

Инфраструктура тестирования веб-сервисов на базе технологии TTCN-3 и платформы .NET

П.Н. Яковенко, А.В. Сапожников
yak@ispras.ru alexsa.msu@gmail.com

Аннотация. Работа посвящена вопросам тестирования веб-сервисов с использованием технологии TTCN-3. Рассматривается отображение языка WSDL в TTCN-3, основанное на синхронном (процедурном) взаимодействии тестового сценария с веб-сервисом. Предлагается подход реализации универсального тестового адаптера средствами рефлексии, предоставляемыми платформой .NET, и средой поддержки времени выполнения языка TTCN-3. Результатом работы явилась разработка прототипа системы автоматизации тестирования веб-сервисов.

1. Введение

Веб-сервисы – технология построения распределенных систем, базирующаяся на языке XML и открытых коммуникационных протоколах. Идеи, лежащие в основе веб-сервисов, восходят к технологии удаленного вызова процедур (RPC), разработанной компанией SUN Microsystems в рамках проекта сетевой файловой системы. В дальнейшем эти идеи получили развитие в технологиях CORBA, DCOM и др. Принципиальное отличие веб-сервисов от альтернативных подходов организации взаимодействия сетевых приложений состоит в том, что технология веб-сервисов ориентирована на слабосвязанную интеграцию разнородных программных систем.

В основу понятия «веб-сервис» заложены несколько базовых принципов [1], которые в совокупности отличают веб-сервис от других категорий серверного программного обеспечения. Среди них следует отметить слабую связанность веб-сервиса с клиентскими приложениями, самодокументируемость веб-сервиса посредством его WSDL [2] спецификации, сетевое взаимодействие с использованием общепринятых, открытых протоколов.

Слабая связанность систем предполагает минимизацию зависимостей между логикой работы взаимодействующих сторон. Чем меньше клиентская сторона выдвигает предположений о стабильности программного интерфейса, предоставляемого сервером, о его доступности во время работы клиента, о неизменности структуры и содержимого сообщений, передаваемых между

клиентом и сервером, тем в большей степени система является слабосвязанной. Технология веб-сервисов предоставляет ряд существенных возможностей для организации слабосвязанной архитектуры, впрочем степень связанности каждой конкретной реализации во многом определяется допущениями, принятыми на вооружение разработчиками сервиса.

Веб-сервис представляет собой программную систему, расположенную на сетевом компьютере и идентифицируемую строкой URI [3] (единообразный идентификатор ресурса) уникальным образом. Полная спецификация интерфейсов веб-сервиса и способов доступа к нему описывается на языке WSDL. Этот «паспорт» веб-сервиса представляет собой XML документ, в котором можно выделить две части. Первая, абстрактная (платформонезависимая), описывает в структурированном виде сигнатуры операций, предоставляемых веб-сервисом, и типы данных формальных параметров и исключений. Вторая часть WSDL документа, содержит платформо-зависимые привязки (binding) типов данных и операций к конкретным форматам кодирования и коммуникационным протоколам, и другую специфическую информацию (например, номер порта), необходимую программе для получения доступа к веб-сервису и взаимодействия с ним.

Веб-сервисы, как и любые другие программные системы, необходимо тестировать. Растущее распространение веб-сервисов и их использование в промышленных приложениях, обуславливает необходимость создания эффективных методик тестирования. В этой работе предлагается подход к автоматизации тестирования веб-сервисов, построенный на базе технологии TTCN-3 [9,10]. Мы считаем обоснованным применение TTCN-3 для тестирования веб-сервисов в силу того, что оба языка WSDL и TTCN-3 следуют единому подходу разделения спецификации (веб-сервиса и тестового сценария) на зависящие и независящие от платформы уровни. Это позволяет разрабатывать платформо-независимые TTCN-3 тесты по абстрактной части WSDL спецификации веб-сервиса и учитывать платформо-зависимые аспекты доступа к сервису (привязки) на уровне адаптации тестов [12,13].

