

На правах рукописи

**Пхоун Найнг**

**МОДЕЛИРОВАНИЕ ПРОЦЕССОВ ДИНАМИЧЕСКОГО  
СВЯЗЫВАНИЯ WEB-СЕРВИСОВ**

05.13.11 – математическое 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  
Гусева Анна Ивановна  
кандидат технических наук, с.н.с.  
Павлов Юрий Борисович

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

Защита диссертации состоится 30 мая 2007 г. в \_\_14:00\_\_ часов  
\_\_ минут на заседании диссертационного совета Д 212.130.03 при  
Московском инженерно-физическом институте (государственном  
университете) по адресу:  
115409, Москва, Каширское шоссе, 31, тел: 323-91-67, в  
конференц-зале главного корпуса.

С диссертацией можно ознакомиться в библиотеке МИФИ

Автореферат разослан «\_\_»- апреля- 2007 г.

Отзывы 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 работы**

В течение нескольких последних лет было приложено много усилий, чтобы развить инфраструктуру, поддержать развертывание, открытие, и использование веб-сервисов. Главные поставщики компьютерной и программной инфраструктуры, включая IBM, Microsoft, Sun Microsystems, настойчиво работают, расширяя свои технологические среды, чтобы поддержать развитие, развертывание, и обслуживание веб-сервисов. Их решения направлены на улучшение безопасности, поддержание транзакций, и улучшение координации веб-сервисов.

До настоящего времени, научные исследования в области веб-сервисов были, прежде всего, сосредоточены на определении и развитии инфраструктуры публикации, поиска и поставки веб-сервисов, которые удовлетворяют функциональным требованиям потребителя. Существующая инфраструктура не обеспечивает поддержку нефункциональных требований, типа производительности, масштабируемости, надежности, стоимости и т.д. и, как следствие этого, - использование в приложениях статической композиции веб-сервисов, которая не предусматривает переадресацию сервисов в процессе выполнения.

В настоящее время в связи с ориентацией на сервис ориентированную архитектуру (SOA) в информационных системах компаний и предприятий, а, в общем плане, в ориентации на концепцию SAAS (Software As A Service) научная общественность стала активно заниматься проблемой динамической композиции веб-сервисов, которая предусматривает переадресацию в процессе выполнения.

### **Цель работы**

Целью диссертации является исследование и разработка методов моделирования динамической композиции веб-сервисов и инструментальных программных средств для реализации динамической композиции веб-сервисов.

Для достижения поставленной цели в диссертации решены следующие задачи:

1. Проанализированы современные методы использования Web-сервисов в информационных системах.
2. Проанализировано состояние стандартизации Web-сервисов.
3. Проанализированы основные подходы к статической композиции Web-сервисов и предложения по динамической композиции Web-сервисов.
4. Разработана модель динамической композиции Web-сервисов.
5. Разработано экспериментальное приложение, в котором продемонстрированы методы статической и динамической композиции Web-сервисов.

### **Научная новизна**

1. Подход к динамической композиции веб-сервисов, предлагаемый в диссертации, предполагает включение в процесс взаимодействия клиента и сервера информационного брокера.
2. Информационный брокер поддерживает реестр веб-сервисов и осуществляет непрерывный мониторинг операционной обстановки.
3. Динамическая композиция веб-сервисов наиболее эффективна, если обеспечена информация о среде функционирования веб-сервисов на базе мониторинга операционной обстановки.
4. Методы и модели статической и динамической композиции веб-сервисов на базе сетей Петри. Использование сетей Петри позволяет построить целый спектр моделей композиции с разным уровнем детализации процессов взаимодействия.

### **Практическая ценность**

1. Разработано экспериментальное веб-приложение на базе платформ Java и PHP, демонстрирующее варианты статической и динамической композиции веб-сервисов.
2. На основе использования сетей Петри построены модели динамической композиции веб-сервисов.
3. Разработанное экспериментальное веб-приложение и модели динамической композиции веб-сервисов на базе сетей Петри использовались в учебном процессе для студентов Союза Мьянма, проводимом на кафедре «Кибернетика» МИФИ. Автор лично проводил занятия со студентами.

### **Основные научные результаты, представляемые к защите**

1. Подход к динамической композиции веб-сервисов, предполагающий включение в процесс взаимодействия клиента и сервера информационного брокера.
2. Методы и модели статической и динамической композиции веб-сервисов на базе сетей Петри.

### **Апробация работы**

Основные результаты диссертации докладывались и обсуждались на следующих научных конференциях и семинарах.

1. Научная сессия МИФИ, Программное обеспечение технологии, - 2004, 2006, 2007.
2. Международный научно технический семинар, Алушта, 2005, 2006.

### **Публикации**

Основные результаты диссертационной работы опубликованы в 5 печатных работах.

### **Структура и объем работы**

Диссертация состоит из введения, четырех глав, заключения, списка использованных источников, содержит 31 иллюстрации.

