DOI: 10.15514/ISPRAS–2020–32(5)–12

Модельный подход к обеспечению безопасности и надежности Web-сервисов

[1,2] *Е.М. Лаврищева, ORCID: 0000-0002-1160-1077 <lavr@ispras.ru>*
[1,3] *С.В. Зеленов, ORCID: 0000-0003-0446-0541 <zelenov@ispras.ru>*

[1] *Институт системного программирования им. В.П. Иванникова РАН, 109004, Россия, г. Москва, ул. А. Солженицына, д. 25*
[2] *Московский физико-технический институт, 141701, Россия, Московская обл., г. Долгопрудный, Институтский пер., д. 9*
[3] *Национальный исследовательский университет «Высшая школа экономики» 101000, Россия, г. Москва, ул. Мясницкая, д. 204*

Аннотация. Дается анализ проблематики надежности и безопасности в мировой практике и в нашей стране. Рассмотрены аспекты моделирования программно-технических систем (ПТС) из сервисных ресурсов и готовых КПИ с обеспечением надежности и безопасности. Приводятся сформировавшиеся базовые и теоретические основы проблемы моделирования, опыта использования современных сервисных средств SOA, SCA, SOAP в ПТС и Web-системах с обеспечением их надежности и безопасности в Интернет. Отмечается, что ПТС и Web-системы создаются методом сборки в современных средах: IBM WSDK + WebSphere, Apache Axis + Tomcat; Microsoft .Net + IIS и др. Должны проводиться верификация и тестирование систем для поиска ошибок, возникающих при исключительных ситуациях (кибератаках, запрещенных доступах к БД и др.). Описаны методы анализа таких ситуаций и применения методов надежности и безопасности для обеспечения устойчивой и безотказной работы сервисных компонентов ПТС в информационной среде Интернет.

Ключевые слова: безопасность; надежность; качество; сервисный ресурс; сервисные службы; Web-системы; анализ уязвимости; ошибочные ситуации; оценка качества

Для цитирования: Лаврищева Е.М., Зеленов С.В. Модельный подход к обеспечению безопасности и надежности Web-сервисов. Труды ИСП РАН, том 32, вып. 5, 2020 г., стр. 153-166. DOI: 10.15514/ISPRAS–2020–32(5)–12

Благодарности: Работа поддержана грантом РФФИ № 19-01-00206.

Model-Based Approach to Ensuring Reliability and Security of Web-services

[1,2] *E.M. Lavrischeva, ORCID: 0000-0002-1160-1077 <lavr@ispras.ru>*
[1,3] *S.V. Zelenov, ORCID: 0000-0003-0446-0541 <zelenov@ispras.ru>*

[1] *Ivannikov Institute for System Programming of the Russian Academy of Sciences,*
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia,
[2] *Moscow Institute of Physics and Technology (MIPT)*
141700, Russia, Moscow region, Dolgoprudny, Campus per., 9.
[3] *National Research University Higher School of Economics (HSE)*
20 Myasnitskaya Ulitsa, Moscow, 101000, Russia

Abstract. In the paper, we analyze problems of reliability and security in the world practice and in Russia. We consider aspects of modeling software/hardware systems from service resources and ready-made reuses with ensuring reliability and security. We present the formed basic and theoretical foundations of the modeling problem, the experience of using modern service tools SOA, SCA, SOAP in software/hardware systems and Web systems to ensure their reliability and security on the Internet. We note that software/hardware systems and Web systems are created by the assembly build method in modern environments: IBM WSDK + WebSphere, Apache Axis + Tomcat; Microsoft .Net + IIS, etc. Verification and testing of systems should be conducted for searching of errors that occur in exceptional cases (cyber-attacks, forbidden access to the database, etc.). We describe methods for analyzing such situations and applying reliability and security methods to ensure stable and trouble-free operation of software/hardware systems service components in the Internet information environment.

Keywords: security; reliability; quality; service resource; service services; Web systems; vulnerability analysis; erroneous situations; quality assessment

For citation: Lavrischeva E.M., Zelenov S.V. Model-Based Approach to Ensuring Reliability and Security of Web-services. Trudy ISP RAN/Proc. ISP RAS, vol. 32, issue 5, 2020, pp. 153-166 (in Russian). DOI: 10.15514/ISPRAS–2020–32(5)–12

Acknowledgments. This work was supported by the RFBR grant no. 19-01-00206.

1. Введение

1.1 Исторические аспекты надежности техники

Теория надежности как самостоятельная научная дисциплина начала складываться после окончания 2-й мировой войны. Военное ведомство США начало интенсивно финансировать работы, связанные с качеством вооружений самого разного назначения. В СССР работы по надежности техники, связанные с военной радиофизической техникой в авиационной, космической, автомобильной и морской областях, были инициированы военно-промышленным комплексом (ВПК) страны.

С 1950г. в СССР проводились разработки по создания надежной техники и оборудования в военной отрасли по постановлению Минрадиопрома СССР и проведения конкретных разработок по программе ВПК (1950-1992). Такие работы проводятся и сейчас в оборонной промышленности в рамках защиты наших земных и атмосферных границ.

В компьютерной науке в 1986 г. появилось усовершенствованное направление классической теории надежности для технических средств – dependability (гарантоспособность), включающее в себя вопросы обеспечения отказоустойчивости, готовности, живучести, достоверности, целостности и конфиденциальности. Современные сервис-ориентированные компьютерные системы (КС) должны использовать надежные гарантоспособные сервисные Web-услуги Интернет для создания программно-технических систем (ПТС) со свойствами работоспособности, безотказности, долговечности и ремонтопригодности.