Преимущества тестирования веб-сервисов с использованием технологии TTCN-3 обсуждаются также в работе [4]. В [5] авторы исследуют вопросы тестирования веб-сервисов, использующих протокол SOAP [8], и предлагают правила импорта XML конструкций для языка TTCN-3, которые могут быть использованы для автоматической генерации TTCN-3 деклараций по XML схеме. Работы [6,7] независимо предлагают два варианта правил преобразования WSDL конструкций в TTCN-3 и механизм взаимодействия тестового сценария с веб-сервисом на базе обмена сообщениями. Однако архитектурные подходы, принятые в этих работах, усложняют разработку и сопровождение тестовых сценариев: в [6] сгенерированные TTCN-3 декларации размещаются в одном модуле, а в [7] авторы выносят часть служебной информации, относящейся к сетевому взаимодействию, на уровень TTCN-3 кода.

В данной работе мы рассматриваем возможность построения системы автоматизации тестирования веб-сервисов на базе технологии TTCN-3, в которой все платформу-зависимые аспекты взаимодействия тестовой системы с веб-сервисом выносятся на уровень адаптации тестов, а обработка привязок веб-сервиса делегируется сторонним средствам. Статья организована следующим образом. В разделе 2 мы предлагаем отображение языка WSDL в язык TTCN-3, предусматривающее синхронное (процедурное) обращение к операциям веб-сервиса. В разделе 3 описывается схема работы тестового адаптера, использующего для взаимодействия с веб-сервисом медиаторы, генерируемые средствами платформы .NET. В разделе 4 изложена архитектура предлагаемой инфраструктуры тестирования. В разделе 5 подводятся итоги работы.

2. Отображение языка WSDL в TTCN-3

Тестирование веб-сервиса при помощи технологии TTCN-3 предполагает наличие трех составляющих: *статической, динамической и времени выполнения*. Статическая часть состоит из набора TTCN-3 модулей, отражающих интерфейс веб-сервиса, и содержит декларации типов данных, сигнатуры операций, определения портов и пр. Совокупность тестовых сценариев образует динамическую составляющую. И, наконец, составляющая времени выполнения представляет собой реализацию интерфейсов TRI [14] и TCI [15], обеспечивающую коммуникационное взаимодействие тестовых сценариев с веб-сервисом. Наличие формального отображения конструкций языка WSDL в конструкции TTCN-3 дает возможность автоматизировать построение первой (статической) и третьей (времени выполнения) составляющих по WSDL документу. К сожалению, WSDL не предоставляет достаточно информации для генерации тестовых сценариев. Сценарии должны либо создаваться тестировщиком вручную на языке TTCN-3, либо генерироваться на базе какой-либо графической нотации (например, диаграмм последовательностей языка UML) или формальной спецификации поведения веб-сервиса.

Одной из сложностей, с которой мы столкнулись при разработке отображения, является отсутствие возможности в языке TTCN-3 определять пользовательские области видимости. Как показал анализ WSDL спецификаций нескольких промышленных веб-сервисов, прямое отображение WSDL документа в один TTCN-3 модуль неминуемо приведет к конфликтам имен. Для разрешения таких конфликтов, необходимо либо модифицировать имена объектов в процессе трансляции, либо распределять одноименные TTCN-3 декларации по разным единицам трансляции (модулям). Предлагаемое в этой работе отображение комбинирует оба варианта, отдавая приоритет модульной структуре, в частности отдельный модуль генерируется для каждой целевой области видимости языка XML, каждой привязки веб-сервиса и каждого элемента *service* WSDL документа.

WSDL спецификация заимствует правила определения типов из языка XML Schema. Типы данных, определяемые для формальных параметров веб-сервиса, возвращаемых значений и исключений описываются внутри элемента *<types>* WSDL документа, который в свою очередь может определять и импортировать несколько XML схем. В целом типы, задаваемые XML схемой, преобразуются в TTCN-3 декларации согласно правилам, определенным в девятой части TTCN-3 стандарта [11]. Основное отличие применяемого нами отображения заключается в другом распределении XML схем по TTCN-3 модулям, при котором отдельный TTCN-3 модуль создается для каждого целевого пространства имен (*targetNamespace*). Это позволяет сохранить именование объектов между WSDL и TTCN-3 кодом и избежать добавления префиксов (пространств имен XML) к TTCN-3 идентификаторам.

