

УДК 004.42

Модель информационной среды университета на основе единого репозитория сервисов

*Козлов М.В., студент
кафедры «Программное обеспечение ЭВМ и информационные технологии»
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана*

*Научный руководитель: Остриков С.П.,
начальник Управление информатизации - ВЦ
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
bauman@bmstu.ru*

ВВЕДЕНИЕ

Сервис-ориентированная архитектура (Service-oriented architecture, SOA) [1], в настоящее время, является основной парадигмой для построения единой информационной среды Университета. Это обусловлено тем, что данная архитектура не привязана к какой-то определённой технологии, что позволяет обеспечить независимость в разработке отдельных программных продуктов. С другой стороны SOA накладывает определенные требования к реализации механизма взаимодействия распределенных приложений через веб-службы. Данная статья предлагает модель информационной среды Университета, на основе единого репозитория сервисов.

За последние 10 лет сложилось представление о том, какие основные информационные потребности существуют в вузе. Написаны приложения, которые представлены в Университете в виде сервисов. Наиболее востребованными из них являются:

- Сервис контингента
- Текущая успеваемость

При отсутствии согласованного механизма разработки и публикации веб-сервисов, перед потенциальными потребителями остро стоит проблема поиска мест размещения сервисов, определения перечня предоставляемых ими услуг, определения типов получаемых данных и формата отправляемых сообщений. В таких условиях, веб-сервисы могут дублироваться. Сервис сменивший адрес идентифицируется клиентом, как переставшее работать приложение. Реализация возможностей взаимодействия сервисов усложняется отсутствием единых интерфейсов, глоссария и технической документации.

Задача данной статьи – сформировать единые подходы по интеграции веб-сервисов в рамках информационной среды Университета.

Итак, любой разработчик информационной системы Университета, желая использовать информацию из других информационных систем, опубликованную в виде веб-сервисов, должен иметь возможность обратиться к единой службе, которая содержит исчерпывающую информацию обо всех существующих в Университете веб-сервисах. В задачи данной службы должно входить управление всеми веб-сервисами ВУЗа. Управление подразумевает под собой регистрацию новых веб-сервисов, ведение справочника существующих веб-сервисов, удаление устаревших сервисов из реестра, оповещение подписавшихся обо всех изменениях в конфигурации веб-сервисов. Единый справочник должен содержать информацию о полном перечне предоставляемых веб-сервисами услуг. Поскольку для успешного взаимодействия с веб-службой необходимо знать, какую информацию она предоставляет, в каком формате, каким образом формировать запросы, на какой адрес и порт, по какому протоколу обращаться и многое другое.

Иными словами, из единого справочника можно будет получить достоверную информацию о любом из зарегистрированных в ВУЗе сервисе.

Следующим шагом является обращение по указанному адресу размещения веб-сервиса. Результатом обработки запроса будут являться данные, используемые в дальнейшем приложением их запросившем.

Таким образом формируется сервис-ориентированная архитектура в единой информационной среде Университета.

Отдельной задачей является создание единого механизма аутентификации пользователей приложений. Решение которой обеспечит формирование сообщества пользователей веб-сервисов, мониторинг востребованности информационных ресурсов Университета. Все это благоприятным образом скажется на качестве программных продуктов. Избавит от дублирования информации в информационных системах, обеспечит интеграцию информационных систем ВУЗа.

Интеграцию веб-сервисов следует выполнять на основе лучших практик, с применением мировых стандартов.

Рассмотрим существующие стандарты, алгоритмы и протоколы, описывающие алгоритмы взаимодействия в рамках сервис-ориентированной архитектуры.

СТАНДАРТЫ И АЛГОРИТМЫ ВЗАИМОДЕЙСТВИЯ

Можно выделить три объекта взаимодействия:

- заказчик (service requestor);
- исполнитель (service provider);
- каталог UDDI.

Схема взаимодействия представлена на рисунке 1.

