На правах рукописи

Вин Зо

МОДЕЛИРОВАНИЕ ПРОЦЕССОВ ИНТЕГРАЦИИ ИНФОРМАЦИОННЫХ СИСТЕМ

Специальность 05.13.11 — математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
кандидата технических наук

Автор:

Москва – 2007

Работа выполнена в Московском инженерно-физическом институте (государственном университете).

Научный руководитель: доктор технических наук, профессор Щукин Борис Алексеевич

Официальные оппоненты: доктор технических наук, профессор Александров Владимир Михайлович

кандидат технических наук, с.н.с Филиппов Вячеслав Алексеевич

Ведущая организация: Московский государственный институт электронной техники (технический университет)

Защита состоится 23 мая 2007 г. в 15:30 часов на заседании диссертационного совета Д 212.130.03 в МИФИ по адресу: 115409, Москва, Каширское шоссе, д.31, телефон 323-91-62 в конференц-зале главного корпуса.

С диссертацией можно ознакомиться в библиотеке МИФИ.

Автореферат разослан _____ апреля 2007 г.

Отзывы на автореферат в двух экземплярах, заверенные печатью организации, просьба направлять по адресу:
115409, Москва, Каширское шоссе, д.31, диссертационный совет, Шумилову Ю.Ю.

Ученый секретарь
диссертационного совета,
д.т.н., профессор Шумилов Ю.Ю.

Общая характеристика работы

Актуальность

Борьба со сложностью предметной области осуществляется с помощью декомпозиции последней. На ранних стадиях развития информационных технологий при автоматизации бизнес процессов предметной области создавали локально функционирующие подсистемы, которые образовывали несвязанные островки автоматизации. В настоящее время наметилась тенденция создавать отдельные бизнес сервисы, имея в виду связывать их в рамках распределенной системы, поддерживающей требующуюся функциональность.

Интеграция всех реально функционирующих приложений, как вновь создаваемых, так и приложений и подсистем со стажем является актуальной задачей информационной технологии. Для решения этой задачи унаследованные приложения заключаются в оболочки, превращающие их в сервисы, доступные для использования другими приложениями. В большинстве случаев интеграция основана на передаче сообщений: специальные «информационные брокеры», берут на себя роль посредников в обмене сообщениями между приложениями. В настоящий момент на рынке предлагаются целые платформы интеграции, маркетинговые материалы по которым обещают решить все имеющиеся проблемы, и заложить основу для решения проблем будущего.

Интеграция – это не единовременный процесс. Практически всегда будет существовать потребность присоединения к действующей распределенной системе еще одного приложения. Так как это должно осуществляться на «ходу», существенно возрастает роль моделирования и тестирования процессов взаимодействия распределенной системы с вновь присоединяемым приложением.

Таким образом, актуальность выбранной темы определяется тенденцией разработки программных комплексов информационных систем как композиций сервисов (SAAS –

Software As A Service). Роль моделирования и тестирования при интеграции получающихся сервисов в распределенную информационную систему, поддерживающую бизнес процессы предметной области, будет только возрастать.

Цели и задачи работы

Целью диссертационной работы является исследование и разработка методов и инструментальных программных средств для моделирования процессов взаимодействия локальной подсистемы с распределенной информационной системой, возникающих при их интеграции.

В ходе исследования решены следующие задачи

1. Проанализированы современные технологии интеграции информационных систем.
2. Проанализированы роль и место тестирования в процессе интеграции.
3. Разработана методика моделирования процесса интеграции на основе использования информационно-программных эмуляторов интегрируемых подсистем.
4. Разработан эмулятор локальной платежной сети, на котором продемонстрированы разработанные методы на примере моделирования процесса интеграции процессинговых систем.
5. На основе использования сетей Петри разработана модель, отражающая логику формирования и посылку сообщения-запроса подсистемой-клиентом и получения сообщения-ответа от подсистемы-сервера.

Методы исследования

Для решения поставленных задач в диссертации использованы методы моделирования программных систем с помощью сетей Петри, методы и средства объектно-ориентированного проектирования и программирования. Для реализации экспериментального приложения использованы средства реляционных и постреляционных баз данных и XML-технологии.

Научная новизна и практическая значимость работы

Научная новизна данного исследования состоит в том что:

1. Разработана методика моделирования процесса интеграции на основе создания программных эмуляторов интегрируемых подсистем и специально настроенной тестовой базы, используемой для генерации сообщений в соответствии с протоколом взаимодействия подсистем.
2. Разработаны модели и средства для построения программной и информационной компоненты, используемые для моделирования процесса интеграции информационных подсистем.

