

Министерство образования Республики Беларусь  
Учреждение образования  
Белорусский государственный университет  
информатики и радиоэлектроники

УДК 004.75:004.77

Турков  
Андрей Валерьевич

МОДЕЛИРОВАНИЕ И ОБРАБОТКА ДАННЫХ В  
СЕРВИС-ОРИЕНТИРОВАННЫХ ПРОГРАММНЫХ СИСТЕМАХ

**АВТОРЕФЕРАТ**

соискание академической степени  
магистра технических наук

по специальности 1-40 80 05 – Математическое и программное обеспечение  
вычислительных машин, комплексов и компьютерных сетей

Научный руководитель  
Бранцевич П.Ю.  
к.т.н., доцент

Минск 2016

## КРАТКОЕ ВВЕДЕНИЕ

Повышение требований к современным информационным системам приводит к появлению новых архитектурных подходов к их созданию. Современные предприятия сталкиваются с проблемами, обусловленными ростом объема данных, необходимостью интеграции с партнёрами и новыми подразделениями, обновлением программной платформы предприятия с сохранением старых, но проверенных временем систем. Одним из современных подходов к решению описываемых проблем является использование сервис-ориентированной архитектуры.

К основным достоинствам сервис-ориентированной архитектуры относятся: открытость за счёт использования стандартов, стандартизация формата описания веб-сервисов; независимость от используемых платформ и языков программирования; ориентированность на сетевое взаимодействие и др.

В настоящее время сервис-ориентированная архитектура широко применяется во многих крупных информационных системах таких предприятий, как Сбербанк, Белавиа, Adobe, eBay и многих других. При этом за счёт открытости подхода создано несколько платформ для организации сервис-ориентированной архитектуры, реализующих все необходимые функции и распространяемых на условиях лицензии с открытым исходным кодом. Это позволяет реализовывать новые возможности малым и средним предприятиям.

Основные положения сервис-ориентированной архитектуры заключаются в применении композиции (объединения) нескольких автономных веб-сервисов для решения определенных задач. Современные информационные системы часто используют в своей работе не только локальные сервисы, над которыми возможен контроль, но и внешние, которые с позиции потребителя услуг представлены в виде «чёрного ящика». К тому же, зачастую одну и ту же функцию предоставляют несколько веб-сервисов. Например, среди наиболее востребованных в потребительском секторе можно выделить функции картографии, предоставления информации по географическим объектам, размещения видео, графических и аудиофайлов, и т.д.

В связи с этим, при проектировании систем с сервис-ориентированной архитектурой возникают две связанные задачи: рациональный выбор композиции веб-сервисов для решения определенной задачи и формирование набора критериев сравнения веб-сервисов. Под рациональным выбором понимается действие лица, принимающего решение, с учетом всей имеющейся у него информации, которое приводит к наиболее предпочтительным для него результатам.

## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

### Цель и задачи исследования

*Целью* диссертационной работы является проведение теоретического обоснования, анализа, разработки и развития методов, алгоритмов и моделей информационных систем с сервис-ориентированной архитектурой, а также повышение эффективности и надежности процессов обработки данных в слабосвязанных системах.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Анализ особенностей сервис-ориентированной архитектуры приложения, а также современных методов, моделей и программных средств её реализации.

2. Анализ состояния работ в области систем организации доступа к веб-сервисам и обработки показателей качества обслуживания веб-сервисов.

3. Исследование распределенной системы вычисления, хранения и агрегирования значений показателей качества обслуживания веб-сервисов.

4. Разработка концепции функционального резервирования веб-сервисов в условиях динамически меняющегося состава веб-сервисов и изменения значений показателей качества обслуживания.

5. Создание программного комплекса с сервис-ориентированной архитектуры на основе использованных методов, моделей и алгоритмов.

*Объектом* исследования являются системы с сервис-ориентированной архитектурой, а также слабосвязанные системы, основанные на сервис-ориентированной архитектуре.

*Предметом* исследования является программное обеспечение систем обработки данных в сервис-ориентированных приложениях с использованием методов и алгоритмов построения, и реконфигурации композиции веб-сервисов на основе показателей качества обслуживания веб-сервисов.

Основной *гипотезой*, положенной в основу диссертационной работы, является возможность разработки методов и алгоритмов, обеспечивающих повышение надежности и эффективности функционирования программных информационных систем, основанных на динамической композиции веб-сервисов.