Элемент WSDL документа	Тэг WSDL	Декларация TTCN-3
Тип данных	<i><types></i>	Тип данных TTCN-3.
Раздел сообщения	<i><part></i>	Формальный параметр или возвращаемое значение сигнатуры.
Сообщение	<i><message></i>	Раскрывается в набор формальных параметров сигнатуры.
Исключение	<i><fault></i>	Исключение в определении сигнатуры.
Операция	<i><operation></i>	Сигнатура.
Тип порта	<i><portType></i>	Порт процедурного типа.
Привязка	<i><binding></i>	Отдельный модуль. Содержит сигнатуры и типы портов.
Сервис	<i><service></i>	Компонентный тип. Размещается в отдельном модуле.
Порт	<i><port></i>	Порт. Размещается внутри определения компонентного типа.

Таблица 1. Соответствие WSDL элементов и TTCN-3 деклараций.

Каждое целевое пространство имен отображается в отдельный TTCN-3 модуль с именем идентичным префиксу целевого пространства. Это относится как к схемам, определенным непосредственно в WSDL документе, так и ко всем импортированным схемам. Для каждой привязки (элемент *<binding>*) формируется отдельный TTCN-3 модуль, в который транслируются декларации сообщений (*<message>*), их разделы (*<part>*), операции (*<operation>*), исключения (*<fault>*) и типы портов (*<portType>*). Наконец, каждый сервис (элемент *<service>*) вместе с определениями портов (*<port>*) также транслируется в отдельный модуль. Типы данных (элемент *<types>*) обрабатываются согласно стандартному отображению языка XML Schema в TTCN-3. Правила трансляции остальных элементов WSDL документа

приведены в таблице 1. Привязки веб-сервиса к конкретным протоколам передачи данных (например, SOAP и HTTP) игнорируются в процессе трансляции WSDL документа; их обработка рассмотрена в следующем разделе.

В таблице 2 приведен пример трансляции WSDL спецификации веб-сервиса. Типы *Request* и *Response* размещаются в модуле *tns* в соответствии с префиксом целевого пространства имен. Привязка (элемент `<binding>`) *Jobs* порождает отдельный модуль *bindingJobs* (значение атрибута *name* с добавлением префикса *binding*). В этом модуле находятся определения сигнатуры *Get*, которая соответствует одноименной операции веб-сервиса, и порт процедурного типа *portJobs*. Раздел *p* входного сообщения *In* отображается во входной параметр сигнатуры, а соответствующий раздел сообщения *Out* – в возвращаемое значение сигнатуры. Определение сервиса (элемент `<service>`) порождает отдельный модуль с именем *serviceJobsWs*. Порт (элемент `<port>`) *p1* соответствует компонентному типу языка TTCN-3. В эту компоненту помещается TTCN-3 порт (точка доступа к компоненте) с фиксированным именем *P*. Тип этого порта определяется исходя из значения атрибута *binding*.

