСЛОВИЯ ПЕРЕДАЧИ ПРОГРАММНОЙ ДОКУМЕНТАЦИИ ИЛИ ЕЕ ПРОДАЖИ.....	35
Библиографический список.....	36
Список источников в INTERNET	38
Приложение 1. Основные обозначения и сокращения.....	39
Приложение 2. Модель базы данных системы	40
Приложение 3. Руководство администратора	44
Установка системы.....	44
Настройка конфигурационных файлов	51
Документация по работе с SBTServer.exe - приложением для визуальной работы с объектами документооборота	53