

На правах рукописи

Со Тхей Вин

**ИССЛЕДОВАНИЕ КОМПОНЕНТНОЙ МОДЕЛИ ОБЪЕКТОВ ДЛЯ
СОЗДАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ОЦЕНКА
ПРОИЗВОДИТЕЛЬНОСТИ WEB-САЙТОВ**

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: кандидат технических наук, доцент
Суханов Андрей Алексеевич

Официальные оппоненты: Доктор технических наук, профессор.
Забродин Лев Дмитриевич

Кандидат технических наук, доцент
Ашарина Ирина Владимировна

Ведущая организация: Московский государственный университет
экономики статистики и информатики

Защита состоится «30» мая 2007г. В 16:00 часов на заседании диссертационного
совета Д 212.130.03 в Московском инженерно-физическом институте
(государственном университете) по адресу: 115409, г.Москва, Каширское шоссе, д.31,

С диссертацией можно ознакомиться в библиотеке института

Автореферат разослан «26» апреля 2007 г.

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

Шумилов Ю.Ю.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. В настоящее время компьютеры широко используются практически во всех сферах профессиональной деятельности. В связи с этим проблема создания эффективного и надежного программного обеспечения приобретает первостепенное значение. Технологии создания приложений на основе модели компонентно-ориентированного программирования, исследуемые в настоящей работе - ещё один шаг к созданию более совершенных, т.е. более надежных и эффективных программ. Они предоставляют принципиально новые и весьма обширные возможности, как пользователям, так и разработчикам программного обеспечения.

Особенно актуальным исследование применения технологии компонентного программирования для создания эффективных программ стало в связи с всеобъемлющим использованием сети Internet и разработкой большого количества сайтов. Известно, что web-приложения могут создавать значительную нагрузку на сервер и стать “узким ” местом, ограничивающим производительность работы пользователя в сети, а от этого в значительной мере зависят и варианты взаимодействия пользователя с сетью. Известно, что 10 сек – это предельное значение времени отклика, после наступления, которого у пользователя теряется ощущение реальной работы с данными. Приемлемая производительность Web-сайтов зависит от многих факторов и, в том числе, от кода используемого приложения и его архитектуры. Массовый пользователь регулярно в процессе своей деятельности сталкивается с необходимостью оценить и, по-возможности, спрогнозировать производительность компьютерных систем или отдельных компьютеров. Исследования времени отклика при взаимодействии пользователя с компьютером проводились с момента его появления и, тем не менее, до настоящего времени нет подходящего для массового пользователя решения этой проблемы. В связи с этим актуальной является и проблема оценки производительности и настройки Web-приложения.

Целями диссертационной работы являются:

- Исследование методов применения компонентно-ориентированного программирования к построению прикладных программ;

- разработка математических моделей для эффективной оценки производительности программно-аппаратных комплексов.

Для достижения поставленных целей необходимо решение **следующие задачи:**

- Провести анализ фундаментальных подходов и технологий (COM, COM+ и Windows DNA), разработанных фирмой Microsoft, к построению программного обеспечения в среде Windows.
- Разработать методику применения архитектуры Windows DNA к созданию распределенных корпоративных приложений масштаба предприятия.
- Провести анализ существующих методов и инструментальных средств оценки производительности веб-приложений.
- Разработать модель оценки производительности Internet-приложений.
- Реализовать программно, разработанные методы и модели для решения практических задач обеспечения учебного процесса по оценке производительности программно-аппаратных систем.

Методы исследования. При решении поставленных задач в работе использовались элементы математического аппарата теории графов, теории сетей Петри, теории планирования экспериментов, имитационного моделирования, а также компонентно-ориентированные технологии на основе компонентной объектной модели.

Научная новизна работы состоит в том, что:

- Разработан новый подход к проведению оценки производительности сетевых приложений на основе методов планирования эксперимента.
- Разработана методика проектирования масштабируемых приложений на основе модели компонентных объектов.
- Разработана имитационная модель нагрузки сетевых приложений на основе применения сетей Петри.

Практическая ценность. Практическая значимость работы состоит в том, что проведенный комплекс исследований, разработанные методы, методики и модели, предназначенные для оценки производительности web-приложений, позволяют:

- в процессе проектирования сократить время создания приложений за счет применения готовых отлаженных компонент с известной функциональностью;

- провести ускоренную оценку времени отклика приложения на посылаемые запросы при известных основных аппаратных характеристиках системы;
- осуществить прогнозирование времени отклика приложения при изменении основных характеристик системы;
- произвести определение аппаратных характеристик системы (синтез характеристик) при априорно задаваемом времени отклика.

Результаты и внедрение работы. Основные результаты работы были реализованы в виде экспериментального Web-приложения с функциональностью Интернет – магазина; имитационной модели нагрузки Web-приложения в виде сети Петри; регрессионной модели, созданной на основе методов планирования экспериментов. Результаты диссертационной работы внедрены в учебный процесс кафедры «Информационные Технологии» МИФИ (курс лекции и лабораторный практикум). Отдельные теоретические и практические результаты использовались лично автором при проведении занятий со студентами Союза Мьянма.

Апробация результатов работы. Работа в целом и отдельные её результаты докладывались и обсуждались на научных конференциях МИФИ в 2004-2007 гг., а также на научных семинарах кафедры «Информационные технологии».

Публикации. По теме диссертации опубликовано три работы, полностью отражающие основные научные результаты.

Структура и объем работы. Работа состоит из введения, четырех глав, заключения и приложений. Основной материал изложен на 100 страницах и содержит 30 рисунков. Список литературы содержит 45 наименований. В приложениях приведены основные результаты применения комбинированного метода, моделирования клиент – серверных систем.

На защиту выносятся:

- методика и алгоритмы создания пользовательских приложений масштаба предприятия
- модель исследования рабочей нагрузки приложения на основе сети Петри
- математическая модель оценки производительности Web-сайтов
- новый подход к проведению оценки производительности сетевых приложений на основе методов планирования эксперимента

СОДЕРЖАНИЕ РАБОТЫ

Во введении обосновывается актуальность темы диссертационной работы, формулируются основные положения и цель, а также задачи исследования. Определяется научная новизна, практическая значимость.