## СОДЕРЖАНИЕ РАБОТЫ

**Во введении** обоснована актуальность темы диссертации, её научная новизна и практическая значимость, сформулирована цель работы.

**В главе 1** рассматриваются основные положения технологии Web-сервисов, современные тенденции развития методов интеграции информационных систем и место Web-сервисов в этих методах. Поднимается проблема создания единого информационного пространства как основа широкого использования Web-сервисов в распределенных системах. В конце первого раздела диссертации поставлена цель и конкретные задачи диссертационного исследования.

Начиная с 1992г, когда в CERN была опубликована идея гипертекста, Web-технологии шагнули далеко вперед. Это связано не только с признанием глобальной полезности новой технологии, но и со стремительным развитием всей компьютерной и сетевой инфраструктуры, которые, по большому счету, и обеспечили развитие Web-технологий. Новая технология оказала серьезное консолидирующее воздействие на всю компьютерную отрасль: гранды компьютерной индустрии поняли, что продвижение вперед возможно только в том случае, если будут выработаны стандарты, следование которым будет выгодно всем – и отрасли в целом и каждой фирме в отдельности.

Коммерческие Web приложения, освоив простую обработку транзакций, пошли дальше - следующим логическим шагом в их развитии стала интеграция бизнес-процессов различных предприятий. Web приложения стали рассматриваться как сервис: на арене появляется сервис-ориентированный Web. В основу этого развития легла технология XML (Extensible Markup Language). Появляется протокол SOAP (Simple Object Access Protocol). Согласно сервис-ориентированному сценарию, Web превращается в набор серверов приложений, обменивающихся информацией в формате XML по протоколу SOAP.

Основой сервис-ориентированного Web является Web-сервис — набор логически связанных функций, которые могут быть программно вызваны через Интернет. Информация о том, какие функции предоставляет данный Web-сервис, содержится в документе WSDL (Web Service Description Language), а для поиска существующих Web-сервисов предполагается использование специальных реестров, строящихся на основе спецификации UDDI (Universal Description, Discovery and Integration).

Концепция Web-сервисов – это концепция новой технологии для разработки и эксплуатации распределенных информационных систем. Существовавшие до появления этой концепции технологии, такие как системы типа DCOM (Microsoft) и CORBA (OMG), не смогли в полной мере обеспечить совместимость различных приложений в неоднородных распределенных сетях.

Ведущие компании-производители программного обеспечения под эгидой WWW-консорциума (W3C) прикладывают значительные усилия для развития технологии Web-сервисов. Ее основу составляют:

1. коммуникационные интернет-протоколы HTTP и/или SMTP;
2. протокол SOAP (Simple Object Access Protocol) для доставки сообщений в XML-формате;
3. язык WSDL (Web Services Definition Language) для описания интерфейса Web-сервисов.

HTTP, SOAP, XML, WSDL – это не единственные аббревиатуры, которые связаны с технологией Web-сервисов. Пожалуй, только технология XML не подвергается сомнению и признается всеми. Однако именно эти аббревиатуры поддерживаются рекомендациями W3C; их спецификации признаются, поддерживаются и развиваются ведущими компаниями-производителями программного обеспечения отрасли.

При рассмотрении архитектуры Web-сервисов различают базовую (Basic Architecture) и расширенную (Extended Web Services Architecture) архитектуру. Базовая архитектура включает в себя обязательный набор средств этой технологии, уже нашедший

применение в различных реализациях Web-сервисов. Расширенная архитектура в большей части ориентирована на дополнительные или перспективные спецификации, расширяющие возможности технологии, такие как:

1. асинхронные сообщения,
2. передача данных в режиме вложений (attachment) SOAP-сообщений,
3. идентификация и конфиденциальность сообщений и др.

Ограничимся рассмотрением базовой архитектуры, которая обеспечивает выполнение следующих технологических операций:

- 1 обмен сообщениями,
- 2 обеспечение удаленного вызова процедур,
- 3 описание услуг Web-сервисов,
- 4 регистрация (публикация) и поиск (обнаружение) описаний услуг.

В базовой архитектуре взаимодействие Web-сервисов определяется как обмен сообщениями между программами, запрашивающими сервис и программами поставщиками сервисов. Программы-клиенты, которые запрашивают выполнение сервиса, и программы-серверы, которые обеспечивают выполнение сервиса, могут менять свою роль в базовой архитектуре взаимодействия. Каждая из программ может одновременно быть как клиентом, запрашивающим некоторую услугу, так и сервером, обеспечивающим услугу. Организации-провайдеры обеспечивают реализацию сервиса и публикацию его описания; организации, использующие сервисы в своих информационных системах, должны знать, как и где можно найти сервис с нужными функциональными характеристиками.

Таким образом, базовая архитектура Web-сервисов предполагает наличие, по крайней мере, трех организаций: провайдера сервиса, регистра сервисов, обеспечивающего регистрацию и средства поиска сервисов и клиента, запрашивающего сервис. Их взаимодействие включает операции декларации, поиска и связывания сервиса. В типичном сценарии

использования Web-сервиса провайдер, имеющий права на сервис, поддерживает программные модули, реализующие сервис, формирует описание сервиса и публикует его в специальном регистре, делая это описание доступным всем заинтересованным потребителям.