### Связь работы с приоритетными направлениями научных исследований и запросами реального сектора экономики

Работа выполнялась в соответствии с научно-техническим заданием и планом работ кафедры «Программное обеспечение информационных технологий» по теме «Разработать модели, методы, алгоритмы для оценки параметров, повышения надежности и качества функционирования аппаратно-программных

средств систем и сетей сложной конфигурации и внедрить в современные обучающие комплексы» (ГБ № 11-2004, № ГР 20111065, научный руководитель НИР – В. В. Бахтизин).

### **Личный вклад соискателя**

Результаты, приведенные в диссертации, получены соискателем лично. Вклад научного руководителя П.Ю. Бранцевича, заключается в формулировке целей и задач исследования.

### **Апробация результатов диссертации**

Основные положения диссертационной работы докладывались и обсуждались на IX МНМК «Дистанционное обучение – образовательная среда XXI века» (Минск, Беларусь, 2015), XII Международной научно-практической конференции «Управление информационными ресурсами» (Минск, Беларусь, 2015).

### **Опубликованность результатов диссертации**

По теме диссертации опубликовано 3 печатных работы в сборниках трудов и материалов международных конференций.

### **Структура и объем диссертации**

Диссертация состоит из введения, общей характеристики работы, четырех глав, заключения, списка использованных источников, списка публикаций автора и приложений. В первой главе представлен анализ предметной области, выявлены основные существующие проблемы в рамках тематики исследования, показаны направления их решения. Вторая глава посвящена проблеме управления ресурсами в сервис-ориентированных программных системах. В третьей главе предложены методы и алгоритмы функционального резервирования веб-сервисов в условиях динамически меняющегося состава веб-сервисов и изменения значений показателей качества обслуживания. В четвертой главе предложена практическая реализация ПО для сервис-ориентированной программной системы.

Общий объем работы составляет 68 страниц, из которых основного текста – 58 страниц, 18 рисунков на 17 страницах, 1 таблица на 1 страницах, список использованных источников из 33 наименования на 3 страницах.

## ОСНОВНОЕ СОДЕРЖАНИЕ

Во **введении** определена область и указаны основные направления исследования, показана актуальность темы диссертационной работы, дана краткая характеристика исследуемых вопросов.

В **первой** главе проведен анализ особенностей сервис-ориентированных систем, выявлены основные существующие проблемы в рамках тематики исследования, показаны направления их решения. Описана тенденция развития от клиент-серверной архитектуры к сервис-ориентированной. Сформулированы основные принципы построения и концепции сервис-ориентированных систем.

В последние несколько лет развития области информационных технологий (ИТ) наметился переход от всеобъемлющих информационных систем (ИС), включающих все необходимые для работы данные и функции, к распределенным системам, использующим внешние ресурсы [3]. Такой переход оправдан следующими факторами:

1. *Рост объёма данных.* Одна система зачастую уже не способна хранить весь объём необходимых данных и обеспечивать приемлемый уровень производительности вычислений.

2. *Интеграция с партнёрами и новыми подразделениями.* Слияние и поглощение организацией ведёт к потребности совмещения гетерогенных систем и приложений, функционирующих по различным протоколам.

3. *Обновление платформы предприятия.* Многие организации не желают отказываться от проверенных и надежных технологий, однако развитие бизнеса приводит к необходимости использования новых программных средств, часто несовместимых с имеющейся ИС [6].

Одним из современных решений обозначенных задач является построение систем на базе сервис-ориентированной архитектуры (СОА). СОА, согласно [4], — это тип архитектуры распределенных систем, характеризуемый следующими принципами:

1. *Логическое представление.* Сервис является абстрактным, логическим представлением программ, баз данных, бизнес-процессора и т.д. Представление определено в терминах фактически выполняемых действий, обычно операций уровня бизнес-логики.

2. *Ориентированность на обмен сообщениями.* Сервис определяется сообщениями, которыми он обменивается с поставщиками и потребителями услуг. Внутренняя структура сервиса (например, язык программирования, с помощью которого реализован сервис, структура процессов, схема БД) намеренно неизвестна: потребитель сервиса не должен знать детали реализации.

3. *Ориентированность на машинно-читаемое описание.* Сервис описывается в мета-данными, доступными для машинной обработки. Такое описание

соответствует публичному качеству СОА: только те детали, которые должны быть известны потребителю услуг должны быть включены в описание.