В первой главе рассматриваются основы принципы организации и проектирования распределенных приложений и, в первую очередь, трехуровневая структура построения приложений, как наиболее часто используемая при построении распределенных Internet-приложений. Основу такой структуры составляют три логических уровня:

- презентационные сервисы, или «тонкие» клиенты;
- сервисы бизнес-логики, или сервисы промежуточного слоя — middle tier;
- сервисы данных — как правило, сервер баз данных и технологии доступа к данным.

Трехуровневая структура приложения применительно к таким средствам, как динамический HTML, активная серверная страница (ASP), web-браузер и интернет-сервер, выглядит так, как показано на рис.1. В такой структуре создание и выполнение сценариев, транзакции и работу с базами данными организуется на разных логических уровнях.

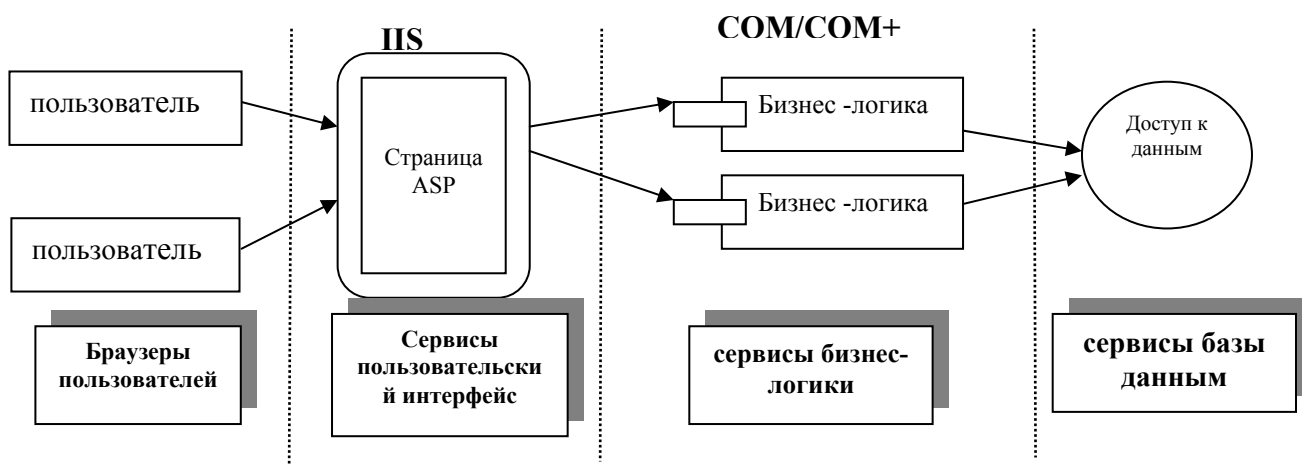


Рис.1. Архитектура 3-х уровневой web-приложения, применительно к IIS, ASP и COM/COM+.

Для создания и поддержки функционирования приложения на каждом уровне используются различные инструментальные средства, представленные в операционной системе в виде соответствующих сервисов. На уровнях представления и бизнес логики широко используются инструментальные средства (Tools) и

системные службы (System Services). Инструментальные средства включают в себя сценарии, создание компонентов, использование технологии RAD и стандартных средств разработки. Системные службы объединяют такие стандартные компоненты инфраструктуры, как службы каталогов, безопасности, управления, связи и сетевые службы. Доступ к элементам всех трех уровней и их объединение осуществляется с помощью модели COM/COM+. Общее представление структуры Windows DNA в многоуровневом приложении показано на рис. 2.