Чтобы использовать некоторый Web-сервис для выполнения своих задач клиент-потребитель должен, прежде всего, получить описание сервиса. Это он делает либо обращаясь непосредственно к провайдеру, либо в регистр сервисов. Далее, используя это описание, клиент встраивает обращение к Web-сервису в собственный программный комплекс, который выполняет процедуру связывания с Web-сервисом и инициирует взаимодействие с модулями, реализующими данный сервис.

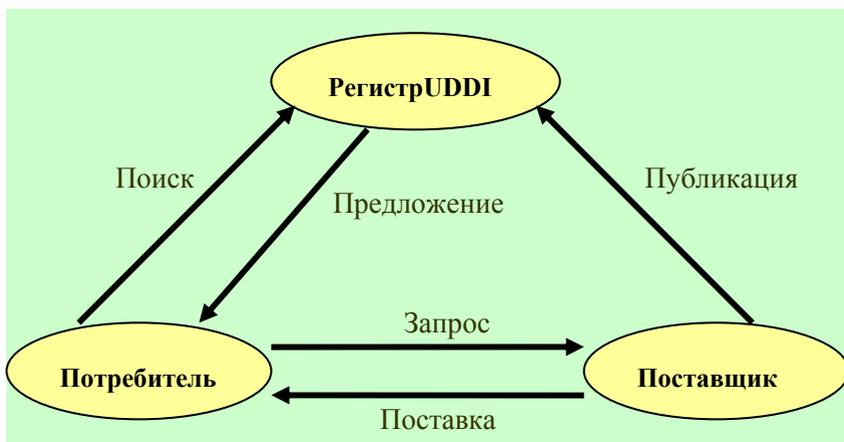


Рис.1. Компоненты и операции Web-сервисов

В компьютерной отрасли парадигма сервисов прошла целый ряд эволюций. Каждый шаг эволюции определялся появлением новых технологий и созданием соответствующей инфраструктуры. Суть любого шага эволюции состоит в улучшении механизма публикации сервисов, поиска сервисов, выполнения запросов и доставки ответов. Web сервисы - последний шаг эволюции в парадигме сервисов. Интернет и Web технологии - катализаторы этого развития. Web сервисы предлагают более легкий и

эффективный способ публикации, поиска, выполнения запроса, и поставки сервиса.

Идея SOA, подогреваемая активным маркетингом грандов IT индустрии, по крайней мере, в принципе, способствует созданию рынка сервисов. Рынок сервисов, особенно Web сервисов, предполагает наличие нескольких сервисов, которые обеспечивают идентичные функциональные возможности и интерфейс.

Такие Web сервисы могут быть доступны как от одного, так и от нескольких поставщиков (сервис-провайдеров), при этом, обеспечивая идентичные функциональные возможности, они могут отличаться по нефункциональным характеристикам. В идеале, потребитель Web сервисов должен использовать тот Web сервис, который лучше всего подходит под его функциональные и нефункциональные требования, например, его использование более экономично и надежно.

Вопрос заключается в том, как идентичные функциональные возможности и интерфейс могут обеспечить независимые производители. Решение проблемы видится в создании и поддержании «единого информационного пространства» той или иной отрасли экономики.

Для реализации данного подхода к представлению информационного пространства можно использовать описание объектов информационного пространства на базе языка XML. В терминологии языка XML под информационным объектом обычно понимается XML документ. Стандарт XML-Schema, предназначенный для декларации информационной структуры XML документов, как раз и позволяет решить проблему моделирования (или формализации) информационного пространства на современном уровне. Разработанные и согласованные XML-схемы, непрерывно поддерживаемые в актуальном состоянии и доступные в режиме онлайн всем взаимодействующим субъектам, образуют ту стандартную среду, которая обладает достаточной гибкостью, чтобы создать условия для автоматизированного взаимодействия компонентов информационных систем.

Следует заметить, однако, что XML позволяет только структурировать информацию, семантика остается вне поля зрения XML-схемы. Для описания информационного пространства желательно было бы включить в него элементы семантики. С этой целью консорциум W3C разработал технологию описания ресурсов Resource Description Framework (RDF). W3C декларирует RDF как способ представления знаний в децентрализованном мире, как основную технологию Semantic Web. То, в чём RDF сильнее всего отличается от XML и других технологий — это то, что RDF предназначен для представления знаний в распределённом мире. Для RDF особенно важен смысл. Всё, с чем работает RDF, имеет определённый смысл, — ссылается на какой-то конкретный объект, или на абстрактное понятие, или на некий факт. Используя RDF описание, можно провести логический вывод, связывающий эти факты, и указать, как можно найти сами факты в огромной базе данных всех знаний, представленных в RDF.