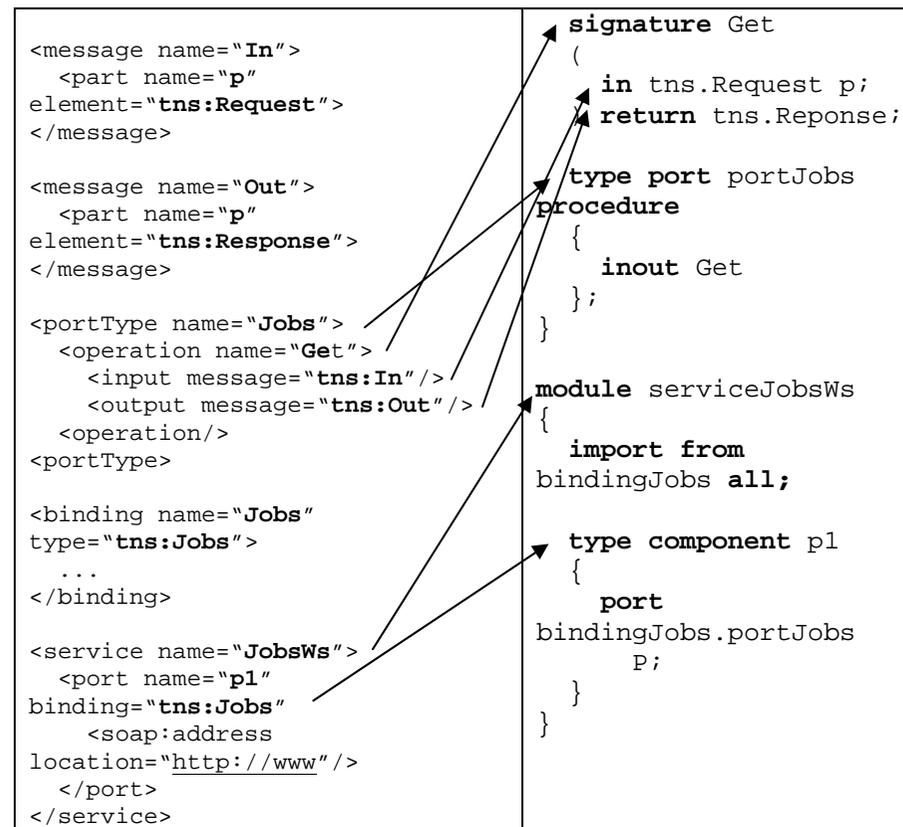
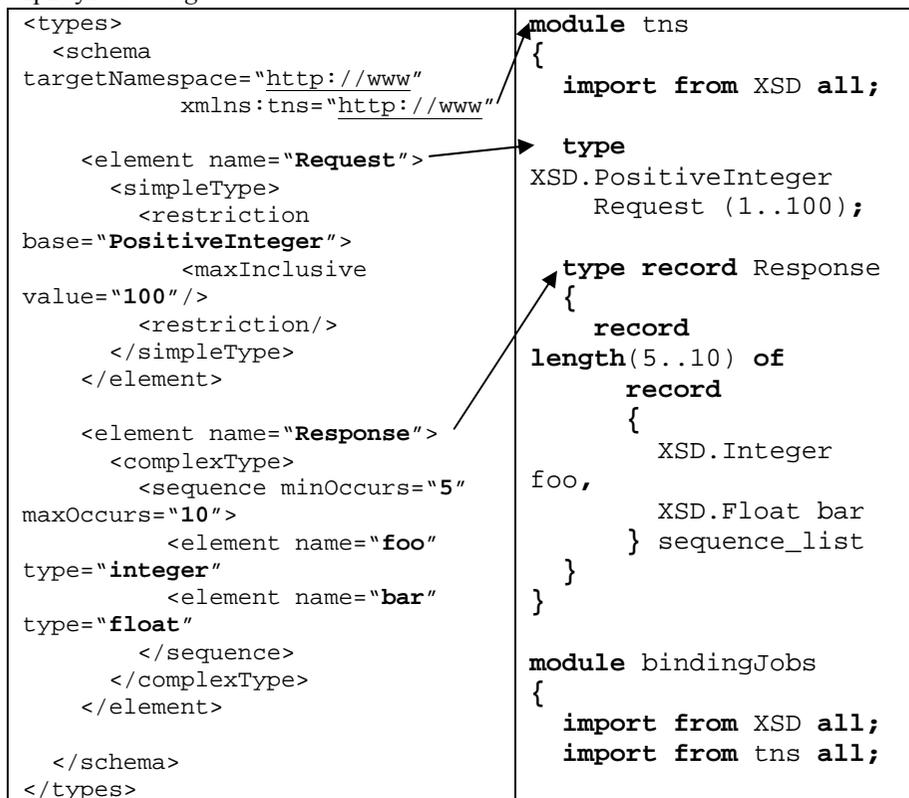


Таблица 2. Трансляция WSDL документа.