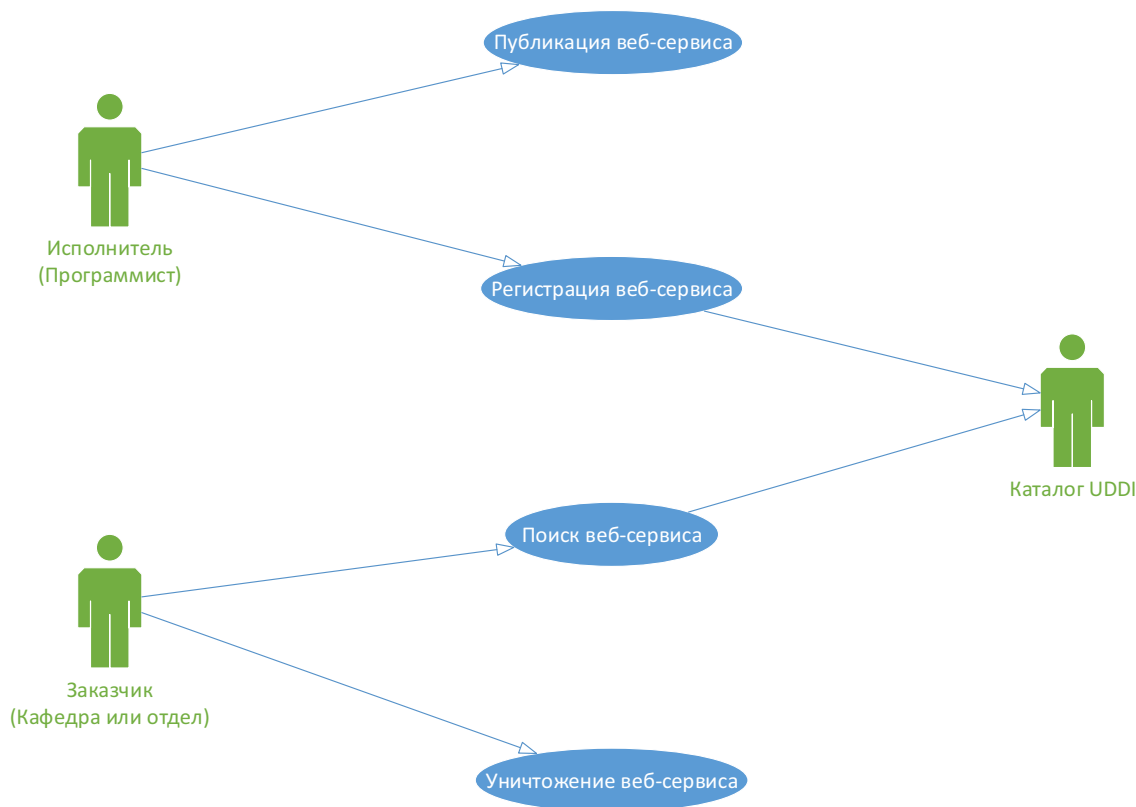


Рис. 1. UML-схема вариантов использования

Когда служба разработана, исполнитель регистрирует её в едином каталоге, где её могут найти потенциальные заказчики. Заказчик, найдя в едином каталоге подходящую службу, импортирует предоставленную спецификацию и разрабатывает, в соответствии полученной информацией, свое программное обеспечение. Для обеспечения взаимодействия используются следующие стандарты:

XML(eXtensible Markup Language): Расширяемый язык разметки, предназначенный для хранения и передачи структурированных данных. Подробную информацию об этом формате представления данных можно найти на [сайте Консорциума Всемирной паутины](http://www.w3.org/)(World Wide Web Consortium, W3C)[2];

SOAP(Simple Object Access Protocol)[14]: Протокол обмена сообщениями на базе XML.

WSDL(Web Services Description Language)[15]: Язык описания внешних интерфейсов веб-службы на базе XML.