В качестве примера создания «единого информационного пространства» в главе приводится описание Common Information Model (CIM) –информационной модели всех основных объектов предприятий электроэнергетики и Generic Interface Definition (GID) - спецификации интерфейсов.

Сущности электроэнергетики представляются как классы объектов с атрибутами и связями. В настоящий момент принят стандарт IEC 61970 Part 301 Common Information Model (CIM) Base, где описаны основные классы объектов электроэнергетики.

Если Common Information Model - это общий язык, на котором могут общаться системы, то стандарт Generic Interface Definition (GID) содержит спецификацию интерфейсов, то есть определяют общий механизм, с помощью которого приложения могут взаимодействовать и предоставлять свои данные. GID-интерфейс может связывать клиентскую и серверную части приложения, два приложения, приложение и хранилище данных, приложение и систему гарантированной доставки сообщений и так далее.

Разработанные стандарты упрощают интеграцию

унаследованных приложений, для которых могут быть созданы совместимые со стандартами GID и CIM оболочки-интерфейсы. Таким образом, создается единая и легкая в использовании интеграционная среда.

В качестве примера можно привести стандарт передачи медицинских документов HL7, разработанный и принятый в качестве национального в США и ставший де-факто международным. В настоящее время завершается разработка третьей версии стандарта, в которой использованы наиболее современные подходы к моделированию предметной области здравоохранения.

С точки зрения создания Web сервисов это означает, что сначала должны быть декларированы, стандартизованы и общеприняты WSDL, на которые должны ориентироваться разработчики. К сожалению, в настоящее время достаточно часто все происходит наоборот: WSDL генерируется на базе разработанного программного кода.

В заключение первой главы ставится задача диссертации.

Наряду с четко определенной функциональностью Web-сервис характеризуется набором нефункциональных характеристик. Именно эти характеристики ответственны за то, что приложение, отправившее запрос к Web-сервису, находится в значительной степени в условиях неопределенности. Момент, когда придет ответ, сильно зависит от многих причин:

- Сервер, на котором находится Web-сервис, может работать в условиях перегрузки, когда число поступающих запросов достаточно велико.
- Сервер, на котором находится Web-сервис, подвергся атаке.
- Линия связи приложения и Web-сервиса перегружена, что приводит к потере пакетов, повторным запросам и, в результате, существенному увеличению времени ожидания ответа.

В настоящее время парадигма работы с Web-сервисами

статична: приложение обращается к Web-сервису по конкретному адресу (URL), который был выбран при разработке приложения и вынуждено пассивно ждать ответа. Однако Web-сервисов с идентичной функциональностью может быть достаточно много, так как их разработка, поддержание и размещение, в конечном счете, определяется получением прибыли. Владелец Web-сервисов может разместить один и тот же сервис на нескольких серверах и, наоборот, идентичные по функциональности сервисы, принадлежащие разным владельцам, могут быть размещены на одном сервере.

Таким образом, если характеристики обслуживания конкретным Web-сервисом, расположенным на конкретном сервере, не удовлетворяют клиента, то он может обратиться к другому сервису на выбранном сервере или к тому же сервису, но расположенному на другом сервере. Конечно, подобную переадресацию можно сделать в статике, выполнив достаточно простую модификацию приложения, но парадигма работы с Web-сервисами приобретет принципиально новое качество, если переадресация будет выполняться в динамике, то есть в процессе работы приложения.

Однако динамическое связывание будет эффективным, если оно производится на основе мониторинга операционной обстановки, в которой функционируют Web-сервисы. Это предполагает создание новой службы и включение ее в процесс взаимодействия клиента и сервера. В диссертации ставится задача исследования различных моделей динамического связывания Web-сервисов.

**В главе 2** развивается подход к моделированию процессов динамического связывания Web-сервисов на базе многоцветных сетей Петри. Рассматривается вопрос создания специальной службы, осуществляющей мониторинг функциональной активности Web-сервисов, как основы динамической композиции Web-сервисов.

Моделирование широко распространено во всех инженерных

дисциплинах и в проектировании программных комплексов используется много различных моделей. Например, объектно-ориентированное проектирование связывают с целым комплексом моделей, оформляемых на языке UML, функциональное проектирование связывают с построением моделей в стиле IDEF0. Каждая модель позволяет взглянуть на проектируемый программный комплекс с определенной точки зрения, отбросив все детали, с этой точки зрения не существенные.

В самом общем виде модель задается множеством объектов и отношений между ними. Однако в таком абстрактном виде она полезна в весьма специфических случаях. На практике весьма важен аспект представления модели: именно удобное графическое представление помогает в решении очень многих вопросов, начиная от общения с заказчиком и формулировкой требований, заканчивая внесением изменений в функционирующий комплекс.

В реальной системе, в частности, в программном комплексе, отношения объектов, свойства объектов изменяются в процессе функционирования. Для решения очень многих вопросов важно иметь возможность наблюдать эту смену состояний не в реальной системе, а в ее модели. Поэтому очень важно иметь «движок», который бы «приводил в движение» модель системы.