3. Поддержка взаимодействия тестового сценария с веб-сервисом

Язык TTCN-3 предлагает гибкий способ обеспечения взаимодействия TTCN-3 тестового сценария с внешней средой и, в частности, с тестируемой системой. Поведение многих операций языка определяется тем, как тестирующий реализовал стандартные TTCN-3 интерфейсы времени выполнения TRI [14] и TCI [15]. Большая часть операций этих интерфейсов имеет отношение скорее к характеру, чем к предмету тестирования, и допускает наличие универсальной реализации, применимой для тестирования систем различного класса. Такие типовые реализации, как правило, поставляются в составе средств разработки языка TTCN-3 и интенсивно переиспользуются в разных проектах. Например, использование потоков (thread) является типичным подходом к реализации механизма параллельных компонент в тестовой

системе, но в случае необходимости (например, при нагрузочном тестировании) тестировщик может предоставить собственную реализацию, распределяющую параллельные тестовые компоненты по отдельным машинам сети. При этом тестовый сценарий не потребует никаких изменений.

Среди всех операций интерфейсов TRI и TCI отдельное место занимает группа операций, ответственная за взаимодействие с тестируемой системой. Реализация этих операций (будем называть ее тестовым *адаптером*) зависит от специфики конкретной системы, и, как правило, возлагается на разработчика тестов. Часто это вызывает серьезные затруднения у тестировщика, т.к. ставит его перед необходимостью изучать на глубоком уровне сетевые протоколы и форматы кодирования данных. Поэтому важным является предоставление инфраструктурой тестирования полноценной поддержки времени выполнения, снимающей с тестировщика обязанности (в т.ч. реализовывать адаптер), не имеющие непосредственного отношения к разработке тестов.

Адаптер тестирования веб-сервисов решает задачи:

- *кодирования данных* - преобразования объектов TTCN-3 системы времени выполнения в SOAP (или другие) пакеты и обратно;
- *передачи данных* - обеспечения сетевого взаимодействия тестового сценария с веб-сервисом.

Система поддержки времени выполнения языка TTCN-3 оперирует объектами абстрактного типа данных *TciValue*. Такой объект может хранить произвольное значение одного из типов данных, определенных в тестовом сценарии. Адаптер может самостоятельно выполнять кодирование *TciValue* объектов и осуществлять сетевое взаимодействие с веб-сервисом, но тогда эти операции должны быть реализованы точно в соответствии с привязками, определенными в WSDL документе. Альтернативным подходом является решение этих задач при помощи существующих средств автоматизации взаимодействия с веб-сервисами (CAB), например, *gSOAP*, *Apache Axis*, утилиты *wsdl* пакета разработчика (SDK) для платформы Win32 и др. При таком подходе адаптер может не учитывать правила привязок веб-сервиса, определенные в WSDL документе, делегируя их обработке используемому CAB.

CAB генерируют по WSDL документу набор функций (классов) - медиаторов, обеспечивающих прозрачное взаимодействие с веб-сервисом. Обращение к медиатору в клиентской программе неявно выполняет удаленный вызов процедуры (операции) веб-сервиса, при этом медиатор берет на себя обслуживание всех платформо-зависимых аспектов взаимодействия, в частности, кодирования SOAP пакетов. Выбор CAB ограничен используемым средством разработки TTCN-3 и определяется тем, какие традиционные языки программирования (C, C#, Java и т.д.) допускаются при реализации TRI/TCI интерфейсов. Как правило, это Си и C++, обладающие ограниченными

возможностями динамического манипулирования объектами во время выполнения тестов. Вместе с тем динамические возможности современных языков программирования, а именно средства рефлексии, позволяют применять гибкие подходы к интеграции тестового адаптера с генерируемыми медиаторами.