4. *Степень детализации.* Сервисы должны иметь как можно меньшее число доступных операций для работы с относительно большими и сложными сообщениями.

5. *Ориентированность на сетевое взаимодействие.* Доступ к сервисам, как правило, реализован с помощью компьютерной сети, однако это не является обязательным.

6. *Независимость от платформ.* Сообщения, которыми обмениваются сервисы по интерфейсам, отправляются в платформо-независимом, стандартизированном формате. XML (англ. eXtensible Markup Language, расширяемый язык разметки) является наиболее наглядным форматом, который удовлетворяет этому ограничению.

**Вторая** глава посвящена проблеме управления ресурсами в сервис-ориентированных системах и композиции таких систем.

В целом сервис-ориентированную архитектуру можно охарактеризовать как подход к разработке программного обеспечения, основанный на использовании служб (сервисов), со стандартизированными интерфейсами [20]. Интерфейс сервиса описывается с помощью формального, независимого от платформы, языка, например, WSDL. Такой подход позволяет достаточно просто интегрировать логику и ресурсы ранее созданных информационных систем для решения новых задач.

Выделим особенности управления ресурсами, в зависимости от выделенных моделей использования. Модели IaaS и PaaS предполагают, что динамические характеристики выполняемых приложений заведомо случайны, так как пользователь может решать произвольные задачи примерно так, как это происходит в операционных системах общего назначения. Модель SaaS отличается от моделей IaaS и PaaS тем, что здесь поставщик контролирует весь жизненный цикл приложения и может знать заранее статические и динамические требования к ресурсам.

Задача управления ресурсами осложняется еще и требованиями к качеству сервиса: время гарантированного отклика, уровень пропускной способности, доступность, отказоустойчивость, время восстановления после сбоя и т.д. Поэтому универсального решения задачи оптимального управления ресурсами в моделях IaaS и PaaS нет даже теоретически. В основном здесь ведутся исследования в области виртуализации элементов вычислительных систем. Современные достижения в этом направлении отражены в работах [21, 22].

Модель SaaS предоставляет больше возможностей по управлению ресурсами. Основная идея – зная характеристики задачи, более точно определять, когда и какие ресурсы должна получить задача с учетом текущего состояния ресурсов распределенной вычислительной системы.

В работе [23] предложено рассматривать 4 типа ресурсов (процессор, оперативная память, устройства ввода/вывода и устройства передачи данных) и принимать решение о балансировке исходя из учета потребностей выполняемой задачи именно в рамках этих четырех типов ресурсов. В этой же работе была предложена модель промежуточного программного обеспечения для балансировки нагрузки устройств ввода/вывода и сети в виде единой шины.

Основные положения сервис-ориентированной архитектуры заключаются в применении композиции нескольких автономных веб-сервисов для решения определенных задач. Современные информационные системы часто используют в своей работе не только локальные сервисы, над которыми возможен контроль, но и внешние, которые с позиции потребителя услуг представлены в виде «чёрного ящика».

Главная проблема композиции веб-сервисов заключается в определении того, являются ли компоненты сервисов композитными. Существует два типа правил композитности, использующие синтаксические и семантические свойства веб-сервисов. Синтаксические правила сравнивают типы операций и протокол взаимодействия сервисов. Семантические правила сравнивают число параметров сообщений, бизнес-роли и единицы измерения. Операции семантической композитности сравнивают семантику сервисных операций [2]. Композитность по параметрам качества сравнивает качественные свойства Веб-сервисов. Целостность или непротиворечивость сервисов проверяет, даст ли выбранная комбинация Веб-сервисов ожидаемый результат. Развитие технологий семантических веб-сервисов даёт возможность создавать единственное унифицированное представление семантического веб-сервиса в различных применениях, позволит точно находить необходимую информацию, упрощать корпоративную и сетевую интеграцию в распределенном веб-окружении [2].

**Третья глава** посвящена проектированию модели отказоустойчивой системы веб-сервисов. Вводится понятие надежности и отказоустойчивости системы с сервис-ориентированной архитектурой. Предложены методы и алгоритмы адаптивной отказоустойчивой системы доступа к функционально подобным веб-сервисам.

Существует четыре технических аспекта в создании надежного программного обеспечения: предотвращение отказа, устранение отказа, обеспечение отказоустойчивости, прогнозирование отказа. Так как полностью исключить отказы программного обеспечения практически невозможно, то основным принципом создания высоконадежных программных систем является обеспечение отказоустойчивости.