С этих точек зрения идеальным инструментом для построения моделей программных комплексов являются сети Петри. Используя сети Петри можно промоделировать как чистую логику программы (одноцветные сети), абсолютно отвлекаясь от информационно-семантической составляющей, так и целый спектр приближений к реальной программе (многоцветные сети).

В данном разделе приведены построенные модели в виде многоцветных сетей Петри для различных подходов к статической и динамической композиции Web-сервисов, разработанных в диссертационном исследовании.

Рассмотрим первый вариант более подробно. В компьютерной практике достаточно часто реально работающие программы скрываются за некоторым фасадом. Например,

некоторый абстрактный «SQL-сервер» скрывает реально работающую реляционную СУБД. Точно также информационный брокер может организовать исследование рынка Web-сервисов, объявить «собственные» Web-сервисы с WSDL, которые удовлетворяют отраслевым стандартам (например, HL7), но, в действительности, пользующиеся сервисами сторонних производителей.

Современная практика производства Web-сервисов, при которой WSDL генерируется на основе исходного программного кода, может привести к разным WSDL для практически функционально идентичных сервисов. В этом случае брокер выполняет работу по приведению этих сервисов к некоторому стандартному интерфейсу, и если одновременно он возьмет на себя функции мониторинга операционной обстановки, то для обращающегося к брокеру клиента действительно будет иметь место динамическая композиция Web-сервисов. На рис. 2. представлена сеть Петри, которая моделирует этот вариант динамической композиции Web-сервисов.

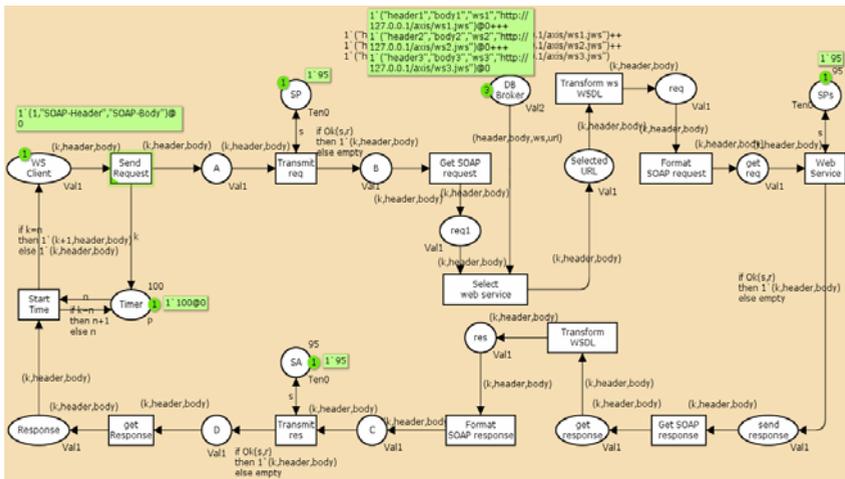


Рис. 2. Первый вариант динамической композиции Web-сервисов

Клиент посылает запрос брокеру и просто ждет ответа. Вся дальнейшая работа от него скрыта. Слабая сторона такой

организации состоит в том, что все взаимодействия осуществляются через брокер.

Во втором варианте информационный брокер организует исследование рынка Web-сервисов и осуществляет их семантическую классификацию. Каждому семантическому коду соответствует единственный WSDL и несколько разных реализаций Web-сервиса.

Клиент на этапе разработки настраивается на декларированный брокером WSDL, но за конкретным URL обращается непосредственно перед выполнением запроса к Web-сервису, посылая брокеру запрос с указанием семантического кода. Брокер возвращает URL Web-сервиса, наиболее подходящий к текущей ситуации.

Во втором варианте клиент берет на себя массу дополнительной работы. Он должен сделать запрос брокеру, чтобы получить URL, запустить таймер вместе с запросом к Web-сервису, посылать брокеру запрос в случае неудачи и т.д. Модель этого варианта взаимодействия представлена на рис. 3.

Как видно из представленных достаточно общих моделей, реализация динамической композиции Web-сервисов требует существенной перестройки всей статической архитектуры использования сервисов. Если она в реальности и будет применяться, то, видимо, в сочетании статического и динамического варианта.