Практическая значимость работы заключается в том, что:

1. Разработан эмулятор локальной платежной сети, заменяющий ее при проведении анализа взаимодействия (тестирования) процессинговой системы банка с локальной платежной сетью. Взаимодействие организуется путем обмена сообщениями по специальному протоколу, построенному на основе стандарта ISO 8583. Эмулятор разработан на языке Visual Basic 6.0 с использованием интерфейса прикладного программирования Winsock и СУБД MS Access. Доступ к данным в эмуляторе выполняется с помощью технологии ADO.
2. Для проведения тестирования с использованием эмулятора разработана схема тестовой базы данных. Семантически база данных определяет иерархическую структуру, позволяющую задавать сбалансированные тестовые ситуации для клиента (Acquire) и сервера (Issuer), а синтаксически это совокупность реляционных таблиц, связанных друг с другом системой ссылок.
3. Для формирования тестовых ситуаций разработана программа-редактор тестовой базы данных.
4. Разработана сеть Петри взаимодействия процессинговой системы с локальной платежной сетью, обменивающихся сообщениями. Сеть Петри разработана с использованием среды CPN Tools.

Достоверность разработанных результатов подтверждается как экспериментами на моделях интеграции, когда

взаимодействующие системы заменены их эмуляторами, так и экспериментами с заменой эмулятором одной из систем.

Реализация и внедрение результатов работы:

Разработанные в диссертации методики и модели процесса интеграции на основе использования информационно-программных эмуляторов интегрируемых систем использованы в учебном процессе кафедры «Кибернетика» МИФИ в курсе для студентов Союза Мьянма «Современные методы и средства проектирования информационных систем» и подготовке диссертаций на соискание степени магистра.

Апробация работы:

Основные результаты диссертации докладывались и обсуждались на ежегодных научных сессиях МИФИ (2004, 2005, 2006, 2007 гг.), на международном научно-техническом семинаре «Современные технологии в задачах управления, автоматики и обработки информации» (2005, 2006 гг.) и на всероссийской межвузовской научно-технической конференции студентов и аспирантов, МИЭТ (2007 гг.).

Основные результаты диссертации опубликованы в тезисах докладов на научных сессиях МИФИ, международном научно-техническом семинаре «Современные технологии в задачах управления, автоматики и обработки информации» и на всероссийской межвузовской научно-технической конференции студентов и аспирантов, МИЭТ.

Публикации:

Основные положения диссертационной работы опубликованы в 5 печатных работах.

Структура и объем работы:

Диссертация содержит 4 главы, введение и заключение, 37 рисунков и 15 таблиц.

Общий объем – 95 страниц машинописного текста. Список использованных источников содержит 44 наименования.

Содержание работы

Во введении обосновывается актуальность темы диссертационной работы и приводится ее краткая характеристика. Формулируются цель работы и задачи исследования. Представляются основные положения, выносимые на защиту.

В первой главе приведен обзор технологий интеграции информационных систем и рассмотрены роль и место моделирования и тестирования в процессе интеграции. Процесс интеграции локальной системы в распределенную информационную систему рассмотрен на примере присоединения нового банка к международной платежной сети.

В настоящее время интеграция информационных систем, в большинстве случаев, подразумевает обмен сообщениями в режиме Remote Procedure Call (RPC) на базе использования систем гарантированной доставки сообщений (например, IBM MQSeries), хотя не исключается и обмен файлами. Если необходимо связать только две подсистемы, то подход типа «точка-точка» является вполне приемлемым. Если же требуется интегрировать локальную подсистему или отдельное приложение в распределенную информационную систему, то в последней обычно существует специальный компонент, через который осуществляется интеграция. Часто это специальная обслуживающая подсистема, называемая интеграционным брокером, который получает сообщения от многих подсистем и направляет их соответствующим адресатам. В современном варианте - это корпоративная сервисная шина (ESB - Enterprise Service Bus). Например, при интеграции банка в платежную сеть роль такого брокера выполняет специальная функция в процессинговой системе головного банка сети.

На сегодняшний день брокеры сообщений могут объединять большое количество взаимодействующих подсистем, образуя по определению компании Gartner Group «корпоративную нервную систему».

В настоящее время для интеграции информационных систем все чаще прибегают к помощи web-сервисов. Это очень перспективный подход к интеграции, который связан с использованием открытых стандартов, разрабатываемых органом

стандартизации Интернет сообщества в лице консорциума W3C и организацией по разработке и принятию промышленных стандартов электронной коммерции - OASIS. В основе этого подхода лежит XML - мета-язык для представления данных. XML является такой же универсальной и базовой технологией для представления, трансформации и обмена данными, как транспортный протокол TCP/IP для Интернета.

Преимуществом этого подхода является то, что он предлагает методологически единое решение как для интеграции в пределах предприятия (EAI - Enterprise Application Integration), так и для B2B-интеграции между предприятиями.

Ниже перечислены основные принципы применения web-сервисов для организации взаимодействия подсистем в децентрализованной, распределенной среде:

- В качестве основного механизма интеграции используется web-сервис.
- В качестве стандарта обмена данными используется XML. Осуществляется «слабое связывание» информационных подсистем на основе пересылки сообщений в виде XML-документов.
- Организуется отраслевой, региональный или корпоративный регистр для публикации web-сервисов. Эти регистры создаются на базе стандарта UDDI (OASIS).