Платформа .NET поддерживает рефлексии и предоставляет набор классов в пространстве имен System.Reflection, дающих возможность исследовать элементы программы во время ее выполнения и динамически манипулировать ими. В свою очередь, TCI интерфейс языка TTCN-3 также предоставляет операции, необходимые для динамического манипулирования объектами TciValue. Знание соответствия между элементами WSDL документа и сгенерированными медиаторами с одной стороны и между элементами WSDL документа и сгенерированными TTCN-3 модулями с другой позволяет реализовать на базе средств рефлексии универсальный адаптер, пригодный для обеспечения взаимодействия тестового сценария с произвольным веб-сервисом, т.е. адаптер, не зависящий от WSDL спецификации веб-сервиса. Этот подход мы применили в инфраструктуре тестирования, рассматриваемой в этой работе.

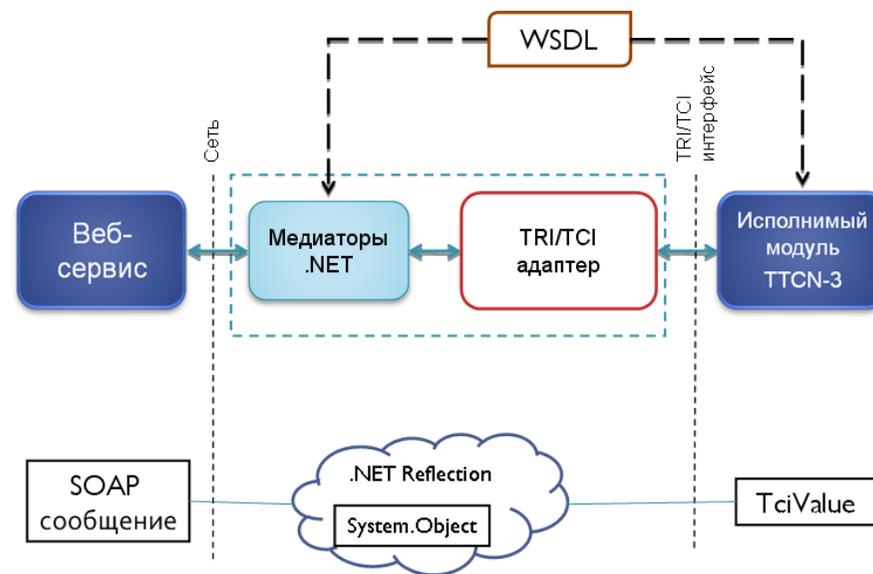


Рис. 1. Схема взаимодействия тестового сценария с веб-сервисом.

Мы использовали утилиту *wsdl.exe* из пакета разработчиков (SDK) платформы Win32 для генерации медиаторов. В качестве целевого языка был выбран C++/CLI, поскольку он обеспечивает удобную интеграцию управляемого кода

медиаторов с остальной частью кода тестовой системы, реализованной на языке Си. Си – единственный язык, поддерживаемый средой Telelogic Tester для генерируемого кода и реализации TRI/TCI интерфейсов. Схема взаимодействия тестового сценария с веб-сервисом приведена на рисунке 1.

При обращении тестового сценария к операции *call*, что соответствует вызову операции веб-сервиса, система поддержки выполнения последовательно кодирует все фактические параметры, вызывая операцию *tcEncode* TCI интерфейса для каждого из них. Эта операция реализуется в адаптере. Адаптер запрашивает тип значения, хранящегося в объекте *TciValue* (посредством стандартной операции *tcGetType*), определяет имя типа соответствующего .NET объекта в медиаторах (по имени типа *TciValue* объекта), конструирует .NET объект средствами рефлексии и заполняет его значениями соответственно содержимому *TciValue* объекта. Порожденный .NET объект сериализуется, и полученный в результате битовый массив возвращается системе поддержки выполнения как результат операции кодирования. После завершения кодирования всех параметров система поддержки выполнения вызывает операцию *triCall* TRI интерфейса, также реализованную в адаптере. Адаптер средствами рефлексии запрашивает у платформы .NET объект-метод (класса *MethodBase*), соответствующий вызываемой операции и выполняет его, передавая ранее подготовленные фактические параметры. Возвращенные значения обрабатываются в обратном порядке.