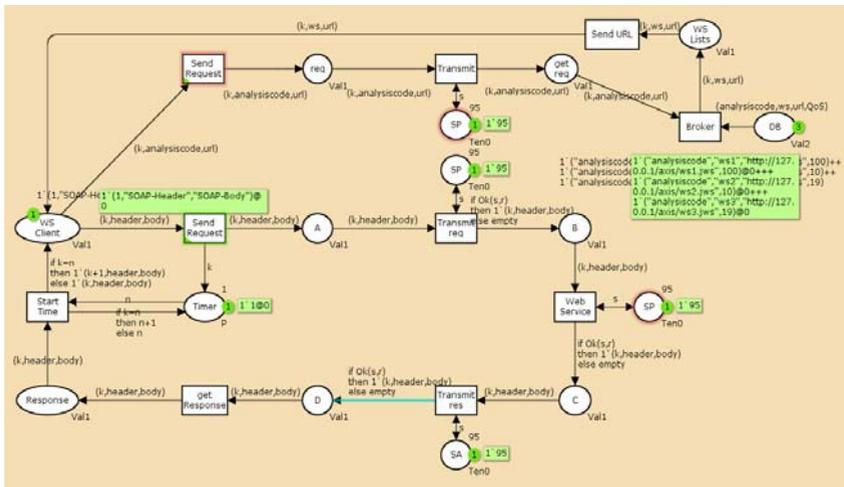


Рис. 3.. Динамическая композиция Web-сервисов с возвратом ответа через брокера

**Глава 3** посвящена вопросам выбора технологических средств, используемых для разработки Web-сервисов и приложений, использующих Web-сервисы. Показывается значимость архитектуры MVC для разработки приложений. Анализируется технология JSF как технология практического построения Web-приложений на основе MVC.

В целях демонстрации методов динамической композиции Web-сервисов в процессе диссертационного исследования разработано Web-приложение, в котором для решения одной и той же задачи используется разная схема переадресации Web-сервисов в процессе выполнения. При этом, в целях проверки совместимости Web-сервисов, построенных с использованием разных технологий, использовались среды разработки и инструментарий Java и PHP.

Для целей исследования можно рекомендовать широко используемый во многих программных разработках инструментальный пакет Apache Axis от Apache Software Foundation, интегрированный в сервер приложений Apache's Jakarta Tomcat. NuSOAP - представляет из себя набор PHP-классов, позволяющих разработчикам создавать и использовать Web-

сервисы, которые используют протокол SOAP для обмена сообщениями. NuSOAP не требует установки PHP-extensions.

Для построения Web-приложений на основе Java используется подход Model-View-Controller (MVC). Согласно этому подходу приложение представляет собой в общем случае организованный набор страниц HTML и JSP, а также сервлетов и компонентов JavaBeans и Enterprise JavaBeans (EJB). Эти элементы Web-приложения организованы в три группы:

- Первая группа, обозначенная в названии подхода термином «Model», предназначена для моделирования прикладной области, или, другими словами, для реализации бизнес процессов. Группа состоит из компонентов JavaBeans и Enterprise JavaBeans (EJB).
- Вторая группа, обозначенная в названии подхода термином «View», предназначена для представления данных Web-приложения клиенту. Группа состоит из страниц HTML и JSP.
- Третья группа, обозначенная в названии подхода термином «Controller», предназначена для управления последовательностью работы компонентов групп «Model» и «View» при реализации ответа на поступивший от клиента запрос. Группа состоит из сервлета-контроллера.

MVC организует диалоговое взаимодействие с клиентом с помощью конкретных модулей этих групп (Рис.4): модули первой группы (их иногда называют командами) обеспечивают доступ к данным и бизнес логику, модули второй группы обеспечивают представление данных и пользовательский вход, а контроллер или диспетчер, организует обработку запросов клиента и обеспечивают соответствующий поток управления и.

Следование архитектуре MVC дает разработчикам Web-проекта определенные преимущества. MVC разделяет модули проекта, уменьшая дублирование кода, централизуя управление, и делая Web-приложение более приспособленным к внесению

изменений. Архитектура MVC способствует коллективной разработке приложений, помогая разработчикам сосредоточиться на выполнении функций, требующих определенной специализации, и взаимодействовать через четкие интерфейсы.

Проект MVC централизует управление такими прикладными компонентами как обеспечение безопасности, вход в систему, вывод на экран. Архитектура MVC способствует подключению к Web-приложению новых источников данных, подключению новых типов клиентов, четко определяя обязанности классов, способствует выявлению и устранению возможных ошибок, то есть максимально способствует успешному сопровождению Web-приложения в течение всего жизненного цикла.

Другим несомненным достоинством архитектуры MVC является возможность стандартизовать модуль контроллера, обеспечивая некоторый каркас или шаблон проектирования, что в англоязычной литературе обозначается термином «framework», дословно – каркас, основа. Нет необходимости каждый раз создавать новый сервлет-контроллер, вполне уместно использовать внешний, грамотно разработанный и настраиваемый «движок».