По программе ВПК (1950-1992) такие средства были разработаны в СССР для авиации, космоса, энергетики, межконтинентальных баллистических ракет, боевых самолетов, вертолетов и бортовой космической техники. В настоящее время проводятся активные работы по созданию новых видов работоспособного оборудования для оборонной промышленности и медицины страны, учитывающие катастрофические жертвы, пожары, вирусные заболевания и т.п.

Отличный экскурс в историю теории надежности в Советском Союзе и за рубежом содержится в публикации И. Ушакова [1].

1.2 Гарантоспособность вычислений на компьютерах

Представитель технического комитета IFIP «Гарантоспособные вычисления и отказоустойчивость» Дж. К. Лапри (J.C. Laprie) в 1992 году написал книгу «Гарантоспособность. Основные определения и терминология» [2]. В Брюсселе в 1992 году были опубликованы «Критерии оценки безопасности информационных технологий (ITSEC)» [3], принятые в Германии, Франции, Великобритании, Италии и США. Эти критерии постоянно совершенствуются и согласуются с разными странами-участницами IFIP. Так, департамент компьютерных наук университета Вирджинии (Department of Computer Science University of Virginia) занимается повышением живучести информационных систем (ИС) в критических инфраструктурах.

Центр надежных и высокоэффективных вычислений Иллинойского университета (Center for Reliable and High-Performance Computing University of Illinois) модифицировал программный продукт Chameleon до уровня функциональной надежности КС с сетевой структурой. В нем Fault-Tolerance Manager должен обеспечивать адаптацию гарантоспособных средств в разных вычислительных средах (однородной, гетерогенной и кластеризованной).

Департамент компьютерных наук университета в Теннесси (Department of Computer Science University of Tennessee) разрабатал средства для отказоустойчивых сетевых вычислений NetSolve, усовершенствуя клиент-серверную структуру Интернет. В ней модульность продукта дает возможность вести гарантоспособные вычисления на двух уровнях: внутреннем и межсерверном.

Ведущими учреждениями в области гарантоспособности США являются: Лаборатория реактивного движения (JPL) Калифорнийского технологического института; Корпорация электронных систем коммутации Bell System (США, 1953 г.), ныне AT&T; компания IBM с Исследовательским центром им. Дж. Уотсона; Лаборатория им. Ч. Дрейпера Массачусетского технологического института; Международный Стэнфордский научно-исследовательский институт; корпорация Boeing. Вопросами гарантоспособности ТС занимаются: во Франции – Лаборатория автоматики и системного анализа (LAAS) Тулузского национального научно-исследовательского центра и компания Schneider Electric, в Англии – компании SRC и CSR, в Германии – компания Siemens и другие.

В зарубежных работах ([4-6]) приводятся основные результаты исследований в области гарантоспособных и надежных отказоустойчивых КС.

Имеются многочисленные отечественные работы в области обеспечения надежности и безопасности ПТС, выполненные в рамках ВПК в период 1950-1992, а также в современных космических и авиационных системах и для критических инфраструктур, в том числе военного и промышленного назначения ([7-14]).

2. Подходы к обеспечению функциональности надежности ПТС

Наиболее весомые практические результаты по обеспечению надежности и качества получены в ходе проекта с МНИИПА (1975-1987) в рамках ВПК. Под руководством В.В. Липаева ([7-9]) была разработана технология сборочного программирования и обеспечения высокого качества создаваемых ПТС для авиационной, космической, морской

отраслях промышленности и систем специального военного назначения ([15-17]). При этом сформировалась методология обеспечения качественных ПТС:

- тестирование каждого ТС и систем из них на тестовых наборах и сбора сведений об ошибках;
- интеграция, сборка отдельных связанных системных и прикладных ресурсов в систему и проведения интеграционного тестирования;
- оценивание надежности и качества созданных специализированных ПТС для применения в авиационной, морской и космической областях;
- анализ рисков ПТС и отказов аппаратных средств бортовых компьютеров и проведение мероприятий по выпуску высоконадежной и качественной новой продукции.

В критических системах результаты обработки информации и управляющие воздействия определяют работоспособность и качество применения сложных систем в чрезвычайных ситуациях, военными силами, космическими объектами, атомных электростанций и т.п., где происходят аварии и катастрофы вследствие недостаточной безопасности функционирования программных средств. Гибнут спутники и самолеты: Марс 1 (1976), Боинг (2018) и др.

Как правило, ошибки возникают вследствие простых ошибок и дефектов в программах, ПО. Часто это связано с жизнью и здоровьем людей и большими материальными потерями. *Информационная безопасность* определяется защищенностью от неслучайных воздействий. Ее можно рассматривать как преднамеренное ухудшение характеристик функционирования системы, которые специально могут искажать информацию Поэтому требуется защищать программы и данные от злоумышленников с помощью систем защиты криптографии и аутентификации данных и пользователей.

Функциональная безопасность зависит от отказов, влияющих на работоспособность и выполнение основных функций, при выполнении которых могут быть дефекты в аппаратуре, программах и данных. Безопасность функционирования определяется:

- техническими отказами и искажениями информации;
- случайными сбоями и разрушениями информации;
- дефектами и ошибками в программах обработки информации и данных;
- недостатками в средствах обнаружения отказов и восстановления работоспособности систем, программ и данных.

Понятия и характеристики функциональной безопасности систем близки надежности. В надежности учитываются все отказы, а в функциональной безопасности только те, которые приводят к катастрофическим ущербам. Поэтому методы и средства функциональной безопасности базируются на методах определения надежности функционирования комплексов программ, систем и БД.