4. TTCN-3 инфраструктура тестирования

Предлагаемая в этой работе инфраструктура тестирования состоит из следующих взаимодействующих компонент:

- среда разработки TTCN-3;
- среда разработки для традиционного языка программирования, обеспечивающая сборку проекта;
- программа трансляции WSDL документа в набор TTCN-3 модулей;
- средство автоматизации взаимодействия с веб-сервисом, генерирующее медиаторы;
- универсальный адаптер, реализующий TRI/TCI интерфейсы на базе сгенерированных медиаторов с использованием средств рефлексии.

Мы реализовали прототип этой инфраструктуры на базе TTCN-3 среды Telelogic Tester, Microsoft Visual Studio и платформы .NET. Процесс получения исполняемых тестов представлен на Рис. 2.

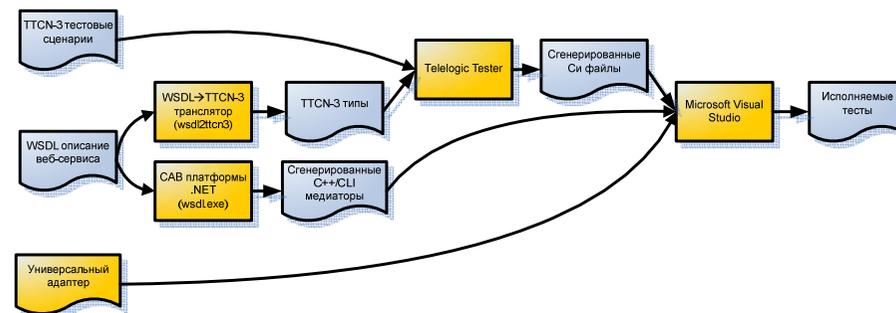


Рис. 2. Формирование исполняемых тестов.

WSDL документ, специфицирующий интерфейс тестируемого веб-сервиса, обрабатывается утилитой *wsdl2ttn3*. Эта утилита реализует предложенное в данной работе отображение языка TTCN-3 в язык WSDL и генерирует набор TTCN-3 модулей, содержащих декларации типов и операций, которые могут быть использованы в тестовых сценариях. Сами сценарии могут разрабатываться разными способами, однако этот вопрос выходит за рамки данной работы. Готовый набор TTCN-3 модулей, содержащих в том числе тестовые сценарии, обрабатывается средой Telelogic Tester, которая, в свою очередь, формирует набор Си файлов и сборочный файл для компилятора среды Microsoft Visual Studio.

Исходный WSDL документ также подается на вход утилите *wsdl.exe* из пакета разработчиков (SDK) платформы Win32, которая генерирует медиаторы для взаимодействия с веб-сервисом на языке C++/CLI. Си файлы, сгенерированные средой Telelogic Tester, медиаторы и универсальный адаптер, обеспечивающий интеграцию медиаторов в TTCN-3 систему поддержки времени выполнения, компилируются средствами среды Microsoft Visual Studio. В результате формируется исполняемый модуль с тестами.

5. Заключение

В этой работе мы предложили подход к построению TTCN-3 системы тестирования веб-сервисов, в котором все платформо-зависимые аспекты взаимодействия тестового сценария с веб-сервисом вынесены на уровень адаптации тестов. Обслуживание привязок веб-сервиса делегируется медиаторам, генерируемым по WSDL спецификации его интерфейса существующими средствами, в частности утилитой *wsdl.exe* из пакета разработчика (SDK) для платформы Win32.

Средства рефлексии, предоставляемые некоторыми языками программирования (в частности языками платформы .NET), а также гибкость интерфейсов времени выполнения языка TTCN-3, используемых для адаптации тестов, позволяют реализовать на базе медиаторов универсальный

тестовый адаптер. Такой адаптер может быть использован для тестирования произвольного веб-сервиса, связь с которым осуществляется средствами платформы .NET, без внесения изменений в его код.