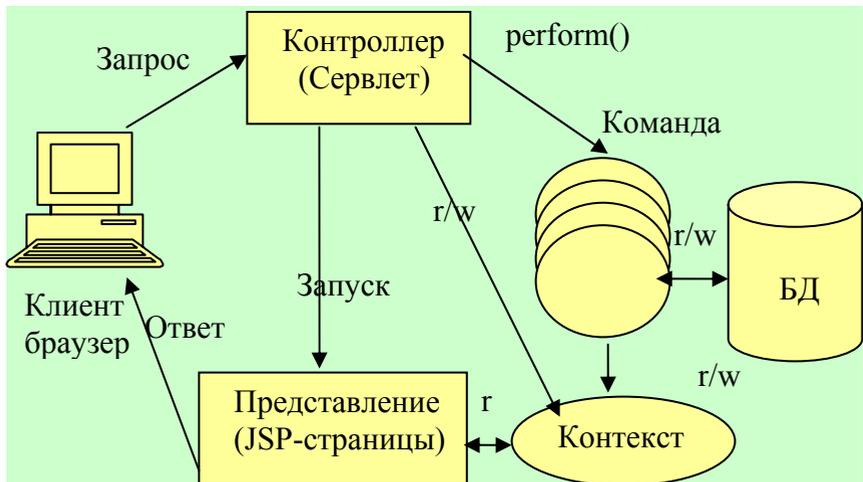


Рис.4. Архитектура Model-View-Controller

Именно в таком качестве выступают технологии JavaServer Faces (JSF) и Struts, используемые при разработке Web-приложений, реализуя шаблон MVC в том или ином виде. Данные технологии направлены на избавление разработчика от реализации повторяющихся, распространенных действий. Как эти, так и другие, менее известные технологии, Struts и JSF используют вариацию MVC – Model 2, ориентированную именно на разработку Web-приложений. Эти технологии являются своеобразными конкурентами и часто сравниваются друг с другом, например. Не вдаваясь в детали, можно сказать, что JSF и Struts имеют частично перекрывающую друг друга функциональность, однако существуют области специфичные для каждой.

**Глава 4** посвящена экспериментальной проверке предлагаемых подходов к статической и динамической композиции Web-сервисов на основе разработанного прототипа Web-приложения.

Для моделирования статической и динамической композиции Web сервисов необходимо разработать прототип Web приложения, на котором можно было бы продемонстрировать все особенности как статического, так и рассмотренных вариантов динамического связывания Web сервисов. Чтобы не выдумывать предметную область, сценарий работы прототипа взят из раздела «Информационные технологии и электроника» отчета НИЦ ЭЛДИС РАН по теме «Методы передачи медицинской информации, включая графическую, в телекоммуникационных сетях телемедицины».

В целом отчет посвящен выработке методологии построения функциональных национальных стандартов, совместимых со стандартом HL7. Разумеется, для целей данного исследования нет особой необходимости в точности следовать этим стандартам.

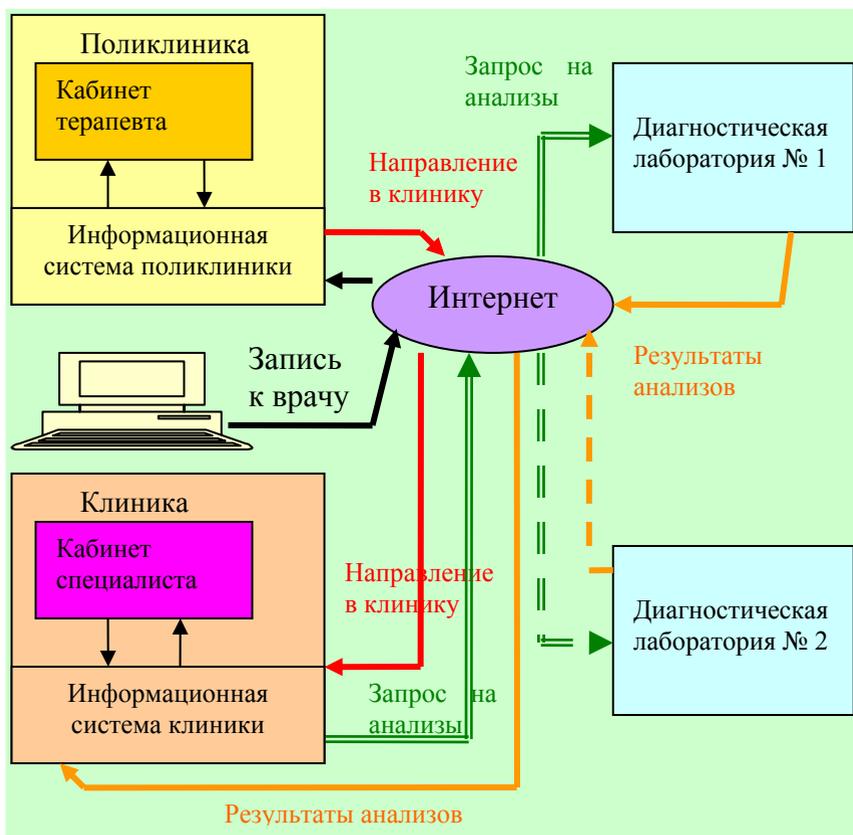


Рис. 5 Статическая схема передачи сообщений

Как прототип информационной системы поликлиники – «ПОЛИКЛИНИКА», так и прототип информационной системы клиники «КЛИНИКА» разработаны как Web-приложения. В основу архитектуры этих систем положен шаблон MVC в вариации Model 2, так как именно он ориентирован на разработку Web-приложений. Как для реализации прототипа информационной системы «ПОЛИКЛИНИКА», так и системы «КЛИНИКА» использовалась технология Java Server Faces (JSF). На рис. 5 представлена архитектура соответствующего приложения. В состав этих информационных систем включены Web-сервисы, которые

позволяют внешним пользователям получить доступ к функциональности этих систем.