На основе комплексных показателей надежности и эффективности веб-сервисов, можно определить следующие основные показатели, характеризующие качество обслуживания веб-сервисов: время ответа, вероятность отказа, готовность, доступность, стоимость и достоверность.

Получение объективной оценки веб-сервиса по интернет-зависимым показателям QoWS для каждого конкретного случая требует вычисление значения показателей на стороне клиента, вычислительные ресурсы которого могут быть ограничены. Вычисление значений показателей QoWS на стороне веб-сервиса относительно географически распределенных пользователей без их непосредственного участия и хранение статистики их изменения в централизованной базе данных – тоже не простая задача. Необходимо компромиссное двустороннее решение, которое позволит использовать накопленную провайдерами и пользователями статистику изменения значений показателей QoWS для эффективной и продуктивной оценки качества веб-сервисов при организации слабосвязанных распределенных систем обработки данных.

Из результатов исследований предлагаемых подходов к вычислению и прогнозированию показателей QoWS, представленных в научно-технической литературе и Интернет, видно, что для объективной оценки качества веб-сервисов необходимы достаточно большие наборы данных QoWS реальных веб-сервисов, распределенных в сети Интернет. Следовательно, требуется эффективный и действенный механизм вычисления и агрегирования распределенных наборов данных QoWS.

Выбор нескольких стратегий прогнозирования позволит учесть ограничения в вычислительных ресурсах и предпочтения пользователя.

Исследование различных подходов к обеспечению отказоустойчивости отдельных веб-сервисов также показало невозможность их применения к созданию отказоустойчивых интернет-ориентированных слабосвязанных систем в целом с заданными требованиями эффективности и переменным динамически меняющимся составом веб-сервисов. Существующие подходы не учитывают необходимости динамической реконфигурации системы и контекстно зависимого от требований пользователя и изменения интернет-среды выбора композиции веб-сервисов. Решением проблемы может служить разработка программного обеспечения, позволяющего автоматизировать реконфигурацию слабосвязанных систем (изменение композиции веб-сервисов) на основе постоянно обновляемых пользователями текущих и системно прогнозируемых значений показателей QoWS.

Таким образом, организация взаимодействия информационных систем по принципу слабого связывания посредством веб-сервисов требует разработки адаптивной стратегии выбора отказоустойчивой композиции веб-сервисов, учитывающей предпочтения пользователей и динамику изменения показателей QoWS.

Можно выделить два основных типа стратегий повышения отказоустойчивости композиции веб-сервисов – последовательные и параллельные стратегии. К последовательным относятся такие стратегии, как *Повтор* и *Замещение* (рису-

нок 1), так как они затрачивают дополнительное (избыточное) время на повышение отказоустойчивости системы. К параллельным относятся стратегии *Один из n* и *Первый из n* (рисунок 2). Обе последние стратегии основаны на избыточности программных и аппаратных ресурсов.