Мы также предложили отображение языка WSDL в TTCN-3 основанное на синхронной (процедурной) парадигме взаимодействия тестового сценария с тестируемой системой. Мы считаем, что процедурный подход в большей степени отражает природу взаимодействия клиентских программ с веб-сервисом, по сравнению с взаимодействием на основе обмена сообщениями. Вместе с тем, язык TTCN-3 позволяет осуществлять асинхронный вызов удаленных процедур, и такая возможность может быть использована для тех случаев, когда цель теста состоит в проверке корректности работы операции веб-сервиса, для которой существенным является неблокирующий характер вызова.

Прототип описанной в этой работе инфраструктуры тестирования был реализован в среде Telelogic Tester и показал свою работоспособность при тестировании публично доступного веб-сервиса компании Amazon Associates, предоставляющего интерфейс для работы с одноименным интернет-магазином.

Литература

- [1] Marks, E. A., and Werrell, M. J. Executive's Guide to Web Services. San Francisco: John Wiley & Sons, 2003. ISBN:0471266523, 742 p.
- [2] Web Services Description Language (WSDL) 1.1 W3C Note 15 March 2001 (<http://www.w3.org/TR/wsdl>)
- [3] RFC 3986. Uniform Resource Identifier (URI): Generic Syntax, January 2005 (<http://tools.ietf.org/html/rfc3986>).
- [4] Xiong, P., Probert, R. L. and Stepien, B. An Efficient Formal Testing Approach for Web Service with TTCN-3. In Proceedings of the 13th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2005), Split (Croatia), 2005
- [5] Schieferdecker, I., and Stepien, B. Automated Testing of XML/SOAP based Web Services. In Proceedings of the 13th Fachkonferenz der Gesellschaft für Informatik (GI) Fachgruppe Kommunikation in verteilten Systemen (KIVS), Leipzig (Germany), 2003
- [6] Schieferdecker, I., Vega, D., and Rentá, C. Import of WSDL Definitions in TTCN-3 Targeting Testing of Web Services. In Proceedings of the 9th International Conference on Integrated Design and Process Technology (IDPT 2006), San Diego (USA), 2006
- [7] Troschütz, S. Web Service Test Framework with TTCN-3. Institute for Computer Science, University of Gottingen (Germany), 2007
- [8] SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) W3C Recommendation 27 April 2007 (<http://www.w3.org/TR/soap12-part1>)
- [9] ETSI. Methods for Testing and Specification (MTS). The Testing and Test Control Notation edition 3. Part 1 TTCN-3 Core Language. ETSI ES 201 873-1, V3.4.1, 2008
- [10] Willcock, C., Deiß, T., Tobies, S., Schulz, S., Keil, S., and Engler, F. An Introduction to TTCN-3. Wiley, 2005
- [11] ETSI. Methods for Testing and Specification (MTS). The Testing and Test Control Notation edition 3. Part 9 TTCN-3: Using XML schema with TTCN-3. ETSI ES 201 873-9, V3.3.1, 2008

- [12] Schulz, S., and Vassiliou-Gioles, T. Implementation of TTCN-3 Test Systems using the TRI. In Proceedings of 14th International Conference on Testing Communicating Systems. Berlin. Germany. 2002
- [13] Schieferdecker, I., and Vassiliou-Gioles, T. Realizing distributed TTCN-3 test systems with TCI. In Proceedings of 15th International Conference on Testing Communicating Systems. Cannes. France. 2003
- [14] ETSI. Methods for Testing and Specification (MTS). The Testing and Test Control Notation edition 3. Part 5 TTCN-3 Runtime Interface (TRI). ETSI ES 201 873-5, V3.3.1, 2008
- [15] ETSI. Methods for Testing and Specification (MTS). The Testing and Test Control Notation edition 3. Part 6 TTCN-3 Control Interface (TCI). ETSI ES 201 873-6, V3.4.1, 2008