UDDI(Universal Discovery, Description and Integration)[3]: Универсальный интерфейс распознавания, описания и интеграции. Каталог веб-служб и сведений о компаниях, предоставляющих веб-службы во всеобщее пользование или конкретным компаниям.

На рисунке 2 представлен стек протоколов взаимодействия, конечным элементов которого является UDDI.

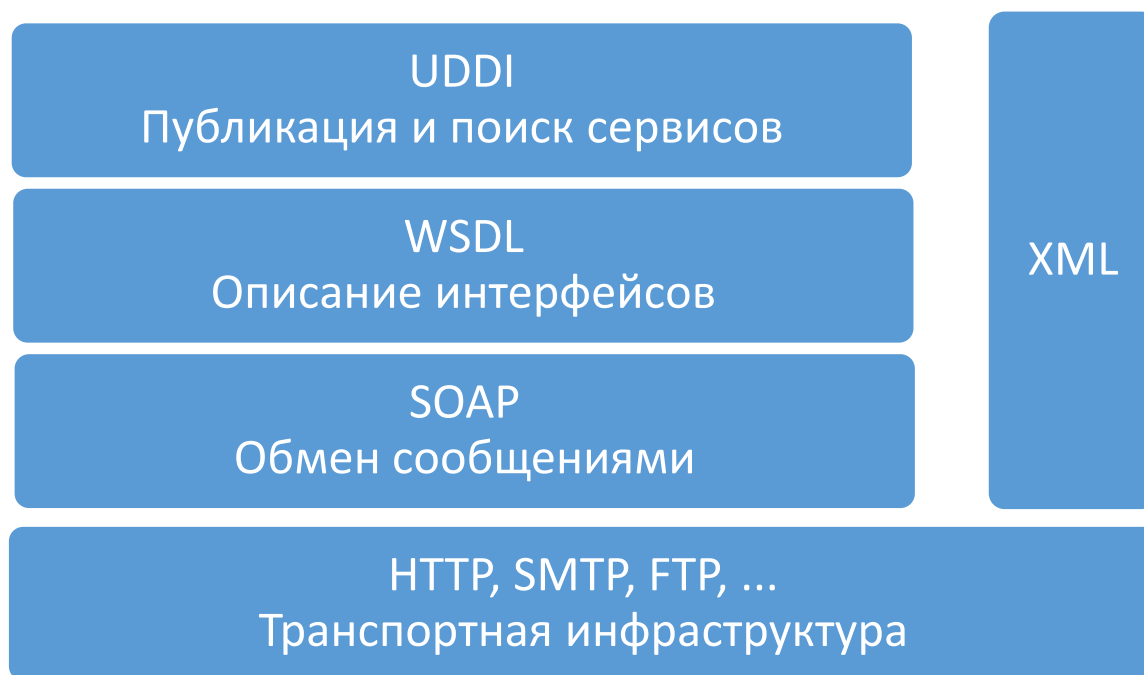


Рис. 2. Стек протоколов взаимодействия

Как следует из самого названия, стандарт UDDI предлагает механизм, с помощью которого компании могут "описывать" самих себя и свои услуги, а затем регистрировать и публиковать себя в UDDI-реестре. Другие организации могут искать такие опубликованные компании, обращаться к ним с запросом, или же "обнаруживать" их посредством SOAP-сообщений. После обнаружения подходящих партнеров организации, используя этот механизм, могут "интегрировать" с будущими партнерами свои услуги и предоставлять их своим клиентам.

Схема обнаружения приложения при использовании UDDI представлена на рисунке 3.



Рис. 3. Схема обнаружения приложения при использовании UDDI

После однократного поиска требуемого сервиса, дальнейшее взаимодействие поставщика и потребителя осуществляется напрямую, с использованием протокола SOAP.

UDDI позволяет работать как с SOAP так и с REST сервисами, проблемным моментом является лишь то, что UDDI заточен по SOAP архитектуру и не совсем соответствует идеям протокола REST. Поэтому в некоторых случаях, для REST сервисов более предпочтительным будет использование технологии DNS.