В настоящее время проблемы обеспечения безотказности, отказоустойчивости, информационной безопасности реализуются в проектах ИСП РАН. Основные направления исследований — борьба с авариями, катастрофами и кибератаками на объекты критической инфраструктуры и коммуникационных средств Интернет и прикладных систем.

Принимая во внимание статистику инцидентов, аварий и катастроф, вызванных отказами КС, рост интенсивности влияния пассивных и активных факторов внешней среды, такие задачи приобретают все большую значимость для банковских и медицинских комплексов, систем электронной коммерции, ГИС и систем мониторинга окружающей среды, приложений распределенного хранения и обработки данных.

На данный момент разработка компьютерных систем и ПТС основывается на сервисных и системных компонентах в рамках моделей SOA/SCA с использованием Web-сервисов и Web-служб Semantic Web и Client-Server Architecture глобальной сети Интернет, как среды взаимодействия разных компонентов Web-систем. Основу технологий КС и ПТС составляет гарантоспособность, надежность, основанная на следующих положениях:

- безотказность (reliability);
- готовность (availability);
- достоверность (high confidence);
- приспособленность к обслуживанию или ремонтопригодность (maintainability);
- информационная безопасность (security) и ее составляющие – конфиденциальность (confidentiality) и целостность (integrity);
- аварийная безопасность (safety).

Теория надежных систем развивается с учетом современных достижений в области элементной базы и технологий моделирования, применения нового математического аппарата теории надежности и конфигурирования сложных ПТС из готовых сетевых и сервисных ресурсов Интернет ([18-21]) и проведения анализа и оценки рисков, устойчивости, прогнозирования неисправностей, отказоустойчивости, защиты, функциональной безопасности, качества систем и др.

3. Моделирование Web-систем из сервисных ресурсов Интернет

3.1 Модели Web-систем

В настоящее время в Интернет среде представлены модели SOA (Service Oriented Architecture) [22] и SCA (Service Component Architecture) [23]], которые образуют используемые для моделирования Web-систем и сайтов из сервисных и готовых reusable компонентов, находящихся в библиотеках и репозиториях Интернет.

SOA – это архитектура из сервисных и системных элементов, которые группируются в среде сервера Интернет с помощью сервисных служб, интерфейсов, входных/выходных данных и портов обмена метаданными, задаваемыми в языке WSDL [24]. Элементы SOA задают описание некоторой функции (Function) с заданным качеством сервиса (Quality service) в IT-стандартах комитета W3C.

SCA – это архитектура из сервисов и компонентов, которые могут быть разработаны различными организациями, как готовые компоненты типа reusable. К ним относятся сетевые сервисы, объекты планирования, доступа к БД и к EJB серверу приложений J2EE. Эта архитектура включает удаленные компоненты повторного использования (КПИ, reuses), которые обмениваются между собой гетерогенными данными из множества общих сетевых Web-сервисов Интернет [25]. Элементы архитектуры могут собираться в систему методом интеграции или конфигурационной сборки требуемых сервисных ресурсов. Данные модели SOA и SCA используются нами при моделировании Web-приложений, Web-систем и сайтов.

3.2 Сервисы, используемые при описании Web-систем

Наряду с компонентным подходом широкое распространение получил сервисный подход. С его помощью реализуется Web-система из готовых системных, функциональных и прикладных компонентов ([17-21], [26]).

Главная идея сервисного подхода Интернет и Semantic Web [27]] состоит в том, чтобы накапливать независимые сервисные компоненты и собирать для разных задач. Функциональные и прикладные сервисы в Интернет – это обычные программные ресурсы, которые реализуют отдельные задачи (в математике, промышленности, экономике, авиации и др.) и накапливаются в Web-библиотеках Интернет. Они обладают способностью к взаимодействию в локальных и/или глобальных сетях Интернет и описываются в языках IDL [28], WSDL [24] и др.

Существуют следующие виды сервисов:

- общие сервисы системных сред (CORBA, DCOM, J2EE, .Net, Appach, JAVA и др.), устанавливают связи с другими сервисами и компонентами через механизмы вызова

RPC, RMI и службы именования, каталогизации и др.;
- сетевые сервисы стандартной модели OSI, API, модели SOA, SCA, которые выполняют обработку пользовательских сервисных ресурсов в сети Интернет;
- готовые программные и сервисные ресурсы (services, artifacts, reuses, assets и др.) пользовательской системы, которые используются как многоразовые КПИ для решения разного рода вычислительных задач, бизнес задач и др. Некоторые из сервисов стали обязательной частью общесистемных средств (VS.Net, IBM, Intel, Linux и др.), а также используются в специальных областях знаний (медицина, биология, генетика, геофизразведка и др.).

3.3 Системные и прикладные сервисы

Применение сервисов производится с помощью клиент-серверной архитектуры. Такая архитектура первоначально была реализована в системе CORBA для управления объектной моделью и ее элементами. Она включает:

- брокер объектных запросов (Object Request Broker — ORB) для взаимодействия клиент-объектов с сервер-объектами на ЯП (Smalltalk, Cobol, Ada-95, Lisp, PL/1, C++, Python, Java, IDLScript и др.);
- общие объектные сервисы (Common Object Services — COS) для управления изменениями, реализациями, транзакциями, подпроцессами и т.п.;
- общие средства обслуживания (Common Facilities — CF) для объединения в различные конфигурации сервисных объектов;
- объектные приложения (Application Objects — AO), над которыми могут производиться операции – открыть, инсталлировать, переместить, поместить, выполнить.

Объект-клиент и объект-сервер обмениваются между собой данными с помощью запросов, каждый из которых обрабатывается брокером ORB на основе описания интерфейсов объектов для клиента, сервера и ядра ORB.