Каково место моделирования и тестирования в процессе интеграции?

В самом простом случае, когда принимается решение об интеграции двух локальных подсистем, требуется выработать подробный протокол обмена сообщениями между подсистемами. Именно на этом этапе разработка серии моделей взаимодействия подсистем в виде раскрашенных сетей Петри может оказать разработчикам существенную помощь. Построение таких моделей позволяет лучше понять порядок и особенности взаимодействия подсистем и, тем самым, более качественно провести их интеграцию. Модель на базе раскрашенной сети Петри это не только статическая конструкция, подобная диаграммам UML и

показывающая связи между отдельными элементами. Имеются специальные среды моделирования, которые позволяют проводить наглядную имитацию процесса взаимодействия. Использование исходной «разметки» сети позволяет анализировать различные сценарии взаимодействия подсистем.

Роль и место тестирования при интеграции некоторой локально функционирующей подсистемы с группой подсистем, интегрированных в единую систему, можно проследить на примере подсоединения нового банка к международной платежной сети. Предполагается, что новый банк имеет достаточно разветвленную инфраструктуру в виде сети банкоматов и POS-терминалов, опыт эмиссии пластиковых карт и достаточное число держателей этих карт. В противном случае ему нет необходимости подключаться к платежной сети, так как это достаточно дорогая процедура.

Несмотря на то, что процессинговая система и вся инфраструктура нового банка проверена и работает, согласован до мелочей стандартный протокол взаимодействия сети и банка, эксперты платежной сети проводят тщательное тестирование этого взаимодействия, прежде чем дать разрешение на подсоединение банка к сети. При этом тестировании платежная сеть заменяется специальным программным эмулятором, а для тестирования разрабатывается специальная тестовая база данных.

В диссертации разрабатывается методика создания упрощенных программных моделей интегрируемых подсистем и отработки на этих моделях, как параметров интеграции, так и всех проблемных ситуаций, возникающих в процессе интеграции. Предполагается, что взаимодействие интегрируемой подсистемы и эмулятора осуществляется путем посылки сообщений-запросов и приема сообщений-ответов, генерируемых на основе специально разрабатываемой тестовой базы данных.

Во второй главе рассмотрены модели, используемые при моделировании процессов интеграции.

Прежде всего, проанализированы модели взаимодействия интегрируемых подсистем на базе раскрашенных сетей Петри, потенциально полезных при отработке протоколов взаимодействия подсистем, отработке сценариев взаимодействия и т.д. Рассмотрено

взаимодействие двух информационных подсистем, при котором одна из них работает в режиме клиента, то есть инициирует запросы к другой подсистеме, а вторая – в режиме сервера, то есть отвечает на поступающие запросы.

На рис.1 представлена сеть Петри, отражающая логику формирования и посылку сообщения-запроса подсистемой-клиентом и получения сообщения-ответа от сервера.

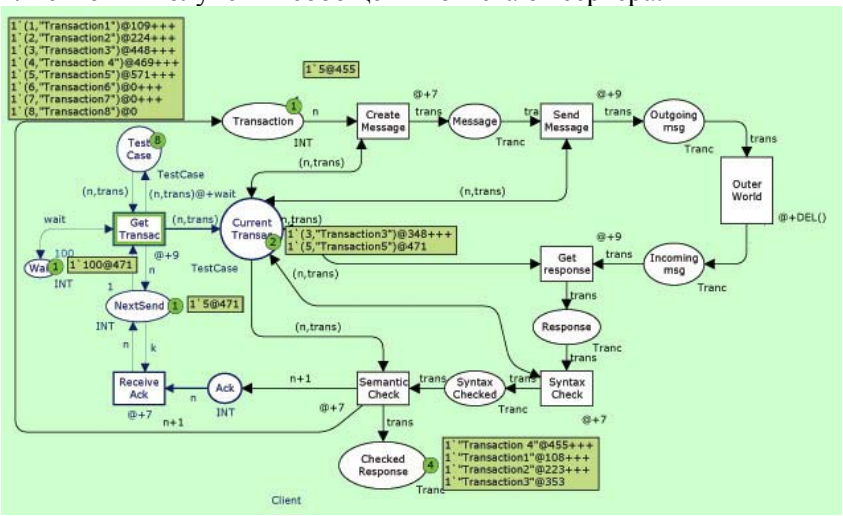

Рис.1.Сеть Петри, моделирующая последовательный алгоритм обработки сообщений подсистемой-клиентом.

Эта сеть моделирует логику подсистемы-клиента в автономном режиме: весь «внешний мир», то есть подсистема-сервер, моделируется задержкой «фишки» на несколько тактов. Практически, это модель, отражающая логику работы «движка», который может быть использован в эмуляторе клиента при моделировании взаимодействия интегрируемых подсистем в стиле RPC. Конкретную сеть, которая будет отражать особенности конкретной подсистемы, то есть будет «более» раскрашенной, можно строить только для конкретных подсистем.