На рис. 5. представлены две диагностические лаборатории. Услуги этих лабораторий реализованы как Web-сервисы, причем в разных вариантах. Варианты этих Web-сервисов всегда имеют одинаковую функциональность, но различаются:

1. В первом варианте Web-сервисы имеют различные WSDL, что, к сожалению, можно наблюдать в реальности.
2. Во втором варианте Web-сервисы имеют идентичные WSDL, что, вообще говоря, и должно быть, если следовать стандарту HL7, но различаются по нефункциональным характеристикам.

При выполнении диссертационного исследования был реализован еще один Web-сервис, который представляет функциональность объекта, не представленного на рис.5. В идеале этот объект - некое отраслевое UDDI – своеобразный информационный брокер, который занимается публикацией Web-сервисов, осуществляет их классификацию, анализирует QoS характеристики публикуемых Web-сервисов, осуществляет мониторинг реального использования Web-сервисов и т.д. Реально реализованный Web-сервис выполняет эти функции в урезанном варианте.

В диссертационном исследовании рассматриваются только информационно-алгоритмические аспекты динамического связывания Web-сервисов и не рассматриваются вопросы бизнес-моделей такой композиции, несомненно поднимаемых этим подходом.

Еще раз отметим, что термин «динамическая композиция Web-сервисов» понимается в смысле возможной переадресации клиента к другому сервису в процессе выполнения.

В заключении приведены основные результаты диссертационной работы.

## ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

1. Проведен анализ подходов к динамическому связыванию (динамической композиции, динамической переадресации) Web-сервисов и предложены конкретные варианты динамической композиции с помощью включения дополнительной службы в процесс взаимодействия клиента и сервера.
2. В диссертации показано, что динамическое связывание Web-сервисов эффективно при создании дополнительной службы, выполняющей функции мониторинга операционной обстановки, в которой происходит эксплуатация Web-приложений, использующих Web-сервисы. Динамическая переадресация, не опирающаяся на текущий мониторинг, может производиться только на основе статических QoS характеристик Web-сервисов, которые не отражают складывающуюся ситуацию.
3. Разработано два варианта функционирования дополнительной службы, названной информационным брокером. В первом варианте служба представляет стандартизованный интерфейс к собственному Web-сервису, который фактически выбирает для выполнения Web-сервис третьего производителя. Этот вариант наиболее подходит к случаю, когда разработано несколько практически функционально идентичных Web-сервисов, но с разными WSDL. Во втором варианте служба возвращает URL Web-сервиса, который наиболее подходит в конкретной ситуации. Этот вариант предназначен для случая, когда разработано несколько функционально идентичных Web-сервисов с идентичными WSDL.
4. В качестве основы для архитектуры Web-приложений предлагается использовать шаблон Model-View-Controller (MVC) в исполнении Java Server Faces (JSF), так как Web-сервисы естественно вписываются в эту архитектуру в качестве команд модели.

5. Для демонстрации процессов динамического связывания разработаны специальные программные средства и Web-приложения на базе платформы Java и PHP, представляющие идею интеграции разнородных Web-сервисов с использованием динамической композиции.
6. Моделирование процессов динамического связывания Web-сервисов выполнено на базе многоцветных сетей Петри. Модели в виде многоцветных сетей Петри могут быть построены на разных уровнях детализации запросов клиента и ответов Web-сервиса. Задание временных задержек на переходах/дугах позволяет получить оценку времени получения ответов на запросы клиента, моделируя различные варианты динамического связывания.

По теме диссертации опубликованы следующие работы:

1. Пхуон Найнг, Технология MVC // «Научная сессия МИФИ-2004. Сборник научных трудов. Т.2. Программное обеспечение технологии» М.: МИФИ, 2004.
2. Пхуон Найнг, Мониторинг Web-сервисных приложений // «Современные технологии в задачах управления, автоматизации и обработки информации: Труды XIV Международного научного технического семинара», Алушта-Самара: Самарский государственный аэрокосмический университет, 2005.
3. Пхуон Найнг, Сервис-ориентированная архитектура // «Научная сессия МИФИ-2006. Сборник научных трудов. В 15 томах. Т.2. Программное обеспечение технологии» М.: МИФИ, 2006.
4. Пхуон Найнг, Система Мониторинга для Web-сервисов // «Современные технологии в задачах управления, автоматизации и обработки информации: Труды XV Международного научного технического семинара», Алушта. – М.: МИФИ, 2006
5. Пхуон Найнг, Интеграция бизнес-решений с использованием Web-Сервисов // Сборник научных трудов. В 15 томах. Т.2. Программное обеспечение технологии» М.: МИФИ, 2007.