Интерфейс клиента (Client Interface) обеспечивает взаимодействие с объектом-сервера и состоит из трех базовых интерфейсов:

- stub-интерфейс содержит описание внешних параметров и операций в IDL;
- интерфейс динамического вызова (Dynamic Invocation Interface — DII) объекта для выполнения программы клиента при поиске интерфейса в репозитории интерфейсов и программ в репозитории реализаций;
- интерфейс сервисов ORB (ORB Services Interface), содержащий набор сервисных функций, которые клиент запрашивает у сервера через брокер ORB.

Stub-интерфейс обеспечивает взаимосвязь клиента с ORB через операцию удаленного вызова RPC и посылает параметры серверу в запросе. Интерфейс DII обеспечивает доступ объектов и их интерфейсов во время выполнения. В каждом вызове указывается тип объекта, тип запроса и параметры в IDL или WSDL.

3.4 Процесс проектирования Web-систем

Процесс проектирования Web-системы состоит в следующем:

- формирование модели Web-системы;
- определение интерфейсов взаимодействия между сервисными компонентами;
- определение или поиск готовых функциональных сервисов и их размещение в библиотеках Интернет с уникальными именами;
- определение схемы связи сервисных функций или компонентов в ЯП через операторы вызова CALL/RMI/RPC и протоколы связи ISO, в которых задаются входные и выходные параметры;
- использование Web-сервера и Web-клиента для передачи запросов от клиента к серверу

из модели Web-системы.
- верификация и тестирование ресурсов ПТС и проведение метрического анализа рисков, отказов, защиты данных, надежности, информационной безопасности и др.

3.5. Сервисы поддержки Web-систем в Интернет

Для сборки (композиции) сервисов используется инструмент Jopera for Eclipse [29], который выполняет:

- композицию сервисов (типа Agile) и визуальный мониторинг отладки композиций сервисов;
- управление изменением интерфейсов сервиса с помощью сообщений об изменениях в Jopera;
- масштабируемость и автономное исполнение процесса запуска Web-систем с помощью сообщений Jopera.

Jopera содержит набор Eclipse-плагинов для связи различных программных элементов и допускает итеративную композицию сервисов (через маршрутизаторы SOAP и RESTful Web-сервис, Grid-сервисы, Java snipets и др.) после моделирования процессов в сети. Для поиска сервисов по их семантическим описаниям используются Feta Client и Feta Engine [30]. Feta Client – это GUI-плагин системы Интернет Taverna, используемый для описания сервиса, а Feta Engine – для задания Web-сервиса.

Для разработки Web-систем в Semantic Web используются средства:

- RDF стандарта W3C (2004) для описания сетевых, семантических ресурсов и метаданных (данные о данных). Служит каркасом для создания отдельных компонентов семантической паутины.
- RDFS (RDF Schema) — надстройка над RDF, которая позволяет создавать классы и свойства объектов.
- OWL (Web Ontology Language) построен на форматах RDF и RDFS, предназначен для описания онтологий, логики и согласуется с современными сетевыми стандартами.
- SPARQL (Protocol And RDF Query Language) — язык запросов для быстрого доступа к данным RDF для получения необходимой информации из сети.
- RIF (Rule Interchange Format) — формат обмена правилами и др.

WSDL – язык описания входных и выходных данных в запросов Интернет, сервисы которого описываются в языках: WSCI (Web Services Choreography Interface); WSCL (Web Services Conversation Language); BPMN (Business process and model and notation); BPEL (Business Process Execution Language for Web Services) и др.

4. Моделирование устойчивости и безопасности Web-систем

Устойчивость КС определяется их способностью обнаруживать и фиксировать отказы и сбои, часть из которых не было предусмотрено на этапе проектирования прикладных систем. Особенно остро такая задача стоит при построении Web-систем в среде Web- Services и Cloud Computing, которые характеризуются глобальной распределённостью компонентов, гетерогенностью и высокой сложностью. При построении Web-систем предусматривается возможность появления исключительных ситуаций (ошибок, отказов и разного рода сбоев) с помощью методов их обнаружения и анализа.

Создаваемые Web-системы из сервисных и сервисно-компонентных элементов представляют собой композитную структуру из ресурсов Web-сервисов Интернет, обеспечивающую параллельное функционирование компонентов ПТС ([15-17], [25]).

Созданная Web-система должна проверяться на:

- нестабильность функционального состава и структуры Web-системы;
- многовариантность результатов обслуживания и обработки вызовов КПИ средствами

Web-сервиса на корректность и функциональную безопасность при выполнении;

- значение характеристики надежности отдельных компонентов и защиту информации (безотказность, достоверность и др.);
- характеристики оперативности Web-сервисов при обработки запросов и непредсказуемости изменения сетевой задержки для клиента, что обусловлено глобальностью, невысоким качеством сети Интернет;
- сложность применения традиционных средств повышения надежности и оперативности из-за невозможности точного диагностирования возникающих отказов, а также отсутствия достоверной информации о значении надежностных характеристик Web-сервисов и оперативности обслуживания.

Таким образом, сегодняшние устоявшиеся методы и технологии измерения, оценки и прогнозирования характеристик Web-сервисов и Web-систем являются недостаточно мощными в среде Интернет. Это в некоторой степени создает проблемы при прогнозировании характеристик компонентно-интегрируемых сервисно-компонентных Web-систем и синтеза (сборки) таких ПС с требуемыми характеристиками: требуется развитие традиционных методов обеспечения отказоустойчивости и повышения надежности ПТС. Современные Web-системы должны обрабатывать возникающие отказы, сбои и потенциально-опасные события, способствовать повышению качества сервисных ресурсов и обеспечивать функциональную безопасность ПТС в глобальной среде Интернет.