На рис.2 представлена сеть Петри, отражающая логику формирования и посылку сообщения-ответа подсистемой-сервером

в ответ на сообщение-запрос от клиента.

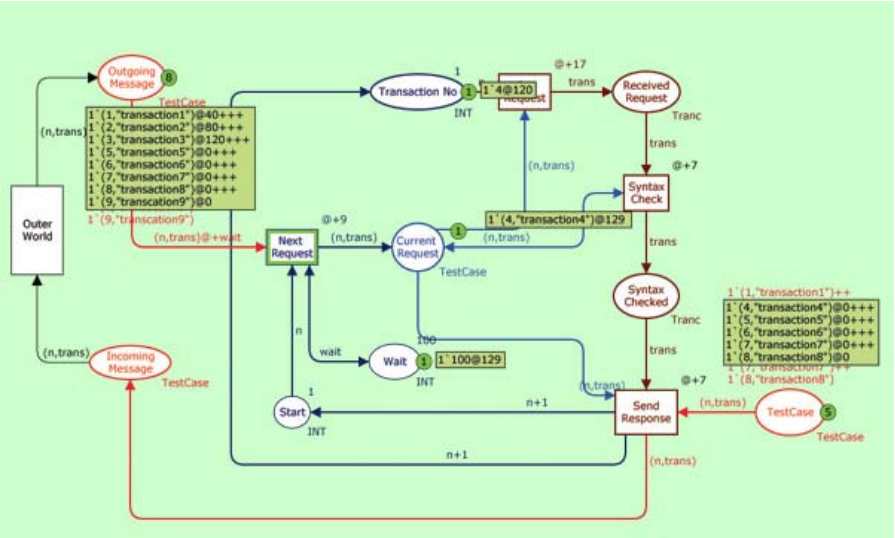

Рис. 2. Сеть Петри, моделирующая последовательный алгоритм обработки сообщений подсистемой-сервером.

Эти две сети – простейшие модели подсистемы-клиента и подсистемы, выступающей в роли сервера, будучи замкнуты друг на друга, образуют иерархическую временную сеть, отражающую процесс взаимодействия двух подсистем.

Сети Петри стандартно строятся для отражения логики работы моделируемых систем. Для моделирования процессов в распределенных системах с учетом развития процесса во времени, с помощью глобальных часов в сеть вводится временной механизм, а к «фишкам» прикрепляются специальные «штампы». Временной штамп «фишки» назначается в начальной разметке и увеличивается с помощью выражений на переходах или дугах. Этот механизм делает «фишку» способной к активации перехода, если ее штамп на этом шаге оказывается меньше значения счетчика глобальных часов. Глобальные часы увеличивают свое значение в случае, если не разрешен ни один переход сети. Моделирование с учетом

времени существенно усложняет сеть Петри, фактически приближая ее к имитационным моделям, однако это позволяет оценить различные константы в протоколах взаимодействия типа количества повторных запросов, таймаутов и т.д.

Вторая часть главы посвящена методам построения программы-эмулятора и формированию связанной с ней тестовой базы данных. Если проверяется функциональность информационной подсистемы-сервера, то вместо клиента, целесообразно использовать программный эмулятор. Наоборот, если подсистема проверяется в роли клиента, то сервер заменяется эмулятором. Соответствующие эмуляторы будут разными.

Программа-эмулятор – это достаточно сложная программа, которую целесообразно строить на основе взаимодействия процессов. Фактически - это целый программный комплекс, предназначенный для решения конкретной задачи моделирования. Каждый из процессов использует для решения задачи либо свои собственные данные и обменивается с другими процессами только результатом своей работы, либо работает с общей областью данных, разделяемых между разными процессами.

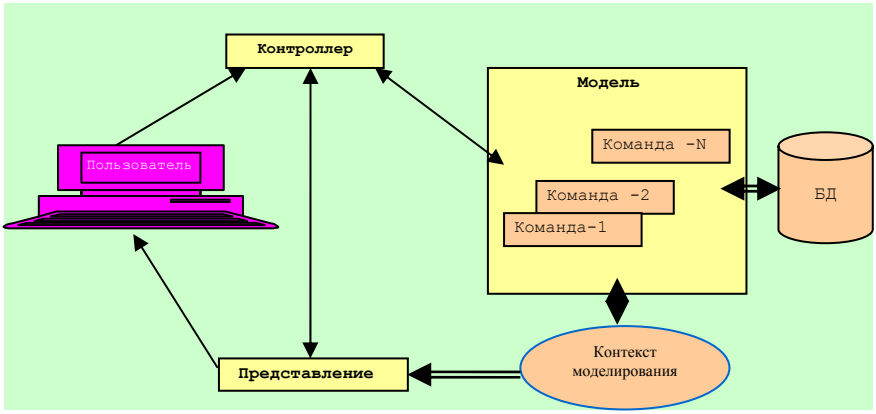

Рис. 3. Архитектура эмулятора в стиле шаблона MVC.

