

Полученные графические подтверждения подтверждают предположение о том, что изображения групп с близкой энтропией искажают показатели точности контекстного поиска.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Башков Е.А., Шозда Н.С.* Поиск изображений в больших БД с использованием коэффициента корреляции цветовых гистограмм // GraphiCon'2002. – Нижний Новгород. – 2002. – С. 458–460.
2. *Qui G., Palmer R.D.* WISE: A Content-based Web Image Search Engine // Proc. Multimedia Computing and Networking 2001. – San Jose, CA, USA. – January 2001. – vol. 4312, №4312. – P. 150–161.
3. <http://images.altavista.com/>.
4. *Байгарова Н.С., Бухутаб Ю.А., Есмеева Н.Н.* Современная технология содержательного поиска в электронных коллекциях изображений. – Институт прикладной математики им. М. В. Келдыша РАН, <http://artinfo.ru/eva/eva2000M/eva-papers/200008/Baigarova-R.htm>.
5. *Wang J. Z., Li J., Wiederhold G.* SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2001. – vol. 23, №9. – P. 947–963.
6. *Long F., Zhang H., Feng D.D.* Fundamentals of Content-based Image Retrieval // Multimedia Information Retrieval and Management - Technological Fundamentals and Applications. – Springer. – 2002. – 69 p.
7. *Edgar S.J., Holliday J.D., Willett P.* Effectiveness of Retrieval in Similarity Searches of Chemical Databases: A Review of Performance Measures // Journal of Molecular Graphics and Modelling. – 2000. – №18. – P. 343–357.
8. *Schettini R., Ciocca G., Zuffi S.* A Survey of Methods for Colour Image Indexing and Retrieval in Image Databases, <http://www.intelligence.tuc.gr/~petrakis/courses/multimedia/papers/color-survey.pdf>.
9. Коллекция изображений группы исследователей Ванга, <http://wang.ist.psu.edu/~jwang/test1.tar>.
10. Прэтт У. Цифровая обработка изображений. – М.: Мир, 1982. – Кн. 2. – 480 с.
11. *Башков Е.А., Вовк О.Л.* Оценка эффективности нового статистического иерархического агломеративного алгоритма кластеризации для распознавания регионов изображений // Системні дослідження та інформаційні технології. – Інститут прикладного системного аналізу НАН України, Київ. – 2005. – №2. – С. 117–130.

С.В. Бирюков

АВТОМАТИЗАЦИЯ ФУНКЦИОНАЛЬНОГО ТЕСТИРОВАНИЯ WEB-СЕРВИСОВ

В настоящее время все большее распространение получают серверные приложения, реализующие web-сервисы. Под web-сервисом здесь понимается любая программная система, доступная посредством сети и использующая стандартизованную систему XML-сообщений. Это приложение, к которому можно обратиться, используя специальный формальный

протокол обмена сообщениями SOAP (Simple Object Access Protocol) и получить ответ – некие потенциально полезные пользователю данные [1]. Растущая популярность web-сервисов обусловлена рядом преимуществ, среди которых можно выделить следующие:

- взаимодействие программных систем независимо от платформы;
- использование открытых стандартов и протокол, чем достигается простота разработки и отладки;
- взаимодействие программных систем через межсетевой экран.

Для получения доступа к web-сервису разрабатывается клиентское приложение, умеющее получать описание внешних интерфейсов сервиса на языке WSDL (Web Services Description Language) и взаимодействовать с ним посредством XML-сообщений согласно этому описанию.

В данной работе рассматривается функциональное тестирование именно серверной части. В этом случае основной интерес представляет правильность формирования ответа web-сервиса на соответствующий запрос клиента. Корректность поведения тестируемого приложения определяется спецификацией требований к нему.

Традиционный подход к функциональному тестированию web-сервисов заключается в следующем. Тестировщик, используя клиентское приложение, вводит некоторую совокупность данных, соответствующих одному из сценариев использования web-сервиса. Введенные данные трансформируются клиентской программой в XML-запрос и направляются к серверу, который формирует соответствующий ответ и отправляет обратно. Клиентская программа разбирает XML-ответ и отображает результат в доступной для восприятия форме. После чего тестировщик сравнивает выходные данные с заранее предсказанными и устанавливает факт корректности работы сервиса [2].

Такой подход обладает рядом существенных недостатков:

- для тестирования используется разработанное клиентское приложение, которое также может содержать дефекты, вследствие чего невозможно однозначно выделить лишь дефекты реализации web-сервиса;
- качество ручного тестирования зависит от квалификации проводящего его специалиста;
- для проведения повторного тестирования всей системы при каждой модификации части кода требуются колоссальные временные ресурсы.

В качестве альтернативного подхода была разработана и внедрена система автоматизированного тестирования web-сервисов. Было установлено, что легче всего автоматизации поддаются этапы обмена сообщениями и сравнения результата теста с предсказанным.

Для автоматизации этапа обмена сообщениями необходимо использовать универсальное клиентское приложение, умеющее формировать произвольные XML-запросы по WSDL-описанию web-сервиса. Также для исключения одного из представленных выше недостатков традиционного

подхода приложение должно быть само протестировано и готово к эксплуатации.