Возможность создавать качественную инфраструктуру Web-сервисной ПТС является критическим условием развития науки, обороны, медицины физических экспериментов и эффективного функционирования предприятий и органов государственного аппарата. Такое видение развития технологий Web-сервисов находится в соответствии с правительственными и международными источниками (программами NEC, NESSI, NECTISE, EPSRC, DTI, EC, FP7 и др.).

Исследование Web-систем проводятся по следующим направлениям:

- адаптация традиционных механизмов обеспечения отказоустойчивости (включая репликацию, восстановление после сбоя, основанное на контрольных точках или коллективной обработки исключений) для сервис-ориентированной архитектуры;
- проведение оценочных испытаний, экспериментальной оценки характеристик надежности;
- анализ Web-сервисов и сервис-ориентированных систем в условиях экстремальных внешних и внутренних воздействий на основе стрессового тестирования и техники «засева» ошибок;
- расширение описания WSDL Web-сервисов за счет включения параметров качества обслуживания QoS и их использования в качестве дополнительных критериев поиска и отбора сервисов.

Концепция сервис-ориентированной архитектуры SOA и сервисно-компонентной архитектуры SCA была предложена для решения проблем обеспечения надежного и безопасного взаимодействия сложных распределенных систем. SOA и SCA обеспечивают построение современных Web-систем на основе слабо связанных программных сервисных модулей (служб) с общедоступными интерфейсами в языке WSDL и специальными механизмами взаимодействия с помощью протокола SOAP (Simple Object Access Protocol) [31]. Описание таких сервисных модулей могут быть обнаружены другими системами в специальных реестрах UDDI (Universal Description, Discovery and Integration), после чего модули могут быть вызваны с помощью SOAP-сообщений в языке XML, передаваемых стандартным интернет-протоколом, таким как HTTP, SMTP, FTP и др.

4.1 Сервисы поддержки надежности Web-систем

При создании и эксплуатации Web-систем в среде облачных вычислений (Cloud Computing) может возникнуть потенциальная уязвимость сервисных ресурсов Интернет к кибер-атакам и информационным вторжениям, что нарушает доступность предоставляемых услуг, целостность и конфиденциальности данных. В связи с этим актуальным является провести исследование механизмов обнаружения исключительных ситуаций, их диагностирование и выявить уязвимости ПО Web-сайтов, систем и Cloud ([12-21], [26-27]).

Главной задачей исследования Web-систем является обеспечение информационной безопасности Web-приложений, Web-систем Cloud Computing, измерение показателей надежности, защита данных Web-систем и облачных Web-систем.

Проверка обработки исключительных ситуаций (exception handling) Web-сервисов в Интернет проводится путем засева ошибок (fault injection) в распределенные Web-системы и облачные системы. Web-услуга воспринимает надежность Web-сервиса через среду коммуникации и функционирования. Распределенная модель взаимодействия, инкапсуляция деталей реализации и отсутствие контроля сервиса Web-услуг значительно сужают номенклатуру воспринимаемых характеристик надежности и сокращают набор доступных показателей для их оценки.

Современная концепция надежности (гарантоспособность) не учитывает вероятностно-временную взаимосвязь между различными вариантами обслуживания, возникновение которой обусловлено асинхронным распределенным характером взаимодействия потребителя и провайдера Web-услуг, а также временными задержками в коммуникационной среде Интернет.

Для достижения высокой надежности и устойчивости Web-систем и сайтов из сервисов требуется знание точных причин и источников возникновения исключительных ситуаций, возникающих во время работы Web-системы и применять наиболее подходящие методы обеспечения отказоустойчивости и восстановления системы после исправления ошибок. К сервис-ориентированным Web-системам можно применять два механизма отказоустойчивости:

- backward error recovery (возврат системы к предыдущему безошибочному состоянию);
- forward error recovery (переход системы в одно из безошибочных состояний).

Это зависит от логики приложения и использования механизмов обработки исключений.

Поскольку возврат к предыдущему безошибочному состоянию не всегда применим к Web-службам из-за того, что большинство из них не хранят своего состояния (т.е. являются stateless), обработка исключений становится наиболее популярным методом обеспечения отказоустойчивости и восстановления после ошибок Web-служб.

В связи с этим необходимо проводить анализ механизмов распространения исключений в Web-системах в среде IBM WebSphere SDK, Apache Axis, Microsoft .Net и др.

Для выполнения такого анализа используется методика засева ошибок/дефектов (fault injection). Засев дефектов является хорошо зарекомендовавшим себя методом оценки надежности и отказоустойчивости вычислительных систем. Несмотря на то, что применение данного метода для исследования распределённых систем в целом является достаточно хорошо изученным, существует определённый дефицит работ в области его применения к Web-сервисам и сервис-ориентированным системам. Важную роль в анализе и оценке надежности Web-сервисов играет тестирование устойчивости Web-систем к некорректным входным параметрам вызова, которые используются при оценке надежности Web-сервисов. В Web-системах могут возникать ошибки (errors), дефекты (faults, bugs), отказы и сбои (failures) в Web-службах.

Отказ/сбой системы диагностируется, когда ошибочный результат распространяется за пределы интерфейса системы. Дефект в КС, например, дефект в ПО, допущенный вследствие

ошибки программиста, может привести к отказу или сбою Web-системы. Обычно различают три группы дефектов КС: дефекты проектирования и разработки, физические неисправности и ошибки взаимодействия (рис.1).