REST (сокр. англ. Representational State Transfer, «передача состояния представления» или «передача репрезентативного состояния») [16] — стиль построения архитектуры распределенного приложения. Данные в REST должны передаваться в виде небольшого количества стандартных форматов (например HTML, XML, JSON). Сетевой протокол (как и HTTP) должен поддерживать кэширование, не должен зависеть от сетевого слоя, не должен сохранять информацию о состоянии между парами «запрос-ответ». Утверждается, что такой подход обеспечивает масштабируемость системы и позволяет ей эволюционировать с новыми требованиями. Протокол REST наиболее подходит сервисам, которым не нужны более сложные взаимоотношения, кроме: «Создать», «Прочитать», «Изменить», «Удалить». Кроме того, REST по сравнению с SOAP, может оказаться и более производительным, так как не требует затрат на разбор сложных XML команд на сервере (выполняются обычные HTTP запросы — PUT, GET, POST, DELETE). Хотя SOAP, в свою очередь, более надежен и безопасен.

Для REST веб-сервисов аналогом WSDL является WADL. WADL еще не получил столь широкого распространения как WSDL, но это в большей степени обусловлено его новизной. Впервые данный формат был предложен как стандарт в 2009 году компанией Sun Microsystems.

UDDI позволяет указывать файл любого другого формата, не только WSDL и WADL, в качестве описания веб-службы. Но использование любых других форматов документов является отклонением от принятых мировых стандартов и может значительно затруднить разработку и доработку существующего программного обеспечения.

В случае, если создание WSDL файла вручную является делом проблематичным и время затратным, особенно для сервисов с большим количеством предоставляемых функций, существуют инструменты автоматической генерации файлов данного типа. Для PHP классов таким инструментом является IDE Zend Studio for Eclipse[9]. Удобной утилитой для генерации WSDL файлов веб-сервисов, написанных на языке C# является [WSDLGenerator](#)[10], либо это можно сделать стандартными утилитами, используя сервис IIS. Необходимо лишь добавить директиву «?wsdl» к *.asmx файлу. Для других языков программирования существуют аналоги, подобные IDE Zend Studio for Eclipse или WSDL генератору.

Услуги протокола UDDI гораздо шире приведенных. В случае интеграции веб-сервисов на уровне министерства, в соответствии со спецификацией UDDI, информация, предоставляемая при регистрации сервисов отдельных образовательных учреждений, в рамках единого хранилища, могут быть задействованы следующие компоненты:

- "Белые страницы" очень напоминают телефонную книгу и включают адрес, контактную информацию и известные идентификаторы. Университет мог бы представить себя следующим образом: BMSTU, Ul.Baumanskays 2-ya, 5, Moscow
- "Желтые страницы" также сходны с телефонным эквивалентом и охватывают распределения по отраслевым категориям, основанным на стандартных таксономиях, таких как NAICS (North American Industry Classification System, Система отраслевой классификации Северной Америки), UN/SPC (Dun and Bradstreet's United Nations/Standard Products and Services Classification, индикаторы компании "Дан энд Брэдстрит" для ООН/ Стандартная классификация продуктов и услуг), код SIC (Standard Industrial Classification, Стандартная отраслевая классификация) и так далее. Регистрация такой информации, как отраслевые коды, коды продуктов, географические коды и коды идентификации бизнеса (например, числа D-U-N-S, назначаемые компанией Dun & Bradstreet), позволяет другим поисковым службам использовать базовую классификационную информацию в качестве отправной точки для усовершенствования индексирования и классифицирования. Так, согласно классификации SIC, МГТУ им Н.Э.Баумана можно отнести к категории "Services", находящейся в подкатегории "Educational Services" (номер 8200 в категории SIC).