В основу архитектуры программы-эмулятора целесообразно положить шаблон Model-View-Controller (MVC). В этой архитектуре (Рис. 3) контроллер (Controller) отвечает за логику моделирования, то есть за последовательность действий эмулятора,

различные команды модели (Model) – за те или иные функциональные преобразования, а представление (View) – за вывод пользователю результатов моделирования.

Эмуляторы, которые предлагается использовать при тестировании процессов интеграции, имеют достаточно упрощенную «модель». Если ориентироваться на обмен XML-сообщениями, то многие «команды модели» будут стандартны. Контроллер программы также достаточно универсален. Все особенности протокола взаимодействия подсистем сосредоточены в тестовой базе данных.

Таким образом, именно в базе тестовых данных сосредоточены все особенности взаимодействия подсистем в соответствии с принятым протоколом, который принимается в процессе разработки проекта интеграции. Речь идет о прикладном протоколе, в стандартном варианте это некоторая последовательность сообщений-запросов и сообщений-ответов. При этом сообщению некоторого типа, инициированному одной подсистемой, ставится в соответствие определенный тип сообщения второй подсистемы.

Последовательность сообщений образует некоторую ситуацию моделирования, направленную на отработку той или иной реальной ситуации. Ситуаций при интеграции подсистем приходится отрабатывать достаточно много, при этом новые ситуации появляются и в процессе эксплуатации и в процессе модернизации подсистем. Такие ситуации моделирования принято называть тесткейсами, так как очень часто моделирование проводится с целью тестирования или отработки некоторого функционального элемента информационной подсистемы.

При подготовке процесса моделирования целесообразно иметь дело не с реальными сообщениями, полностью готовыми к отправке, а с их специально разрабатываемыми шаблонами, которые используются для автоматической генерации реальных сообщений. Формирование очередного i-ого сообщения в программе эмуляторе клиента производится на базе шаблона i-ого сообщения, хранящегося в текущем тесткейсе. Какая информация доступна в этот момент:

- Шаблон сообщения;
- Последнее отправленное сообщение-запрос (request);
- Сообщение-ответ на последнее отправленное сообщение-запрос (response);
- Тестовая база данных интегрируемых подсистем.

Поля сообщения заполняются в следующем порядке:

1. Формируются идентификаторы основных объектов, так как эти идентификаторы являются ключами соответствующих таблиц базы данных (например, определяется номер кредитной карты);
2. Заполняются все поля, однозначно определяемые данными идентификаторами;
3. Заполняются все остальные поля.

Шаблон сообщения-запроса содержит определение каждого поля i-ого сообщения:

- Literal; Конкретное значение, задаваемое пользователем (например, конкретный номер кредитной карты).
- (<, >, <=, >=, #, =) last request; Значение, формируемое в соответствии со значением поля последнего отправленного запроса. Для знака «=» используется константа «Copy from request»
- (<, >, <=, >=, #, =) response; Значение, формируемое в соответствии со значением поля последнего полученного ответа. Для знака «=» используется константа «Copy from response»
- (<, >, <=, >=, #, =) (name_table,key,atr_name);
  name_table – имя таблицы базы данных;
  key – имя (номер) поля формируемого сообщения, значение которого уже сформировано и представляет идентификатор записи таблицы name_table.
  atr_name – атрибут таблицы базы данных. Значение, формируемое в соответствии со

значением атрибута atr_name таблицы базы данных name_table.

- (<, >, <=, >=, #, =) (field_num);
  field_num – номер (имя) поля формируемого сообщения, значение которого уже сформировано. Значение, формируемое в соответствии со значением уже сформированного поля сообщения.
- Unique Value; Значение, формируемое генератором уникальных значений.
- Valid Value; Значение, формируемое генератором случайных чисел в соответствии с форматом поля.
- Non Valid Value; Значение, не соответствующее формату поля.
- Field Omitted; Значение опущено.

Более просто, но аналогично, формируется ответ в эмуляторе сервера.

Для проведения автоматического тестирования на стороне клиента необходимо организовать проверку правильности ответов сервера. Для автоматической проверки правильности ответов сервера доступна следующая информация:

- Шаблон для проверки каждого поля сообщения-ответа;
- Все поля последнего отправленного сообщения-запроса (request);
- Разобранное по полям (то есть прошедшее синтаксическую проверку) сообщение-ответ на последнее отправленное сообщение-запрос;

Шаблон для проверки сообщения-ответа содержит определение правильности каждого поля i-ого сообщения-ответа:

- Literal; Конкретное значение, задаваемое пользователем (например, конкретный номер кредитной карты).
- (<, >, <=, >=, #, =) request; Значение поля сообщения-ответа должно сравниваться со

значением поля сообщения-запроса. Для знака «=» можно использовать константу «Copy from request»

- Valid Value; Значение поля должно входить в заданный список значений.
- Field Omitted; Поля не должно быть в ответе.
- Non Checked; Значение поля не проверяется.