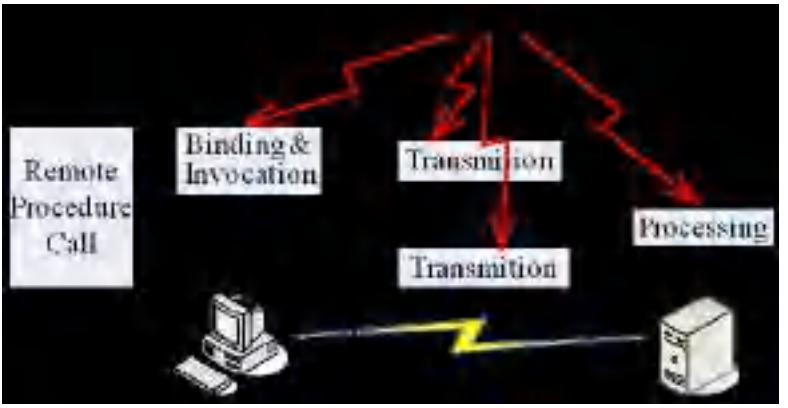

Рис. 1. Потенциальные места возникновения ошибок, отказов и сбоев в сервис-ориентированных Web-системах

Fig. 2. Potential sources of errors, faults, and failures in service-oriented Web-sistems

Опыт показывает, что в Web-сстемах сервисного типа возникают:

- сбои сети и отказы удаленной Web-службы;
- внутренние ошибки и сбои Web-службы;
- ошибки привязки на стороне клиента.

Указанные ошибки/дефекты являются общими и могут проявляться в любой Web-службе во время её работы. К традиционным сетевым отказам и сбоям относятся недоступность службы DNS или же потери и искажения сетевых пакетов. Кроме того, безотказная работа Web-службы зависит от безотказного функционирования ПО, такого как Web-сервер, сервер приложений и система управления базами данных.

Такие сбои могут возникать при принудительном и неожиданном прекращении работы серверов приложений WebSphere, Apache Tomcat и IIS.

Ошибки могут возникать и на стороне клиента при раннем связывании или вызове динамического интерфейса (Dynamic Invocation Interface). Например, «Ошибка в пространстве имен целей», «Ошибка в имени Web-службы» и т.д. происходят из-за изменений параметров вызова и/или несоответствий между WSDL описанием Web-службы и фактическим интерфейсом вызова. Сбои и отказы самих Web-служб могут быть связаны с программными и системными ошибками времени выполнения (run-time errors), которые генерируют пользовательские или системные исключения. Ошибки времени выполнения, такие как «переполнение стека» или «нехватка памяти», приводят к исключениям на уровне системы в целом. Исключение, возникающее вследствие выполнения операции типа «Деление на ноль» также перехватывается и генерируется на системном уровне.

Типичными примерами ошибок времени выполнения приложений являются «Несоответствие типа операнда» или «Выход индекса за пределы массива».

Ошибки связывания сервисов являются, по сути, набором тестов робастности и реализуются на стороне клиента с помощью передачи Web-сервису ошибочных значений параметров вызова. Трассировку и документирование появления исключительных ситуаций можно сделать с помощью стандартного инструментария Java или C#.

В качестве технологий реализации Web-сервисов (конкретной реализации прикладного программного интерфейса для создания Web-служб, а также сервера приложений для их развертывания) предлагается использовать инструменты:

- IBM WSDK + WebSphere;

- Apache Axis + Tomcat;
- Apache Axis + Glassfish;
- Microsoft .Net + IIS.

В качестве интегрированной среды разработки могут быть использованы Eclipse IDE, Netbeans IDE (для Java Web-сервисов), а также Microsoft Visual Studio (для .Net Web-сервисов).

4.2 Исследование уязвимостей NVD и CVE Баз данных

Созданная Web-система из готовых сервисных ресурсов и компонентов может работать с БД Интернет, где размещаются большие данные, к которым идет обращение из других систем и сайтов [17-18]. В БД могут содержаться сведения об уязвимостях к кибератакам, информационным вторжениям, нарушения доступности к сервисным услугам, целостности и конфиденциальности данных. Требуется провести обнаружение аварийных ситуаций, диагностирование выявленных исключительных ситуаций и уязвимостей в Web-системе. Причиной возникновения неадекватных ситуаций могут быть:

- сбои в сети и ошибки удаленной службы при выполнении операторов link;
- сбои и ошибки внутреннего типа при вычислении некоторого сервиса компонента;
- ошибки взаимодействия между клиентом и сервером.

Ошибки на стороне клиента происходят из-за изменений параметров или передаваемых данных и несоответствия типов описания данных в WSDL и/или IDL.

Ошибки сборки (связывания) реализуются на стороне клиента путем передачи ошибочных значений передаваемых данных через параметры. В случае возникновения ошибок передачи данных проводится *трассировка стека исключительных ситуаций* на стороне клиента и Web-системы. Эту трассировку проводит IBM WSDK в процессе взаимодействии с Web-сервисом.

Пример трассировки стека исключений, соответствующего перехвату ошибки («Несовпадение типов операндов») на стороне клиента, использующего реализацию JAX-RPC от Sun Microsystems привеведен на рис.2.

```
java.rmi.ServerException: JAXRPC.TIE.04:
Internal Server Error
   (JAXRPCTIE01: java.lang.NumberFormatException:
   For input string: "578ER")
at com.sun.xml.rpc.client.dii.BasicCall.invoke(BasicCall.java:497)
at ai.c1.xai12.wstest.InvoceWS.invoce(InvoceWS.java:125)
at ai.c1.xai12.wstest.InvoceWS.invoceByVector(InvoceWS.java:75)
at wstest.Main.main(Main.java:42)
```

Рис. 2. Пример трассировки стека исключений
Fig. 2. Exception stack trace example

Информацию об уязвимости получается из общедоступных источников или специализированных БД уязвимости (БДУ), которые предоставляют информацию об уязвимости, возникшей в связи с атаками, нарушениями несанкционированного доступа к защищенным данным БД и др. Поставщиком уязвимостей в БД является общий словарь CVE в виде XML-формата или SGL-дампов (www.osvdl.org). На основе обнаруженных ошибок, исключительных ситуаций и отказов проводится оценка надежности ПО Web-систем, изготовленных из сервисных и компонентных ресурсов Интернет средствами Web-служб.