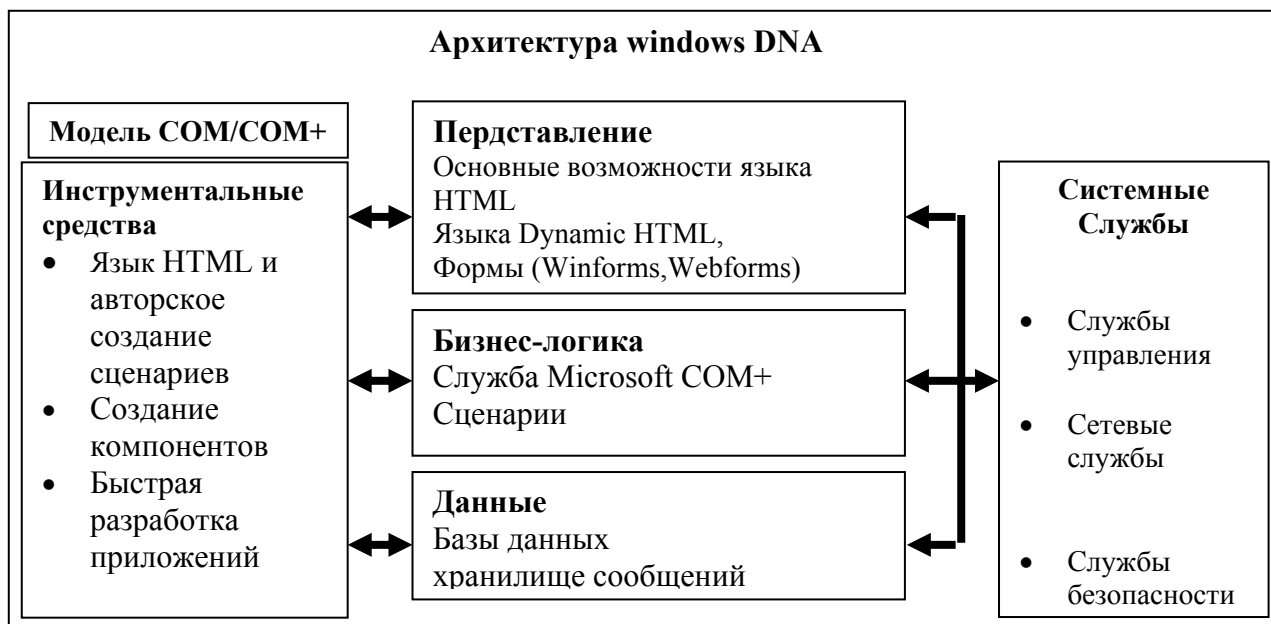


Рис. 2. Структура Windows DNA

Ценность компонентного подхода заключается в том, что он позволяет разработчикам программного обеспечения создавать приложения для онлайн-службы на основе программных компонентов, поставляемых в двоичном коде и готовых к развертыванию на любом уровне модели приложений. Эти компоненты поддерживают компоновку программных пакетов, разделение систем на подсистемы и обеспечивают распределенную функциональность. Компонентная модель позволяет разрабатывать приложения для любых структур путем включения в них программного кода любого типа или функциональности приложений, например, службы управления интерфейсом пользователя или бизнес-объекта. Компонент может иметь один или несколько интерфейсов, причем каждый из них предоставляет набор методов и свойств, которые можно получить по запросу и установить с помощью других компонентов и приложений. Например, компонент реализующий модель покупателя в системе может предоставлять пользователю различные свойства: имя, адрес и номер телефона.

Для удобства проектирования многоуровневых приложений с явным разделением обязанностей между уровнями в среде Windows DNA широко используется шаблон проектирования MVC (Model-View-Controller - «Модель-Представление-Контроллер»). Применение шаблона MVC позволяет избежать дублирования кода и построить приложение с разделением бизнес – логики и представления.

Шаблон, реализованный на презентационном уровне (рис.3), управляет двумя системными процессами, выполняемыми на уровне бизнес – логики. Один из них обеспечивает работу приложения с очередью, а другой – связь с базой данных.

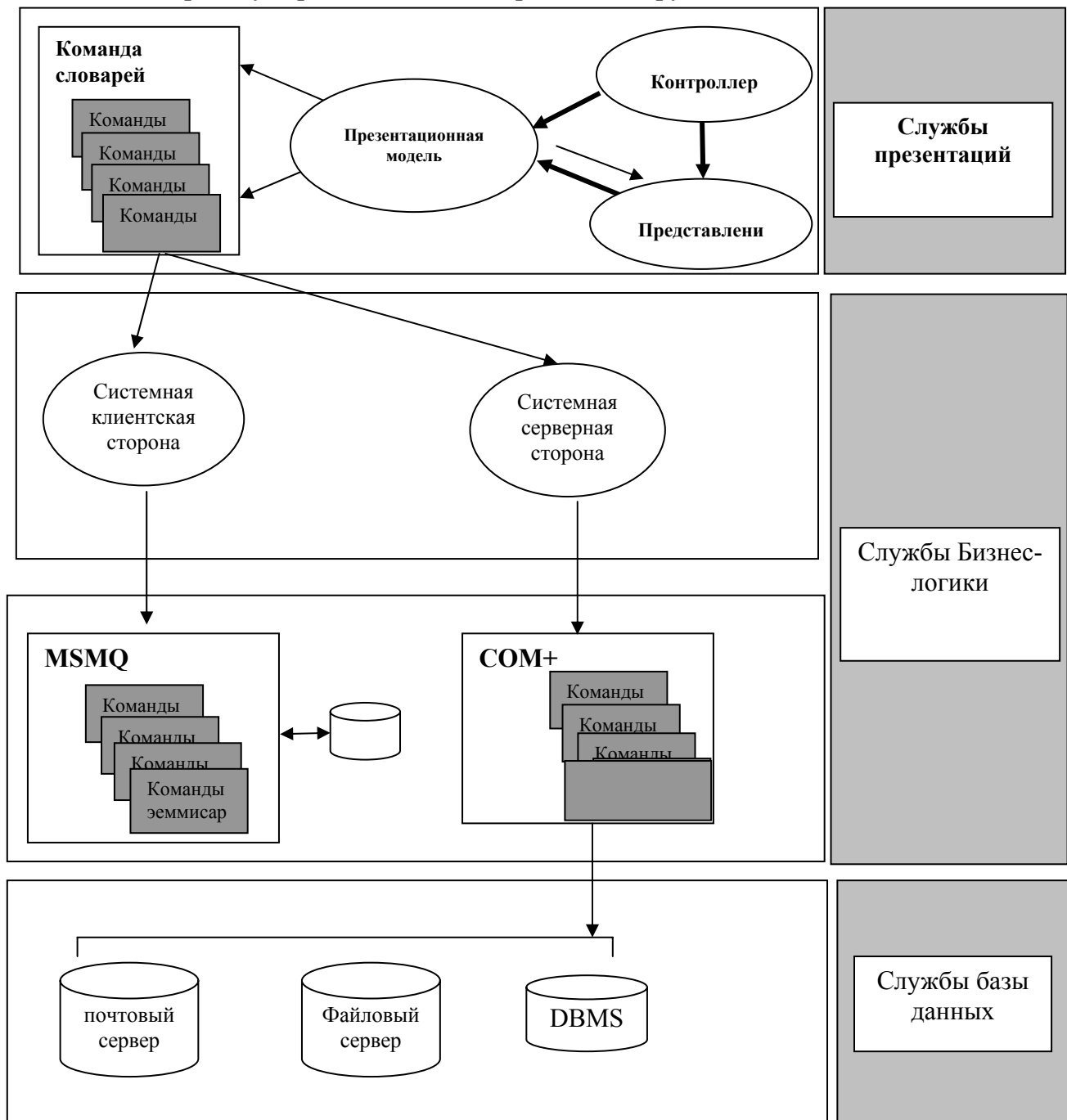


Рис. 3. Распределённый шаблон MVC в трехуровневом приложении Windows DNA

Приведенный рисунок отражает наличие следующих шагов необходимых при проектировании и реализации распределенного приложения:

- Разделение модели системы на клиентскую и серверную части.
- Создание процедур маршалинга и демаршалинга компонентных объектов на стороне клиентов и серверов.
- Проектирование команд серверной части для выполнения автоматической классификации и интерпретации данных транзакции.