- "Зеленые страницы" содержат техническую информацию о службах, представляемых компаниями. Они включают ссылки на спецификации для Web-служб, а, при необходимости, также поддержку указателей на различные файлы и механизмы обнаружения, основанные на URL. Узлы UDDI-регистров хранят предшествующую информацию и дублируют окружающие данные для того, чтобы предоставлять такой же каталог информации из любого узла. Эта система напоминает архитектуру DNS, используемую при поиске доменных имен в сети Интернет.

Конкурентом UDDI может выступать ebXML — технология, созданная в 2001 году при поддержке ООН. Спецификации ebXML расширяют возможности основных технологий веб-сервисов. Для организации действительного взаимодействия типа business-to-business по Сети помимо основных технологий веб-сервисов требуется ряд дополнительных технологий. Например, консорциум электронной коммерции XML (Electronic Business XML, ebXML) определил всесторонний набор спецификаций для промышленной модели обмена XML-документами между торговыми партнерами. Спецификация обмена ebXML-сообщениями основана на "SOAP с вложениями" и не использует WSDL, но добавляет несколько сервисов качества обслуживания, такие как обеспечение безопасности, гарантированная доставка сообщений и согласованность модели процесса взаимодействия.

В настоящее время система ebXML потеряла свою актуальность, поскольку все основные поставщики программного обеспечения либо отказались от использования данной технологии (Sun Microsystems), либо отодвинули ее на третий план (IBM).

В мире веб-сервисов основные поставщики промышленного программного обеспечения уже достигли примирения по стержневым стандартам. Microsoft, IBM, Sun Microsystems, BEA Systems, Oracle, IONA и другие компании договорились о реализации SOAP, WSDL и UDDI. Однако нефундаментальные стандарты зачастую оспариваются. К таким спорам относятся разногласия Microsoft и IBM по вопросу определения последовательности действий, то есть продолжается противоборство [XLANG](#)[4] и [WSFL](#)[5] (Web Services Flow Language). Кроме того, остаются конкурирующие предложения по реализации контекста безопасности.

Дополнительные технологии в основном сконцентрированы в следующих ключевых направлениях:

- безопасность;
- поток управления процессом;
- транзакции;
- передача сообщений.

Продолжая совместный проект по расширению семейства спецификаций для поддержки веб-услуг на основе XML, корпорации IBM и Microsoft объявили о новой разработке - Web Services Inspection.

Спецификация WS-Inspection определяет процесс поиска описания веб-услуг XML на веб-серверах приложениями, позволяя разработчикам искать на веб-серверах XML-услуги. WS-Inspection дополняет технологию создания глобального каталога UDDI, которая также была разработана по инициативе IBM и Microsoft, - данный стандарт упрощает процесс поиска доступных услуг на веб-сайтах, не включенных в UDDI-директории.

WS-Inspection построен на основе технологии Microsoft SOAP Discovery, встроенной в Visual Studio .NET.

В качестве метода для поиска веб-услуг XML, WS-Inspection предлагает:

- Стандартный способ для поиска документов, содержащих описания веб-услуг;
- Возможность для провайдеров веб-услуг XML группировать наборы связанных описаний услуг, позволяя заинтересованным сторонам определять, какие описания связаны с той или иной услугой;
- Способ установки соотношения между XML-услугами и другими типами контента.

Реализации WS-Inspection доступны в последних версиях продуктов WebSphere Application Server Toolkit 6.1.1 и Microsoft Visual Studio .NET. WebSphere Application Server Toolkit можно скачать бесплатно на веб-сайте [IBM](#)[6].

ТРЕБОВАНИЯ К ЕДИНОЙ КОЛЛЕКЦИИ СЕРВИСОВ

Коллекцией сервисов в дальнейшем будет называться среда, в которой зарегистрированы все актуальные веб-сервисы Университета, для последующего их поиска другими приложениями и интеграции в свои системы.