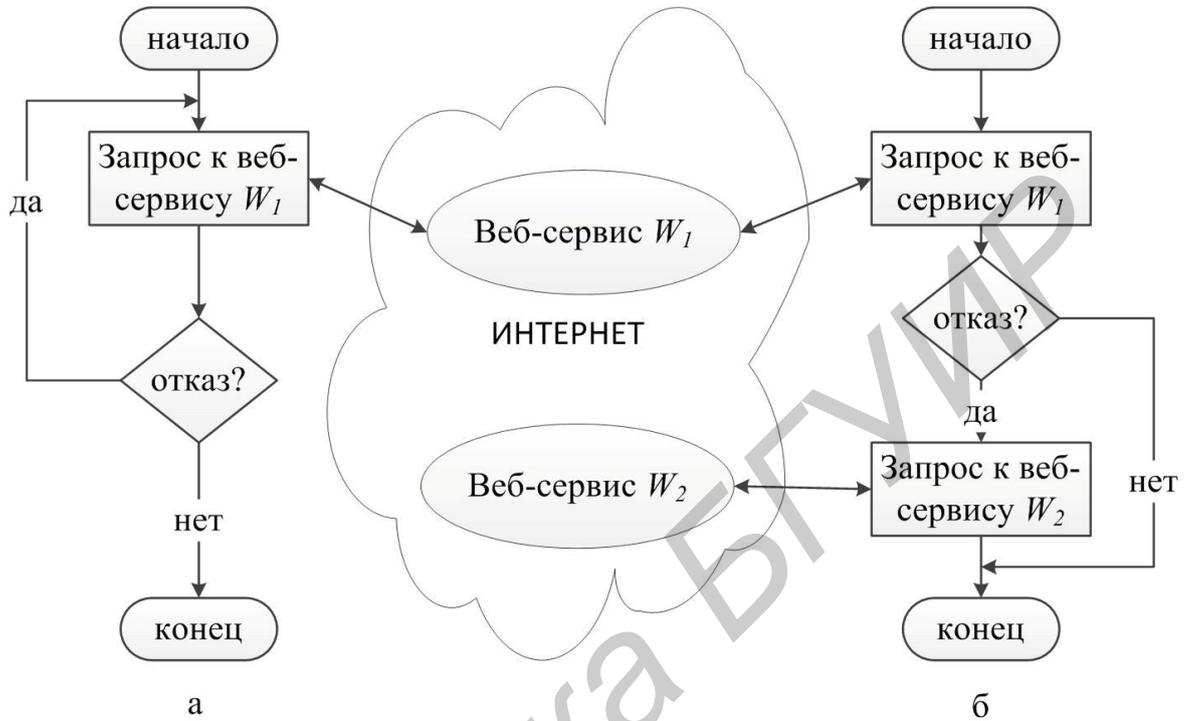


Рисунок 1 – Параллельные стратегии (а) *Повтор*, (б) *Замещение*

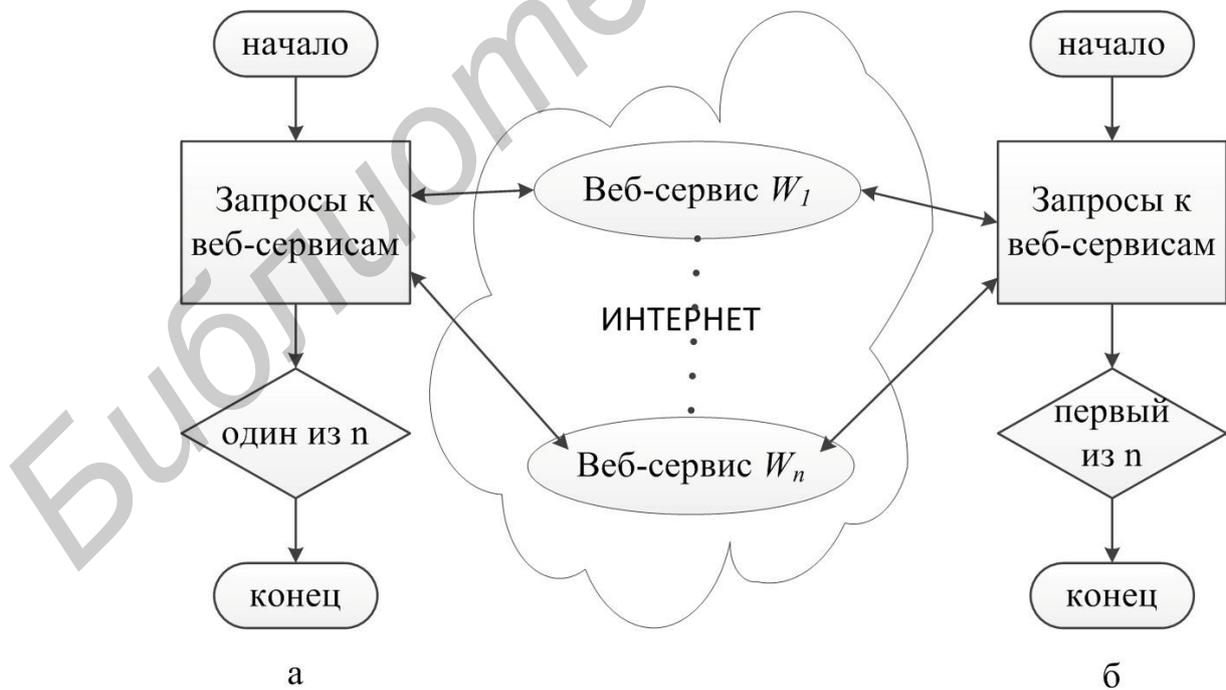


Рисунок 2 – Последовательные стратегии (а) *Один из n*, (б) *Первый из n*

Комбинированная последовательная стратегия выбора отказоустойчивой композиции веб-сервисов представляет собой комбинацию стратегий *Повтор* и *Замещение* (рисунок 3а). В случае отказа вызываемого веб-сервиса стратегия *Повтор-Замещение* на основе требований пользователя и показателей качества обслуживания экземпляров функционально подобных веб-сервисов определяет, надо ли повторно вызывать отказавший веб-сервис или перейти к вызову его экземпляра – кандидата на выполнение требуемой функции.