Презентационная модель, представление и контролер существуют в службах презентационного уровня. При этом **презентационная модель** определяет ответственные презентационно-ориентированные команды словарей для управления визуальными операциями и команды запроса к системным моделям, реализуемым на уровне бизнес-логики. Системные модели и их команды включаются в службы уровня бизнес-логики. Команды на стороны клиентов проектирует, как представителей (эmissаров) тех команд, которые должны выполняться на стороне серверов.

Декомпозиция, поддерживаемая методологией Executants and Emissaries, разбивает уровень бизнес логики на два логических подуровня – подуровень исполнения (Executants), который предназначен непосредственно для исполнения бизнес-операций и подуровень представления (Emissaries), который предназначен в первую очередь для обеспечения эффективного взаимодействия между уровнем представления и подуровнем исполнения.

Во **второй главе** описана разработка методики применения технологии компонентного программирования для создания Internet - приложений пользователя, на примере магазина электронной коммерции.

Основными этапами этой методики являются следующие этапы:

- Постановка задачи и определение требований к приложению.
- Функциональная декомпозиция приложения по уровням.
- Выбор инструментальных средств, используемых на каждом уровне.
- Программная реализация функциональности каждого уровня.
- Отладка и тестирование приложения.

Ниже сформулированы основные требования к функциональности Internet - приложения со стороны пользователя и со стороны администратора.

Требования со стороны пользователя

Пользователь должен иметь возможность:

- Просматривать информацию о новостях и еженедельных журналах.
- Добавлять элементы в корзину пользователя.
- Удалять элементы из корзины пользователя.
- Просматривать информацию о книгах и ценах.
- Получать уведомление о готовности заказа.

Требования со стороны администратора

- Просматривать информацию о совершенных заказах.
- Просматривать сумму на личном и безналичном счете заказчика.
- Просматривать информацию о пользователях и организациях, сделавших заказ.
- Заменять статус заказа и уведомлять пользователя о готовности заказа.

Для реализации этих требований на уровне бизнес логики были созданы классы и компоненты, исполняемые на серверной стороне.

Для серверной стороны были созданы классы **cDatabase**, **cCustomerDB** и **cCustomer** и компоненты (**DButility.dll**, **customerDB.dll** и **customer.dll**), которые реализуют функции покупателя.

Класс **cDatabase**, содержит описание методов выполнения хранимых процедур или операторов SQL. Данный класс предназначен для:

- скрытия технологии и деталей реализации доступа к данным
- обеспечение независимости от источника данных

Ниже в качестве примера приведены имена и назначение методов класса **cDatabase**.

Метод ExecSP – предназначен для выполнения хранимых процедур, которые не возвращают никаких данных, а предназначены для обновления или удаления записей.

Методы ExecSQL, ExecSQLReturnRS, ExecSQLReturnVArray - позволяют формировать команды языка SQL для поиска и выборки нужных данных.

Метод ExecSPReturnRS – обеспечивает выполнение хранимых процедур и возвращает набор записей (RecordSet), в отличие от метода **ExecSP**.

Метод ExecSPReturnVArray - возвращает массив набора записей.

Для реализации методов, обеспечивающих работу с базой данных, был создан класс **cCustomerDB**, содержащий описание следующих методов:

Метод Insert – обеспечивает добавление нового покупателя (имя, пароль и адрес.и..т.д).

Метод GetData - обеспечивает извлечение данных о покупателе (имя, адрес и детали заказов).

Метод Update - обеспечивает обновление информации о покупателе (имя, адрес и детали заказов)

Метод GetIDfromLogin - обеспечивает вход в сайт (проверку имени и пароля).

Метод Search - обеспечивает поиск покупателя в базе данных по имени пользователя.

Класс **cCustomer** содержит методы обращения со стороны клиента:

Метод Add – обеспечивает добавление нового покупателя (имя, пароль, адрес.и..т.д) в базу данных, путем вызова метода **Insert** класса **cCustomerDB**.

Метод GetDetailsByID - обеспечивает извлечение данных о покупателе.(имя, адрес и детали заказов) в базу данных, путем вызова метода **GetData** класса **cCustomerDB**.

Метод Update - обеспечивает обновление информации о покупателе (имя, адрес и детали заказов) в базу данных, путем вызова метода **Update** класса **cCustomerDB..**

Метод GetIDfromLogin - обеспечивает вход в сайт (проверку имени и пароля) в базу данных, путем вызова метода **GetIDfromLogin** класса **cCustomerDB**.

Метод SearchByName - обеспечивает поиск покупателя в базе данных по имени пользователя в базу данных, путем вызова метода **Search** класса **cCustomerDB**.

Остальные компоненты, реализующие обращение к таблицам базы данных с именем «заказы» и «продукты» **orderDB.dll**, **order.dll**, **productDB.dll** и **product.dll** содержат классы **cOrderBasket**, **cOrderstatus**, **cBasketContents** для заказов и **cProduct**, **cProductDB** для продуктов. В этих классах описаны методы, которые обеспечивают

пользователям сайтов поиск информации, добавление новых заказов корзину и получение информации о заказах и о продуктах.

На следующем рис. 4 показана взаимосвязь вышеуказанных модулей.