Хорошей практикой является создание единого словаря тегов сервисов. Это позволит избежать затруднений в именовании часто используемых объектов работы сервиса. Пример тому объект «факультет», написание которого может отличаться: Faculti, Faqulty, Faqulti и т.д.

Для публикации Web-службы XML в UDDI Services через веб интерфейс, необходимо выполнить следующие операции:

- Создать объект Provider, в котором вы сможете публиковать свою службу.

- Ввести информацию о Web-службе XML, например имя Web-службы и категории (такие, как географическое положение). В соответствии с указанными категориями будет осуществляться классификация Web-службы.

- Добавить к Web-службе объект Binding, состоящий из точки доступа (URL) и описания. Например, для Web-службы Hello World следует ввести строку `http://localhost/webservice1/service1.asmx?op=helloworld`.

Если для Web-службы вы подготовили файл WSDL, можно при желании добавить его в качестве объекта tModel, а затем добавить новый объект tModel в качестве объекта Instance Information, относящегося к объекту Binding. Это необходимо для обнаружения wsdl файлов различными IDE, как например Visual Studio.

Также можно опубликовать Web-службу в UDDI Services с использованием функций API. В UDDI имеется два обширных класса API: Publish API и Inquiry API. Publish API предлагает механизм, с помощью которого поставщики служб могут зарегистрировать себя и свои службы в UDDI-регистре. Этот интерфейс позволяет подписчикам служб искать доступные службы. Inquiry API обеспечивает два типа вызовов: механизмы поиска (представляющий собой алгоритм углубления, посредством которого можно начать поиск с самого высокого информационного уровня и продолжить углубляться, пока не найдена требуемая информация) и извлечения, применяемый, когда вся информация, которую необходимо найти в службе, имеется в наличии.

РЕГИСТРАЦИЯ СЕРВИСА В ЕДИНОМ КАТАЛОГЕ

Сообщения в Publish API представляют собой команды, которые используются для опубликования и обновления информации, содержащейся в UDDI-совместимом регистре. Каждая компания должна первоначально выбрать один сайт-оператор (Operator Site), который будет хранить ее информацию. После того как он выбран, эту информацию можно будет только обновлять. Вызов методов в Publish API требует авторизации и выполняется обычно в рамках механизма защиты. HTTPS используется для всех методов в Publish API. Этот интерфейс включает следующие функции:

Четыре сообщения для сохранения каждой из четырех структур: `save_business`, `save_service`, `save_binding`, `save_tModel`. Каждое из этих сообщений "сохранить" (save) принимает в качестве входных данных authToken и один или несколько соответствующих структур.

Четыре сообщения для удаления каждой из четырех базовых структур: `delete_business`, `delete_service`, `delete_binding`, `delete_tModel`. Все они принимают соответствующий ключ uuid в качестве параметра. Успешное завершение или сбой

сообщений удаления указывается в соответствующем коде возвращаемой SOAP-структуре dispositionReport. Для того, чтобы удалить бизнес, который был создан для BMSTU, нам потребовалось бы передать следующую структуру XML:

```
<delete_business generic="1.0" xmlns="urn:uddi-org:api" >  
  
    <authInfo>Some auth info here </authInfo>  
  
    <businessKey>83B31200-7581-11D5-882B-0004AC48CC1E  
  
    </businessKey>  
  
</delete_business>
```

Поиск веб-сервиса в едином каталоге

Поиск информации о сервисах в UDDI производится с помощью Inquiry API. Запросы, которые предоставляет класс Inquiry API, могут быть разделены на две категории, а именно: просмотр и углубление:

Просмотр: UDDI поддерживает четыре типа обращений для этой модели. find_business обычно, вызывается, когда доступна любая частная информация, и возвращает структуру businessList, которая является укороченным набором ключевой информации из структуры businessEntity. Для подходящего совпадения можно несколько раз проходить через BusinessList, а затем использовать сообщение find_service для того, чтобы получить отдельные типы службы. Каждое из этих обращений "поиск" (find) принимает также факультативный элемент findQualifiers, который меняет метод поиска, используемый по умолчанию. Примерами findQualifiers являются "exactNameMatch", "caseSensitiveMatch", "sortByNameAsc" и так далее. Эти вызовы "поиск" поддерживают атрибут maxRows, который определяет максимальное число возвращаемых результатов поиска. На рисунке 4 представлена схема взаимодействия классов, при использовании Inquire операции.