5. Заключение

В работе представлен подход к построению Web-систем из сервисных и сервисно-компонентных ресурсов, содержащихся в Web-services, в библиотеках и хранилищах

Интернет с обеспечением надежности ПТС в глобальной среде Интернет. Описываются стандартные сервисные ресурсы SOA, SCA, SOAP для создания ПТС для некоторой предметной области знаний. Эти сервисные ресурсы отрабатываются в средах: IBM WSDK + WebSphere, Apache Axis + Tomcat, + Glassfish; Microsoft .Net + IIS.

Приводится классификация возникающих аварийных, исключительных ситуаций и проведения трассировок уязвимостей (атак, запрещенных доступов в БД и др.). Описаны методы моделирования ПТС из готовых ресурсов Интернет и подходы к обеспечению отказоустойчивости, безопасности, защиты и оценки надежности и безопасности сервисных ресурсов и Web-систем с учетом собранных сведений об ошибках и исключительных ситуациях.

Список литературы / References

[1]. Ushakov I. Is Reliability Theory still alive? Reliability: Theory & Applications, vol. 2, no 1, 2007, pp. 6-19.

[2]. Laprie J.C. Dependability: Basic Concepts and Terminology. Springer, 1992, 245 p.

[3]. Information Technology Security Evaluation Criteria (ITSEC): Preliminary Harmonised Criteria. Document COM(90) 314, Version 1.2. Commission of the European Communities, 1991.

[4]. Avizienis A., Laprie J.-C., Randell B., Landwehr C., Dobson I.E Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Trans. on Dependable and Secure Computing, vol. 1, no. 1, 2004, pp. 11-33.

[5]. Chan P.P.W., Lyu M.R., Malek M. Making Services Fault Tolerant. Lecture Notes in Computer Science, vol. 4328, 2006, pp. 43–61.

[6]. Tartanoglu F., Issarny V., Romanovsky A., Levy N. Coordinated Forward Error Recovery for Composite Web Services. In Proc. of the 22nd Symposium on Reliable Distributed Systems (SRDS), 2003, pp. 167-176.

[7]. Липаев В.В. Надежность программного обеспечения. М., СИНТЕГ, 1998, 231 стр. / Lipaev V.V. Reliability of the software. M., SINTEG, 1998 г., 231 p. (in Russian).

[8]. Липаев В.В. Методы обеспечения качества крупномасштабных программных систем. М., СИНТЕГ, 2003 г., 510 стр. / Lipaev V.V. Quality assurance methods for large-scale software systems. M., SINTEG, 2003, 510 p. (in Russian).

[9]. Липаев В.В. Надежность и функциональная безопасность комплексов программ реального времени. Москва, Светлица, 2013 г., 193 стр. / Lipaev V.V. Reliability and functional safety of real-time software complexes. Moscow, Svetlitsa, 2013, 193 p. (in Russian).

[10]. Андон Ф.И., Коваль Г.И. и др. Основы инженерии качества программных систем. Киев, Наукова думка, 2007 г., 670 стр. / Andon F.I., Koval G.I. et al. Fundamentals of quality engineering of software systems. Kiev, Naukova Dumka, 2007, 670 p. (in Russian).

[11]. Горбенко А.В., Засуха С.А. и др. Безопасность ракетно-космической техники и надежность компьютерных систем. Авиационно-космическая техника и технология, no. 1(78), 2011 г., стр. 9–20 / Gorbenko A.V., Zasukha S.A. and other Safety of rocket and space technology and reliability of computer systems. Aerospace engineering and technology, no. 1 (78), 2011, pp. 9–20.

[12]. Лаврищева Е.М., Пакулин Н.В., Рыжов А.Г., Зеленов С.В. Анализ методов оценки надежности оборудования и систем. Практика применения методов. Труды ИСП РАН, том 30, вып.3, 2018 г., стр. 99-120 / Lavrischeva E.M., Pakulin N.V., Ryzhov A.G., Zelenov S.V. Analysis of methods for assessing the reliability of equipment and systems. Practice of methods. Trudy ISP RAN/Proc. ISP RAS, том 30, вып. 3, 2018 г., стр. 99-120 (in Russian). DOI: 10.15514/ISPRAS-2018-30(3)-8.

[13]. Lavrischeva E.M., Mutilin V.S., Ryzhov A.G. Designing variability models for software, operating systems and their families. Trudy ISP RAN/Proc. ISP RAS, vol. 29, issue 5, 2017, pp. 93-110. DOI: 10.15514/ISPRAS-2017-29(5)-6.

[14]. Тарасюк О.М., Горбенко А.В. Безопасность и устойчивость Веб- и облачных систем. Практикум. Министерство образования и науки Украины, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», 2017 г., 40 стр. / Tarasyuk O.M., Gorbenko A.V. Security and Resilience of Web and Cloud Systems. Workshop. Ministry of Education and Science of Ukraine, National Aerospace University «Kharkiv Aviation Institute», 2017, 40 p. (in Russian).