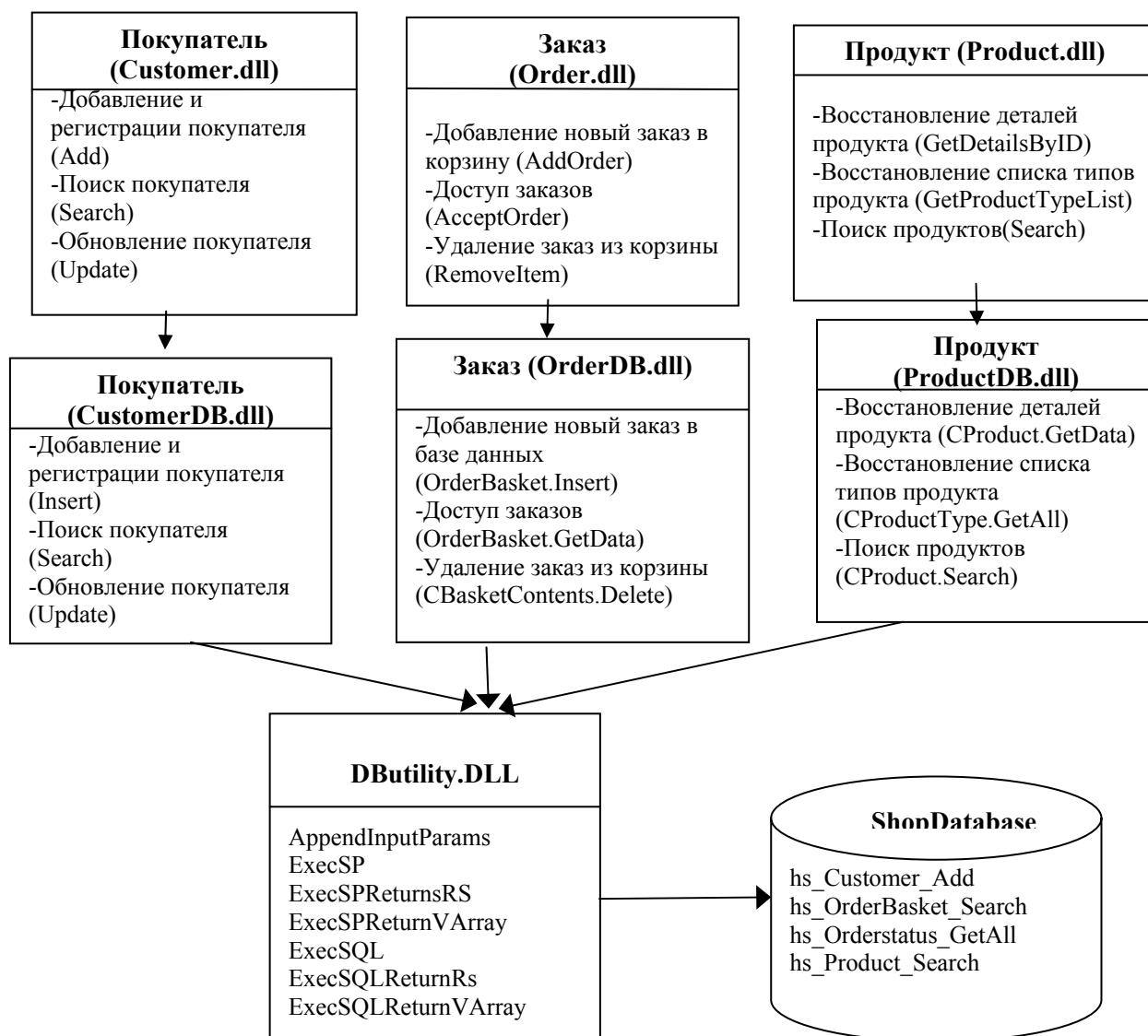


Рис. 4. Компоненты бизнес логики и сервисы компонентов (*DBUtils.DLL*) для коммуникации между уровнем бизнес логики и базами данных.

На уровне представления данных был создан пользовательский интерфейс (web-страница) с использованием технологии ASP (активные серверные страницы) с компонентами навигации, позволяющими пользователям искать информацию о продуктах, добавлять книги в корзину, оформлять заказы и т.д.

На уровне данных в среде Microsoft SQL Server была реализована база данных, логическая схема которой представлена на следующем рис. 5.