Построенные по такой методике тесткейсы позволяют проводить сколь угодно длительное тестирование с автоматическим занесением в log-файл всех ошибочных ситуаций.

Третья глава диссертации посвящена технологическим вопросам реализации эмулятора. Первый вопрос, который обязательно надо решить, это вопрос о коммуникации эмулятора с удаленной интегрируемой подсистемой. С этой целью анализируется интерфейс сокетов и показывается, что интерфейс WinSock позволяет реализовать все функции, необходимые для функционирования эмулятора.

Для коммуникации эмулятора и интегрируемой подсистемы интерфейс типа сокетов предоставляет все необходимые средства для создания соединения и совместной работы с интегрируемой подсистемой. Интерфейс сокетов обеспечивает возможность взаимодействия между процессами независимо от того, выполняются они на одном компьютере или на разных. Кроме того, интерфейс предоставляет единый набор функций для работы с различными стеками протоколов. WinSock позволяет работать со всеми распространенными протоколами связи, но для поставленной задачи наиболее интересно взаимодействие по протоколу TCP/IP, используемому для создания приложений клиент - сервер.

Спецификация Winsock описывает стандарт, по которому программы windows обязаны общаться с сетями TCP/IP. Интерфейс Winsock не входит в состав windows, а реализован в виде динамически загруженной библиотеки DLL. Модуль Winsock.dll находится между стеком протоколов TCP/IP и клиентским приложением. Он управляет интерфейсом к стеку TCP/IP.

При вызове функции Winsock отводит область памяти для хранения информации о новом сокете. Настройка сокета зависит от

типа сетевого взаимодействия и от функций, выполняемых программой (клиент или сервер). Для настройки сокета на сетевую работу необходимо записать IP-адрес и номер порта удаленного компьютера в соответственную структуру. Далее все функции будут пользоваться этой информацией.

Второй, не менее важный вопрос, связан с выбором СУБД для хранения тестовой базы данных. Тестовая база данных состоит из двух частей: базы шаблонов, используемых при генерации сообщений, и тестовых таблиц, содержащих фрагменты данных интегрируемых подсистем.

Так как в современных условиях технология интеграции, в основном, ориентируется на обмен XML-сообщениями между интегрируемыми подсистемами, СУБД должна быть ориентирована на эффективную работу с текстовой информацией и хранение многозначных полей переменной длины. С этой задачей лучше всего справляются постреляционные СУБД типа Cache или D3. Возможен выбор XML базы данных. В силу того, что практически повсеместно принят в качестве стандарта обмен XML-сообщениями, целесообразно определить структуру типов сообщений с помощью DTD или XML схемы. XML схема более подходит для этих целей, так как она детально определяет форматы полей и поэтому может быть эффективно использована на этапе синтаксического анализа приходящего сообщения.

Обобщенная схема базы шаблонов тестов представлена на рис. 4. На ней каждая вертикальная связь выражает отношение типа 1:N, а каждая горизонтальная - отношение типа 1:1. Для управления данными в реализованном эмуляторе выбрана СУБД MS Access. Это связано с тем, что в выбранной предметной области (локальные платежные сети) сообщения должны быть максимально компактны – это далеко не XML. Так как все типы сообщений имеют, в общем случае, разные поля по номенклатуре и количеству, то схема базы данных определялась декомпозицией по полям стандарта ISO 8583. Этот вариант является наиболее гибким, хотя и несколько более сложным в реализации.

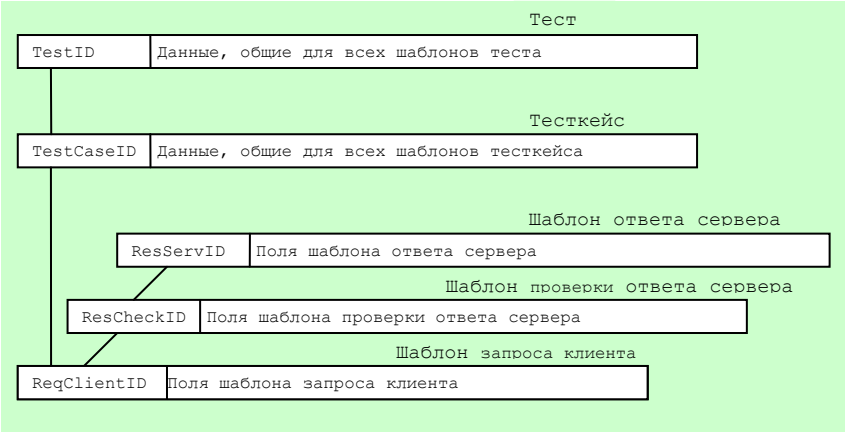

Рис.4. Иерархическая схема базы данных для хранения шаблонов сообщений.

Четвертая глава диссертации посвящена экспериментальной проверке предлагаемой методики на примере интеграции процессинговой системы банка в локальную платежную сеть. Локальные платежные сети организуются банками для проведения взаимного обслуживания пластиковых карт без выхода в международные платежные сети. Протоколы взаимодействия в локальных и международных платежных сетях существенно отличаются.