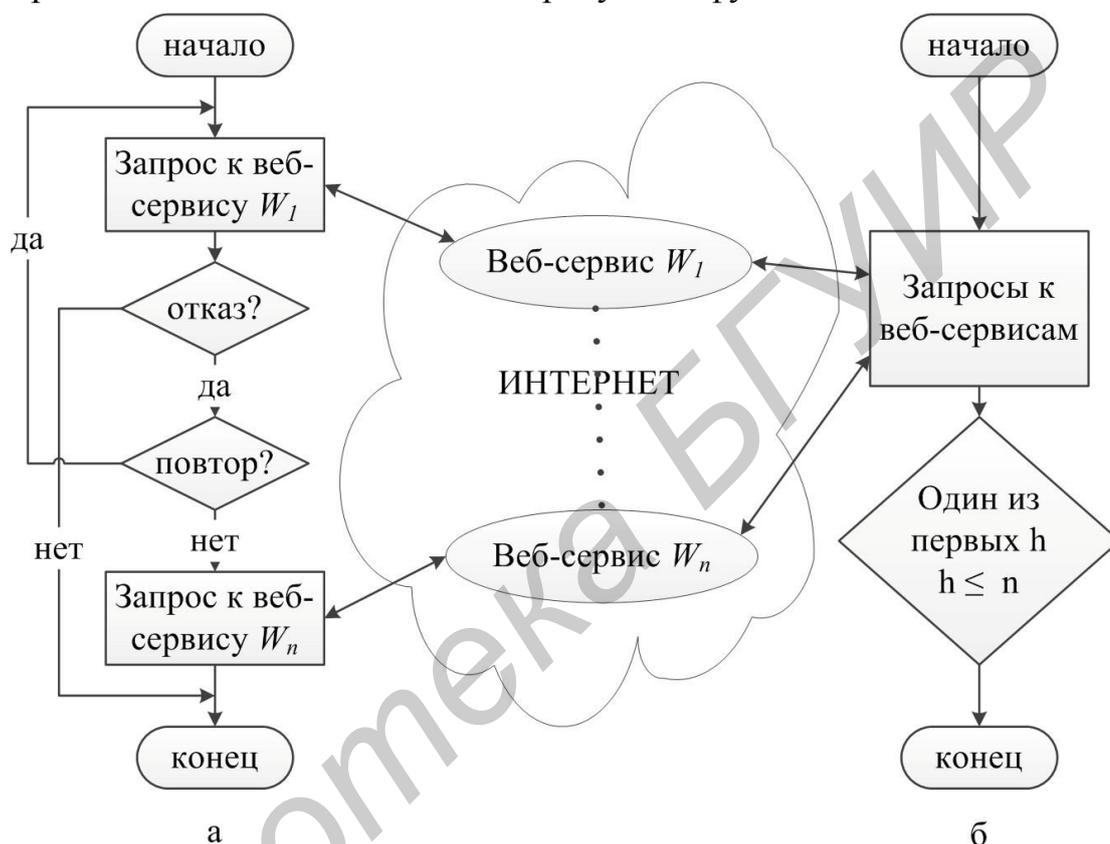


Рисунок 3– Комбинированные стратегии (а) последовательная, (б) параллельная

Комбинированная параллельная стратегия представляет собой комбинацию стратегий *Один-из-n* и *Первый-из-n* (рисунок 3.3б). В соответствии с этой стратегией будут одновременно вызваны  $n$  экземпляров функционально подобных веб-сервисов и использованы  $v$  ответов, удовлетворяющих заданным условиям, для последующего определения голосованием окончательного результата работы веб-сервиса. Эта стратегия идентична стратегии *Первый-из-n* при  $h = 1$  и идентична стратегии *Один-из-n* при  $h = n$ .