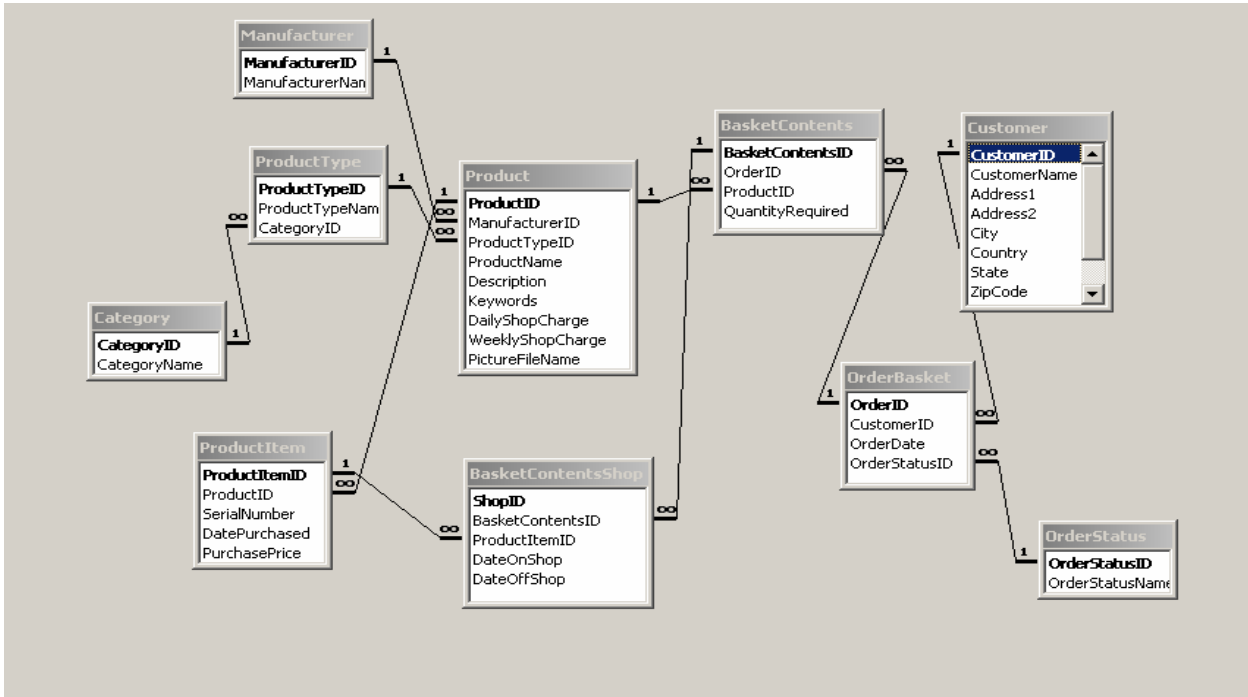


Рис. 5 Схема базы данных магазина электронной коммерции

Третья глава посвящена анализу подходов и разработке методов оценки производительности клиент – серверных систем. Клиент - серверная система – это сочетание аппаратной платформы, операционной системы, сетевого и программного обеспечения. Все эти компоненты оказывают влияние на производительность Web-серверов и интрасетей. Однако, с точки зрения пользователя Web-сайта, наиболее важными характеристиками производительности Web-серверов в работе было выбрано **время отклика**, как наиболее важная характеристика для массового пользователя.

Временные задержки, формирующие время отклика сервера (рис.6) на поступающие от клиента запросы складывается из двух определяющих факторов: времени, потраченного различными сообщениями на перемещение от браузера пользователя до Web-сайта - **время сети** – ($t_{\text{сети}}$) и **времени Web-сайта** ($t_{\text{сайта}}$). **Время Web-сайта** в свою очередь складывается из **времени обслуживания** ($t_{\text{обсл.}}$) и **времени нахождения запроса в очереди** ($t_{\text{очер.}}$). Так как запрос в течение своего выполнения может воспользоваться ресурсами процессора несколько раз, суммарное время обслуживания j –го запроса на i - ом ресурсе характеризует **время потребности в обслуживании** и выражается в виде,

$$t_{обсл_i} = \sum_j t_i^j \text{ где } t_i^j - \text{ среднее время обслуживания, приходящееся на одно}$$

посещение ресурса.

Время нахождения в очереди ($t_{очер.}$) зависит от длины очереди и определяется, как

$$t_{очер_i} = \sum_j t_{ож_i}^j, \text{ где } t_{ож_i}^j - \text{ время, затрачиваемое } j\text{-ым запросом на ожидание}$$

доступа к i -ому ресурсу. Сумма **времени обслуживания и времени нахождения в очереди** составляет время пребывания запроса на ресурсе. В этом случае **время**

отклика может быть выражено как $t_{откл.} = \sum_i (t_{обсл_i} + t_{очер_i})$

Так как измерение каждого из этих времен достаточно трудоемко, то определить **время отклика** также очень сложно.



Рис 6. Составные части времени отклика

Для оценки производительности в работе предложен комбинированный метод, суть которого состоит в использовании имитационного моделирования, методов планирования эксперимента и средств нагрузочного тестирования.

В качестве имитационной модели клиент - серверной системы была выбрана раскрашенная сеть Петри.

Обычная сеть Петри, формально задаваемая как $N=(P,T,I,O)$,

где $P=(p1,p2,p3,p4)$ - конечное множество позиций;

$T=(t1,t2,t3,t4)$ - конечное множество переходов;

$I : P \times T \rightarrow \{0,1\}$ - прямая функция инцидентности,

$O : T \times P \rightarrow \{0,1\}$ - обратная функция инцидентности,

используется для отражения статике моделируемой системы.

Для представления динамических свойств вводится функция разметки (маркирования сети) $\mu : P \rightarrow \{0,1,2,\dots\}$. С помощью функции μ позиции сети помечаются целыми неотрицательными числами. При графическом задании сети Петри разметка отображается помещением внутри вершин-позиций соответствующего числа точек, называемых метками. Размеченная сеть Петри формально описывается следующим набором $N_\mu = (P, T, I, O, \mu_0)$, μ_0 - начальная разметка сети.

Для облегчения визуального восприятия сети Петри, а также для ее смысловой структуризации была построена размеченная иерархическая временная сеть Петри в среде CPN Tools с моделированием временных нагрузок выполнения обращений пользователя (заявок) к web – приложению, расположенному сервере. Модель реализует три уровня обработки заявки в системе: **уровень клиента**, **уровень сети** и **уровень сервера**. В этой модели позиции отражают роль человека, создающего заявки, а переходы - действия, производимые сервером при обработке заявки. В данной модели используется понятие фишки для отражения перемещения заявки от клиента до сервера.