Для демонстрации предлагаемых методов был построен эмулятор локальной платежной сети, который позволяет заменить эту сеть при тестировании процессинговой системы нового банка, при его включении в локальную платежную сеть. Эмулятор может работать как в режиме клиента (Acquire), так и в режиме сервера (Issuer). На рис. 5 представлен экран для управления работой эмулятора в режиме клиента.

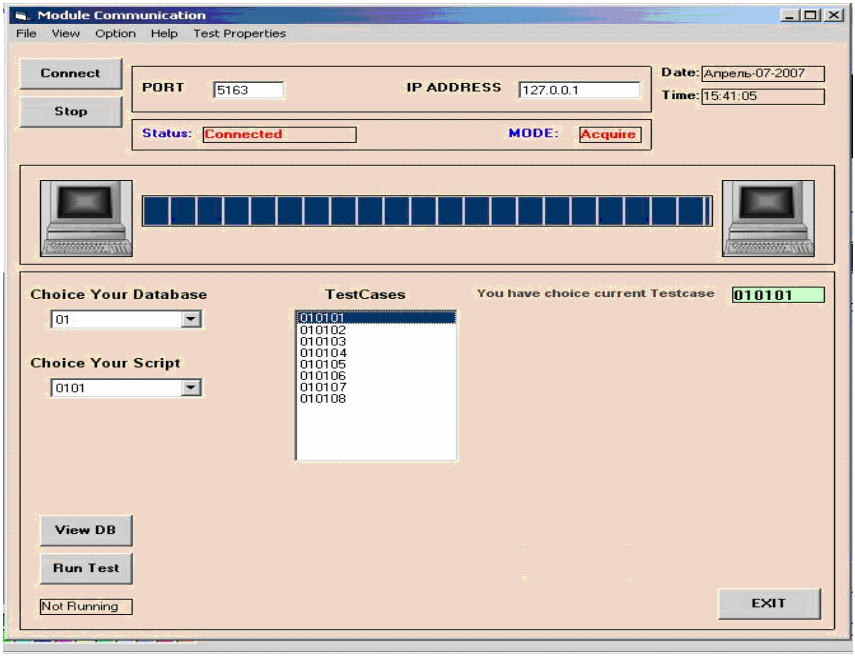

Рис. 5. Управление работой в режиме клиента (Acquire).

Архитектура эмулятора построена на базе шаблона MVC, что позволило реализовать два варианта контроллера, и общую модельную часть. Контроллеры ориентированы на функциональное и нагрузочное тестирование интегрируемых подсистем. «Модель» эмулятора выполнена в виде совокупности «команд», которые реализуют различные функциональные преобразования и извлечение тестовых данных из СУБД. При взаимодействии процессинговых систем, с целью получения максимальной компактности используются специальные структуры сообщений и специальные форматы полей (ISO 8583). Это потребовало реализации в модели специальной «команды» формирования сообщений и разработки специального редактора для формирования тестовой базы данных. На рис. 6 представлен фрагмент тесткейса, в котором представлен пример шаблона запроса на авторизацию (1100) – разрешение эмитента на выдачу запрашиваемой суммы и пример шаблона ответа эмитента (1110).

На рис.7 представлен сгенерированный запрос и полученный на него ответ.

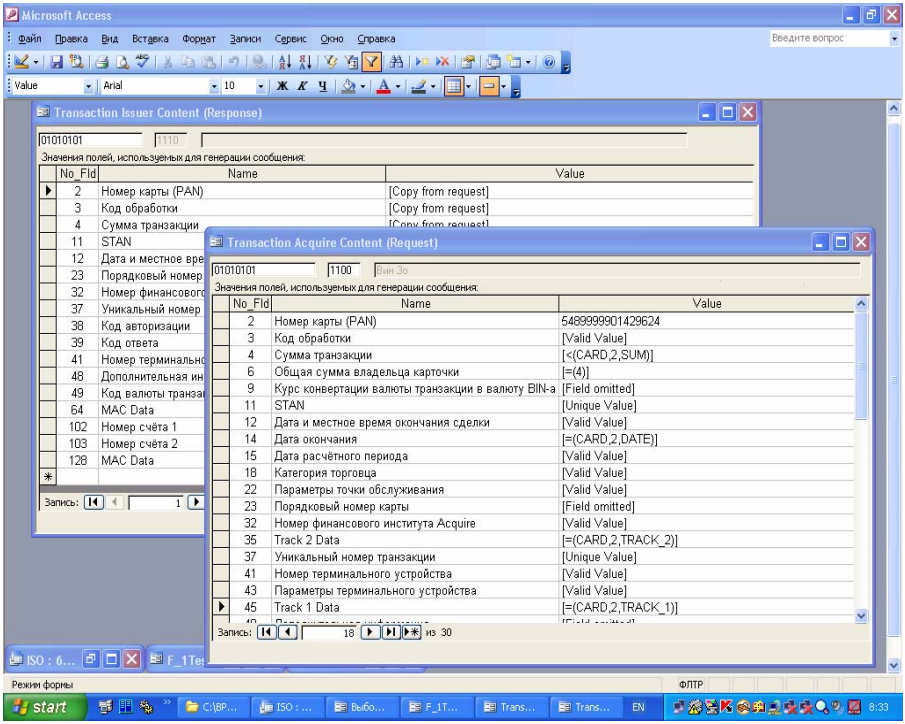

Рис.6. Вариант шаблона запроса и шаблона ответа в базе эмулятора.