[15]. Лаврищева Е.М., Грищенко В.Н. Связь разноязыковых модулей в ОС ЕС ЭВМ. Москва, Финансы и статистика, 1982 г., 137 стр. / Lavrischeva E.M., Grishchenko V.N. Linking multilingual modules in the OS of ES computers. Moscow, Finance and Statistics, 1982, 137 p. (in Russian).
[16]. Лаврищева Е.М., Грищенко В.Н. Сборочное программирование. Киев, Наукова думка, 1991 г., 282 стр. / Lavrischeva E.M., Grishchenko V.N. Assembly programming. Kiev, Naukova Dumka, 1991, 282 p. (in Russian).
[17]. Липаев В.В., Позин Б.А., Штрих А.А. Технология сборочного программирования. М., Радио и связь, 1992 г., 287 стр. / Lipaev V.V., Pozin B.A., Shtrikh A.A. Assembly programming technology. M., Radio and communication, 1992, 287 p. (in Russian).
[18]. Лаврищева Е.М., Петренко А.К. Моделирование семейств программных систем. Труды ИСП РАН, том 28, вып. 6, 2016 г., стр. 49-64 / Lavrischeva K.M., Petrenko A.K. Software Product Lines Modeling. Trudy ISP RAN/Proc. ISP RAS, vol. 28, issue 6, 2016, pp. 49-64 (in Russian). DOI: 10.15514/ISPRAS-2016-28(6)-4.
[19]. Лаврищева Е.М. Рыжов А.Г. Применение теории общих типов данных стандарта ISO/IEC 11404 GDT к Big Data. Евразийский союз ученых, no. 31, 2016 г., стр. 99-110 / Lavrischeva E.M., Ryzhov A.G. application of the theory of general data types standard ISO/IEC 11404 GDT to Big Data. Eurasian Union of Scientists, no. 31, 2016, pp. 99-110 (in Russian).
[20]. Лаврищева Е.М., А.Г. Рыжов. Подход к моделированию систем и сайтов из готовых ресурсов. Труды XX Всероссийской конференции «Научный сервис в сети Интернет», 2018 г., стр. 321-345 / Lavrischeva E.M., Ryzhov A.G. Approach to the modeling of systems and sites from ready resources. In Proc. of the XX All-Russian Conference on Scientific Services on the Internet, 2018, pp. 321-345 (in Russian),
[21]. Лаврищева Е.М., Петренко А.К. Технология сборки интеллектуальных и информационных ресурсов Интернет. Труды XXI Всероссийской конференции «Научный сервис в сети Интернет», 2019 г., стр. 469-488 / Lavrischeva K.M., Petrenko A.K. Technology of Assembly of intellectual and information resources of the Internet, In Proc. of the XXI All-Russian Conference on Scientific Services on the Internet, 2019, pp. 469-488 (in Russian).
[22]. Service-Oriented Architecture Ontology. The Open Group, 2010, 114 p.
[23]. Paik H., Lemos A.L., Barukh M.C., Benatallah B., Natarajan A. Service Component Architecture (SCA). In Web Service Implementation and Composition Techniques. Springer, 2017, pp. 2-3-250.
[24]. Web Services Description Language (WSDL), Version 2.0, Part 1: Core Language. W3C Recommendation, 26 June 2007.
[25]. Боркус В. Web-сервисы: современные стандарты. Аналитический обзор. PCWeek, 2005 г. / Borkus V. Web services: modern standards. Analytical review. PCWeek, 2005 (in Russian),
[26]. Лаврищева Е.М. Software Engineering компьютерных систем. Парадигмы. Технологии. CASE-системы программирования. Киев, Наукова думка, 2013, 280 стр. / Lavrischeva E.M. Software Engineering of Computer Systems. Paradigms. Technology. CASE- programming systems. Kiev, Naukova Dumka, 2013, 280 p. (in Russian).
[27]. Matthews B. Semantic Web Technologies. E-Learning, vol. 6, no. 6, 2005.
[28]. Interface Definition Language. The Object Management Group. URL: https://www.omg.org/spec/IDL, accessed 20.10.2020.
[29]. Jopera for Eclipse. URL: http://www.jopera.ethz.ch, accessed 20.10.2020
[30]. Taverna plugins. URL: http://www.mygrid.org.uk/usermanual1.7/taverna_plugins.html, accessed 20.10.2020
[31]. SOAP Version 1.2, Part 1: Messaging Framework (Second Edition). W3C Recommendation, 27 April 2007.

Информация об авторах / Information about authors

Екатерина Михайловна ЛАВРИЩЕВА – доктор физико-математических наук, профессор, главный научный сотрудник ИСП РАН, профессор МФТИ. Область интересов: надежность и качество, моделирование сложных систем, Web-системы, сборочное программирование.

Ekaterina Mikhailovna LAVRISCHEVA – Doctor of Physical and Mathematical Sciences, Professor, Principal Researcher at ISP RAS, Professor at Moscow Institute of Physics and Technology. Areas of interest: reliability and quality, modeling og complex systems, Web-systems, assembly programming

Сергей Вадимович ЗЕЛЕНОВ – кандидат физико-математических наук, старший научный сотрудник ИСП РАН, доцент кафедры системного программирования ВШЭ. Область интересов: методы проектирования и разработки ответственных систем, формальные методы программной инженерии, методы верификации и валидации, тестирование на основе моделей.

Sergey Vadimovich ZELENOV – Ph.D. in Physics and Mathematics, Senior Researcher at ISP RAS, Associate Professor of System Programming Department at HSE. Areas of interest: design and development methods for critical systems, formal methods of software engineering, verification and validation methods, model-based testing.