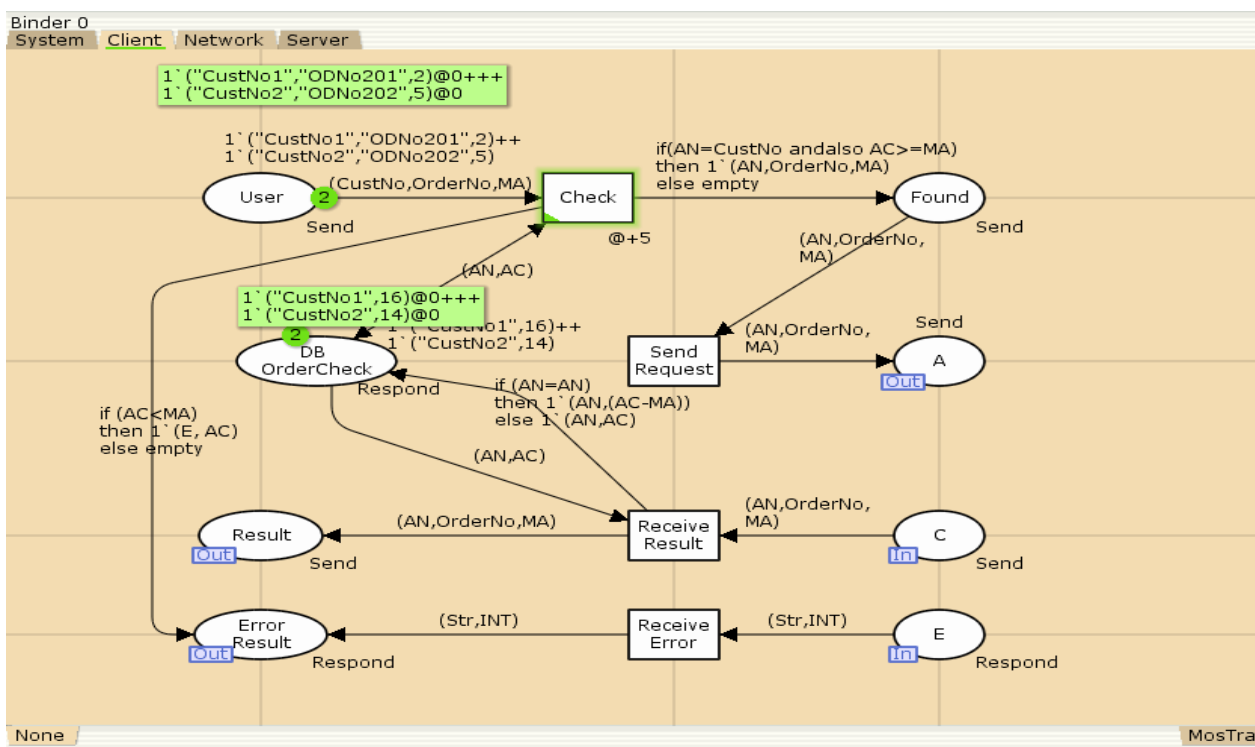


Рис 7. Уровень клиентов

Первый уровень иерархической сети моделирует взаимодействие пользователя и сети (рис 7). Второй уровень иерархической сети моделирует передачу запросов по сети.

Третий уровень иерархической сети моделирует взаимодействие сети и сервера. На третьем уровне сервер, получив от пользователя через сеть запрос проверяет базу данных (CheckOrderDB) (рис 8) и посылает ответ клиенту снова через сеть.

Инструментальное средство CPN Tools позволяет задавать временные интервалы задержек на переходах и на стрелках, выходящих из переходов. Перемещение запроса по сети моделируется перемещением фишек.

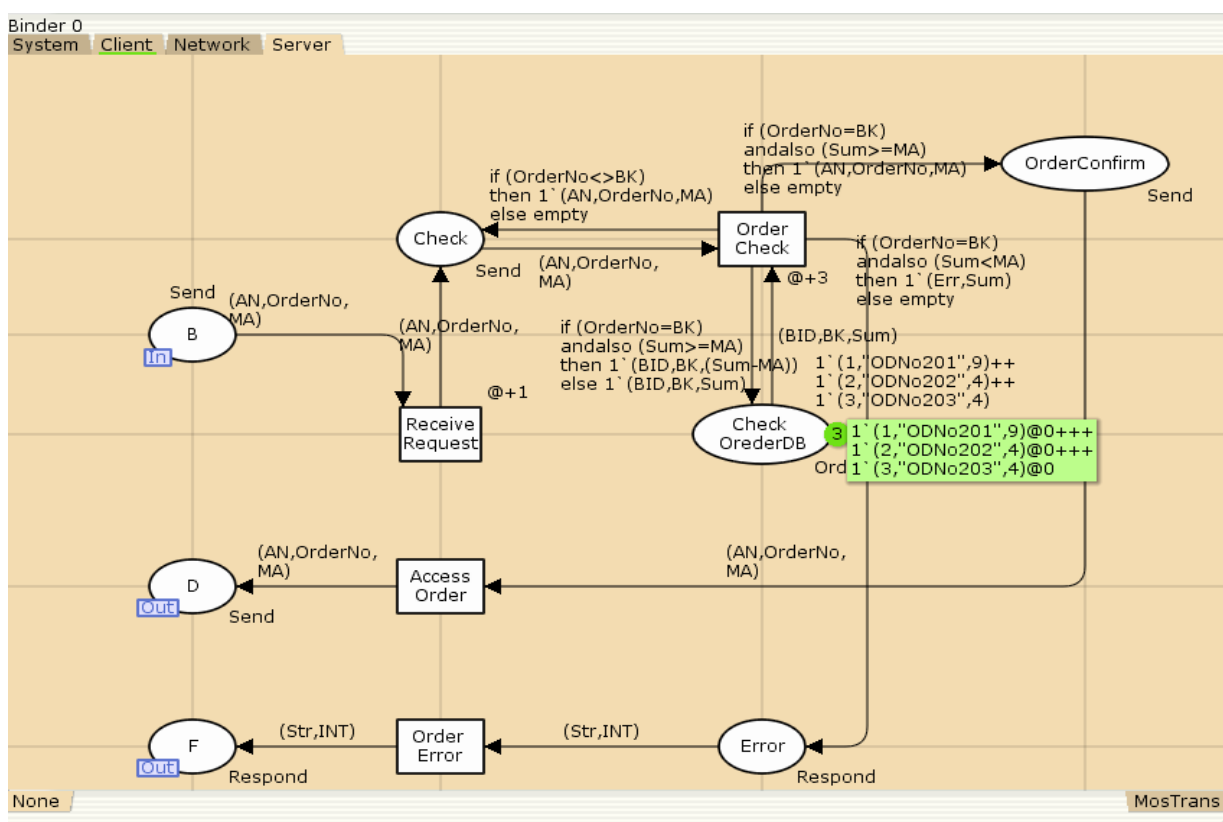


Рис 8. Уровень серверов

В результате работы модели иерархической сети можно получить:

- общее качество заявок в системе, количество заявок за определенный период и длительность данного периода.
- суммарное время задержки за определенный период.

Результаты работы модели (суммарное время задержки) были использованы для построения модели оценки производительности Web-сайта с использованием методов планирования эксперимента (ПФЭ и ДФЭ).