Кроме эмулятора в главе представлены результаты моделирования взаимодействия двух процессинговых систем, выступающих в роли клиента и севера, полученные с использованием иерархической сети Петри. Построение раскрашенной сети Петри, моделирующей взаимодействие процессинговых систем в локальной платежной сети, позволяет выявить особенности такого взаимодействия, в частности, получить различные статистические характеристики, которые помогают выбору констант, связанных с временными таймаутами, числом повторений запросов и.т.д.

Рис.7. Фрагмент отчета о результатах тестирования.

Для сбора статистики были определены так называемые мониторинги моделирования, предусмотренные в среде CPN Tools. Обработанная информация, полученная в результате подключения мониторинга к позиции сети, из которой отправляется запрос, представлена на рис 8а и 8б.

На рис. 8а по горизонтальной оси отложена вероятность потерь сообщения в процентах, а по вертикальной – среднее время обслуживания одного сообщения.

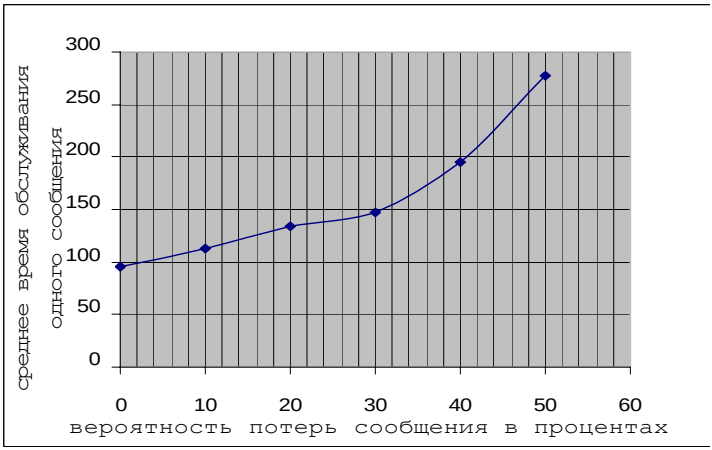

Рис.8,а. Среднее время обслуживания сообщения в зависимости от потерь сообщений в сети.

На рис. 8а по горизонтальной оси отложена вероятность потерь сообщения в процентах, а по вертикальной – число повторений сообщения.

Рис.8,б. Количество повторных обращений в зависимости от потерь сообщений в сети.

Основные результаты работы

При выполнении данной работы получены следующие основные результаты:

1. Проведен анализ методов интеграции информационных подсистем, используемых на практике, и поставлена задача моделирования самого процесса интеграции, решение которой позволит получить качественную интегрированную систему.
2. Разработаны оригинальные методы моделирования взаимодействия интегрированных подсистем на базе раскрашенных сетей Петри.
3. Разработан программный эмулятор и тестовая база данных для тестирования процесса взаимодействия процессинговых систем банков, объединяемых в локальную платежную сеть.

Результаты работы, проведенные эксперименты на реализованных моделях показывают, что поставленные цели разработки методов, алгоритмов и инструментальных программных средств для моделирования интеграционных процессов и приложений можно считать достигнутыми.

Основные публикации по теме диссертации

1. Вин Зо. Использование раскрашенных сетей Петри для моделирования // «14-я Всероссийская межвузовская научно-техническая конференция студентов и аспирантов, Микроэлектроника и информатика - 2007», - М.:МИЭТ, 2007.
2. Вин Зо. Анализ походов интеграции приложений //Современные технологии в задачах управления, автоматики и обработки информации: Труды XV Международного научного технического семинара. Сентябрь 2006 г., Алушта. – М.: МИФИ, 2006.
3. Вин Зо. Использование Web-сервисов в Cache` // Научная сессия МИФИ-2006. Сборник научных трудов. В 15 томах. Т.2. Программное обеспечение технологии. - М.: МИФИ, 2006.

4. Вин Зо. Походы к интеграции технологий баз данных // Современные технологии в задачах управления, автоматики и обработки информации: Труды XIV Международного научного технического семинара. Сентябрь 2005 г., Алушта. - Самара: Самарский государственный аэрокосмический университет, 2005.

5. Вин Зо, Зо Вин Аунг. Разработка прототипа информационной системы "Регистратура"// Научная сессия МИФИ-2004. Сборник научных трудов. В 15 томах. Т.2. Программное обеспечение технологии. М.: МИФИ, 2004.