Исследование последовательных и параллельных стратегий отказоустойчивости отдельных веб-сервисов показало невозможность их применения к созданию отказоустойчивых слабосвязанных систем в целом с заданными требованиями эффективности и переменным динамически меняющимся составом веб-сервисов. Решением проблемы может служить разработка адаптивной отказо-

устойчивой системы доступа к функционально подобным веб-сервисам, на основе разработанного алгоритма стратегии отказоустойчивости и параметрической модели стратегии отказоустойчивости постоянно обновляемых пользователями текущих и системно прогнозируемых показателей качества обслуживания веб-сервисов.

В четвертой главе рассмотрена практическая реализация ПО для сервис-ориентированной системы. В области организации взаимодействия программного обеспечения стремление к повторному использованию готовых интеграционных решений и их программных реализаций нашло свое отражение в разработке сервис-ориентированной архитектуры и различных типовых решений, положенных в основу многоуровневой организации корпоративных информационных систем. Архитектурные слои или уровни не имеют четких границ по их применению и физическому размещению на узлах распределенной вычислительной системы.

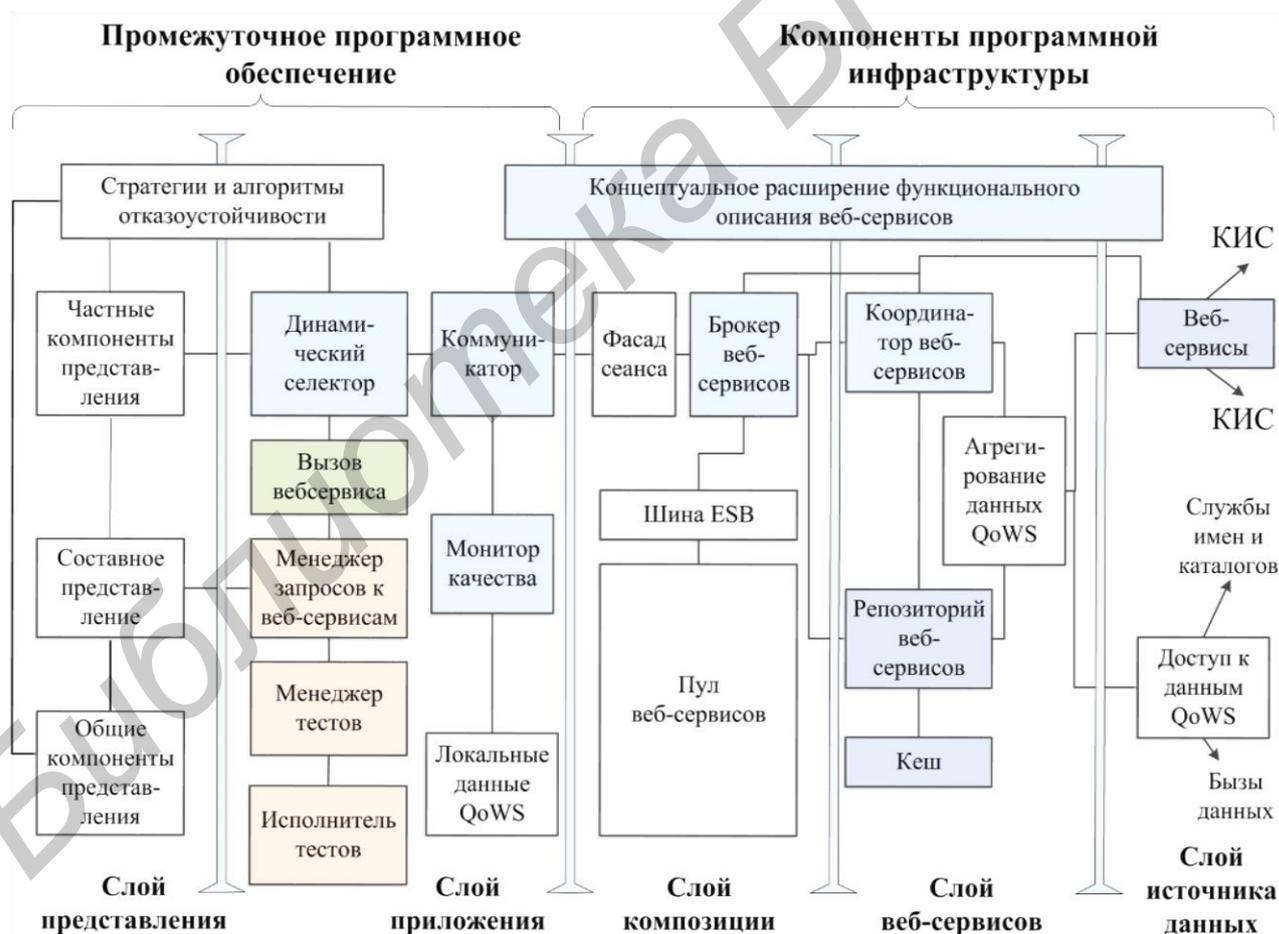


Рисунок 4 – Архитектурные слои программной реализации

На рисунке 4 представлены только основные типовые решения, определяющие логику расслоения системы и отвечающие функциональному назначению

элементов инфраструктуры и компонентов адаптивной отказоустойчивой системы. Типовые решения проектирования, такие как, объект переноса данных (DTO, Data Transfer Object), и связи между ними не показаны на рисунке. Согласно рассмотренной выше схеме расслоения Маринеску[34] такое разделение системы четко разграничивает ответственность и назначение каждого слоя и позволяет с минимальными трудозатратами вносить в них изменения [64]. Предметная логика элементов программной инфраструктуры и адаптивной отказоустойчивой системы распределена между слоем композиции и слоем веб-сервисов.

## **ЗАКЛЮЧЕНИЕ**

### **Основные научные результаты диссертации**

В ходе диссертационного исследования наряду с преимуществом сервис-ориентированного подхода к организации взаимодействия программ и программных систем были выявлены основные недостатки существующих методов, алгоритмов и средств организации взаимодействия корпоративных информационных систем на основе сервис-ориентированной архитектуры и веб-сервисов. Большинство из них объясняется несогласованностью поставщиков веб-сервисов при использовании отдельно взятых языков, протоколов и стандартов применительно к разработке как самих веб-сервисов, так и ориентированных на них систем.