Для представления времени отклика Web-сервера на запросы пользователя была выбрана линейная полиномиальная вида:

$$M(Y) = Y(x_1, x_2, \dots, x_n) = \beta_0 + \sum \beta_i x_i + \sum \beta_{ij} x_i x_j + \dots$$

Где $x_1 \dots x_n$ - входные факторы в условных единицах, а $M(Y) = Y(x_1, x_2, \dots, x_n)$ - математическое ожидание функции отклика.

Выборочные коэффициенты регрессии b_0, b_i, b_{ij} , являющиеся оценкой теоретических коэффициентов $\beta_0, \beta_i, \beta_{ij}$ вычисляются в соответствии с выражением приведенным ниже.

$$b_i = \frac{\sum_{u=1}^N x_{iu} \bar{y}_u}{N}$$

Оценка функции отклика $\hat{Y}(b_0, b_i, b_{ij})$ вычисляется по результатам проведенных экспериментов.

В четвертой главе рассматриваются средства проведения эксперимента, строятся полиномиальные модели для оценки времени отклика клиент – серверной системы и интерпретируются полученные результаты. В качестве средства получения реальных экспериментальных данных

Реальные экспериментальные данные для их использования в сети Петри выбирались на основании проведения нагрузочного тестирования, которое осуществлялось с применением средств нагрузочного тестирования Microsoft Application Center Test (ACT). Это средство позволяет смоделировать большую группу пользователей путем открытия нескольких подключений к серверу и быстрой отправки HTTP-запросов. Для создания нагрузки на Web – сайт был разработан ряд алгоритмов и тестовых сценариев. Один из примеров, разработанных алгоритмов показан на рис 9.

Результаты тестирования использовались для выбора уровней варьирования физических переменных и определения оценки функции отклика при построении регрессионных моделей.

В качестве выходных параметров для проведения тестирования были выбраны следующие параметры:

- количество виртуальных пользователей

- задержка между запросами
- время тестирования

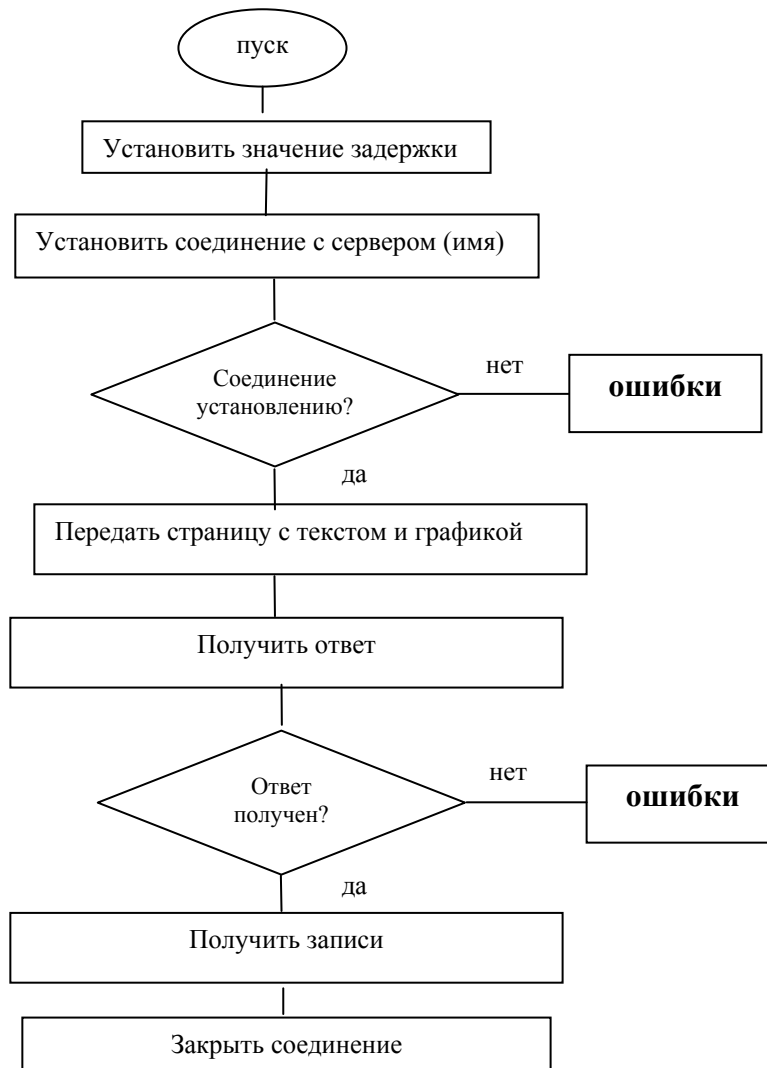


Рис 9. Алгоритм прохождения запроса в клиент - серверном приложении электронного Internet - приложения

Заключение

В **заключении** приведены основные результаты диссертационной работы.

Основные результаты работы

1. Разработана методика проектирования масштабируемых приложений на основе модели компонентных объектов
2. Построена модель клиент – серверной системы на основе размеченной сети Петри и получены результаты моделирования для построения модели Web-сервера.
3. Предложен комбинированный метод, моделирования клиент – серверных систем на основе использования имитационного моделирования, методов планирования эксперимента и средств нагрузочного тестирования.

4. Построены модели Web-сервера на основе применения методов планирования эксперимента
5. Разработаны алгоритмы и тестовые сценарии для применения средств нагрузочного тестирования, использованные для моделирования нагрузки web-сервера.

Основные публикации по теме диссертации

Список опубликованных работ по теме диссертации:

- А.А.СУХАНОВ, СО ТХЕЙ ВИН, «Исследование возможностей среды Visual Basic для создания приложения на основе модели компонентных объектов». Научная сессия МИФИ 2004»
- А.А.СУХАНОВ, СО ТХЕЙ ВИН, «Web – портал кафедры и анализ производительности Web-приложений». Научная сессия МИФИ 2005»
- СО ТХЕЙ ВИН, «Сравнение производительности приложений, создаваемых на основе объектно-ориентированного подхода и технологии компонентных объектов». Научная сессия МИЭТ 2007»