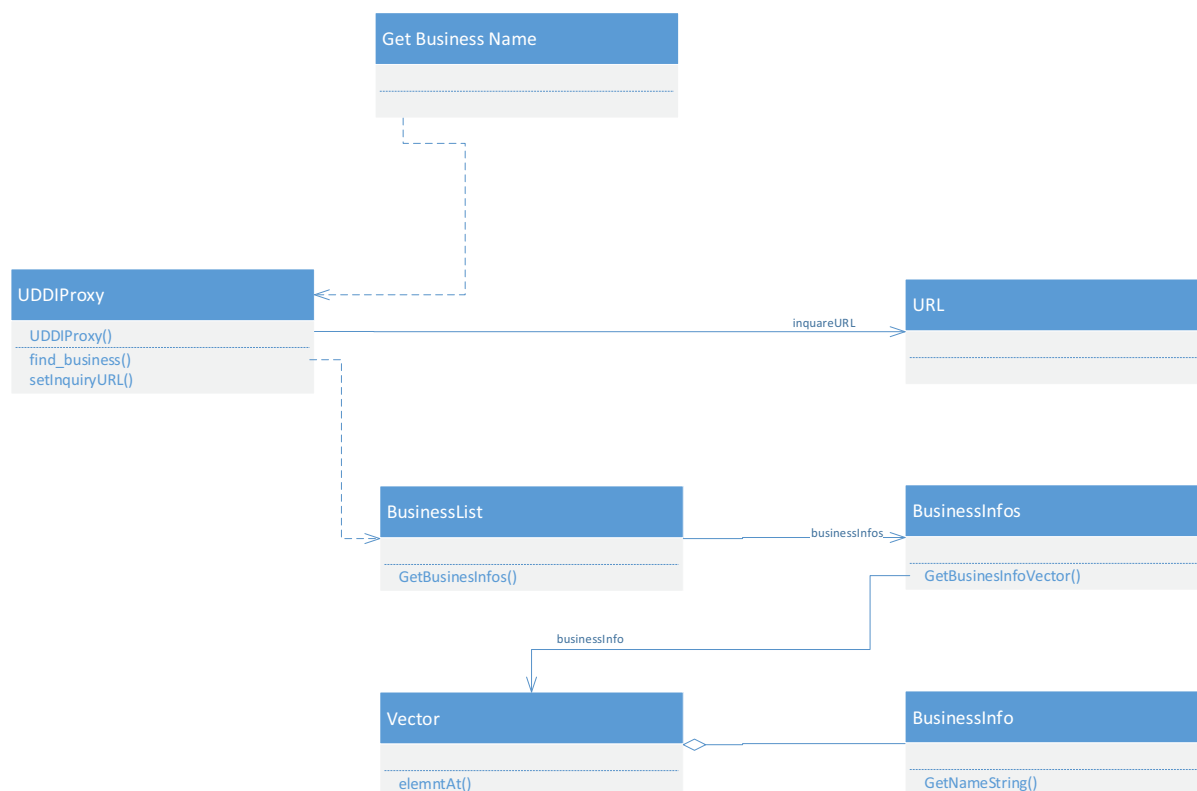


Рис. 4. Модель взаимодействия UML при использовании Inquire операции

Углубление: если доступен ключ для одного из четырех основных типов данных, этот ключ можно использовать для получения информации, касающейся отдельного экземпляра этого типа данных. Этот вызов "получить" (get) применяется для получения информации, относящейся к любому типу данных для данного ключа. Таким образом, это вызовы: `get_businessDetail`, `get_serviceDetail`, `get_bindingDetail`, `get_tModelDetail` и, наконец, `get_businessDetailExt`, который предоставляет подробную информацию о `businessEntity`.