Предложена концептуальная модель программной инфраструктуры для организации взаимодействия корпоративных информационных систем, позволяющая на случай отказа основного веб-сервиса реализовать избыточность композиции посредством резервирования функционально подобными веб-сервисами.

Предложен алгоритм адаптивной отказоустойчивой системы доступа к функционально подобным веб-сервисам, позволяющий перестраивать композиции веб-сервисов с учетом изменения состава представленных в сети Интернет функционально подобных веб-сервисов, значений показателей качества обслуживания веб-сервисов и характеристик самой интернет-среды.

Предложена программная инфраструктура и программное обеспечение для организации доступ к функционально подобным веб-сервисам, позволяющее повысить отказоустойчивость композиции веб-сервисов за счет избыточности ее компонентов – веб-сервисов корпоративных информационных систем.

Предложена компонентная архитектура и типовые решения проектирования, позволяющие автоматизировать процесс разработки программной инфраструктуры и программного обеспечения для адаптивной отказоустойчивой системы организации взаимодействия корпоративных информационных систем посредством веб-сервисов.

## **Рекомендации по практическому использованию результатов**

Полученные результаты формируют теоретическую и практическую базу для разработки ПО компьютерных систем для решения задач создания отказоустойчивой сервис-ориентированной системы.

Разработанные методы и алгоритмы анализа могут применяться в автоматизированных отказоустойчивых системах.

## **СПИСОК ОПУБЛИКОВАННЫХ РАБОТ**

1. Турков А.В. Использование сервис-ориентированной системы для повышения качества дистанционного обучения / А.В. Турков, Н.А. Горбачевский // «Дистанционное обучение – образовательная среда XXI века»: материалы IX международной научно-методической конференции аспирантов, магистрантов и студентов. – Минск: БГУИР, 2015. – с. 289.

2. Горбачевский Н.А. Визуализация с использованием веб-сервисов как способ повышения эффективности дистанционного обучения / Н.А. Горбачевский, А.В. Турков // «Дистанционное обучение – образовательная среда XXI века»: материалы IX международной научно-методической конференции аспирантов, магистрантов и студентов. – Минск: БГУИР, 2015. – с. 252.

3. Турков А.В. Использование построенной на веб-сервисах визуализации в процессе подготовки управленческих кадров / А.В. Турков, Н.А. Горбачевский // «Управление информационными ресурсами»: материалы XII международной научно-практической конференции аспирантов, магистрантов и студентов. – Минск: АУПРБ, 2015. – с. 108.