Этап сравнения результата теста с предсказанным на первый взгляд может быть легко автоматизирован путем обычного текстового сравнения эталонного и полученного XML-ответа. Однако здесь существует проблема, заключающаяся в том, что корректный XML-ответ не всегда идентичен эталонному с точностью до символа. Это связано с присутствием в ответе строк, отображающих текущую дату и время, а также других данных, изменяющихся от запроса к запросу (например, счетчик числа обращений к сервису). Тем более недопустимо побайтовое текстовое сравнение в случае негативных тестов, когда к web-сервису сознательно посылается запрос, генерирующий ошибку сервиса. В этом случае ответ содержит стек вызовов функций сервиса с указанием номеров строк, в которых была сгенерирована ошибка. Очевидно, что эти номера изменяются при каждой модификации кода.

Для решения этой проблемы было предложено сравнивать только необходимую часть ответа, выбирая из XML нужные данные посредством XPath – языка запросов к XML [3].

Исходя из возможности реализации представленных идей, был проанализирован анализ наличия на рынке подходящих инструментальных средств.

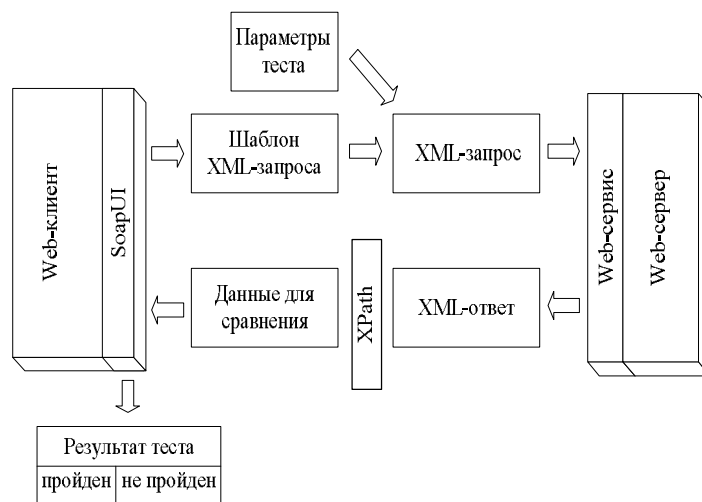


Рис. Общая схема автоматизированного теста

В результате было выбрано свободно распространяемое программное средство SoapUI [4].

В соответствии со спецификацией требований к web-сервису была построена его формальная модель в виде графа потока управления. Подготовлен набор позитивных тестов, обеспечивающий покрытие всех связей

графа, а также негативных тестов, устанавливающих факт отсутствия ненужных связей [5].

С учетом особенностей выбранного инструментального средства была разработана общая схема выполнения автоматизированного тестового сценария (рис.). SoapUI, получая WSDL-описание сервиса, формирует множество шаблонов всех возможных XML-запросов. Каждый тестовый сценарий представляет собой перестановку некоторого подмножества множества шаблонов. Тестовые сценарии формируются тестировщиком единойжды на этапе проектирования тестов и хранятся в файле проекта SoapUI.

При выполнении тестового сценария SoapUI, анализируя идентификатор теста, автоматически извлекает из базы данных параметров необходимую информацию и подставляет ее в шаблон XML-запроса, формируя тем самым окончательный запрос. Затем запрос направляется непосредственно к web-сервису, который, согласно своей функциональности, производит преобразования входных данных в выходные и направляет ответ обратно в виде XML-сообщения. Получив ответ, SoapUI выполняет XPath-запрос к пришедшему XML-сообщению, выбирая необходимые для сравнения с эталоном данные. Выбранные данные и эталон проходят процедуру текстового сравнения, после чего объявляется вердикт в форме «Тест пройден / Тест не пройден».

В данной работе представлен подход к автоматизации функционального тестирования web-сервисов с использованием программного средства SoapUI. Подход позволил автоматизировать наиболее трудоемкие этапы тестирования – обмен сообщениями и сравнение результатов работы web-сервиса с предсказанными. Он оказался особенно эффективен при организации регрессионного тестирования после каждой итерации исправления ранее обнаруженных дефектов. Среди дальнейших направлений развития данного подхода наиболее интересным представляется автоматизация этапа генерации тестовых сценариев на основе формализации спецификации требований.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Долотин И.* Веб-сервисы. Основы. Цикл статей сайта "Технологии веб-сервисов". Статья первая. // http://www.ubs.ru/ws/ws_basics1.html.
2. *Стотлемейер Д.* Тестирование Web-приложений: Средства и методы для автоматизированного и ручного тестирования программного обеспечения Web-сайтов. – М.: Кудиц-Образ, 2003. – 240с.
3. *Портье Б.* Начало работы с XPath. Изучение основ.// http://www.ibm.com/developerworks/ru/edu/x-xpath/?S_TACT=105AGX99&S_CMP=EDU.
4. SoapUI Feature Overview // <http://www.soapui.org/features.html>.
5. *Бейзер Б.* Тестирование черного ящика. Технология функционального тестирования программного обеспечения. – СПб.: Питер, 2004. – 318с.