АУТЕНТИФИКАЦИЯ

Наиболее важными из дополнительных технологий веб-сервисов являются технологии безопасности. Безопасность подразумевает конфиденциальность и целостность данных веб-сервисов. Никто, кроме конечного получателя данных, не должен иметь возможность изучать или изменять содержание сообщения. Вопрос безопасности также важен в плане контроля доступа к веб-сервисам, особенно при совместном использовании нескольких веб-сервисов.

Для авторизации и аутентификации предложено несколько стандартов (например, Security Authorization Markup Language, SAML). Также имеются стандарты для шифрования открытым ключом (XML Key Management Specification, XKMS). Основой всей безопасности Интернета является протокол защищенных сокетов (Secure Socket

Layer, SSL), а для HTTP-протоколов - HTTPS (secure HTTP), обеспечивающие базовый уровень безопасности.

Помимо HTTPS, брандмауэров, SAML, XKMS, использования цифровых подписей и XML-кодирования компания Microsoft предложила системы WS-License (управления удостоверениями безопасности) и WS-Security (распространение удостоверений безопасности, относящихся к взаимодействию с веб-сервисами). Аутентифицировать пользователей в системе UDDI позволяет Центральная Служба Аутентификации (CAS), протокол, обеспечивающий технологию единого входа в веб-сервисах. Данная служба установлена и запущена в МГТУ им. Баумана, но требует определенной настройки перед подключением сервиса. Протестировать работу единой системы аутентификации пользователей можно по адресу: <https://srv2k8r2-ca-01.auth.eusvc.bmstu.ru:8443>.

МОНИТОРИНГ ИСПОЛЬЗОВАНИЯ СЕРВИСОВ

Компания IBM предоставляет отличный инструмент для мониторинга использования сервисов IBM Services Monitoring and Management (ISMM). Среда управления и мониторинга сервисов дает возможность выполнять мониторинг сервисов, предоставляет методы для руководства и управления, позволяет контролировать развернутые сервисы, обеспечить гибкость развертывания сервисов и их взаимодействий с целью удовлетворения потребностей бизнеса.

Более подробную информацию о мониторинге сервисов и средствах, позволяющих это выполнить, можно прочесть в статье Юйчан Цзяо, инженера-программиста, IBM [«Управление и мониторинг сервисов во время исполнения»](#). [7]

СХЕМА ВЗАИМОДЕЙСТВИЯ МОДУЛЕЙ

Для того, чтобы создать единую информационную среду Университета, необходимо обратиться во все отделы, предоставляющие работающие веб-сервисы с просьбой сгенерировать WSDL или WADL файл и кратко описать сервис в UDDI регистре. При чем сделать это можно как через веб-интерфейс, так и используя Publish API UDDI.

После размещения информации в регистре UDDI, можно воспользоваться возможностями Visual Studio, которая позволяет автоматически получать информацию из UDDI регистра обо всех изменениях сервиса. Для этого необходимо сделать Subscribe к данному сервису и указать его в зависимостях проекта. После чего Visual Studio автоматически разберет WSDL файл указанный в параметрах, на составляющие и предоставит удобные инструменты доступа к предоставляемым веб-сервисом функциям. На данный момент Visual Studio работает только с файлами в формате WSDL, но

существует утилита, преобразующая WADL файл в прокси-класс, который можно подключить к проекту[11].

По умолчанию взаимодействие с веб-службами по сети возможно с использованием стандартных отраслевых протоколов, включая протокол SOAP. А именно, клиент и веб-служба обмениваются данными с помощью сообщений SOAP, в которых входные и выходные параметры инкапсулированы в виде XML. К счастью, в случае клиентов веб-служб сопоставление параметров с элементами XML и последующая передача сообщений SOAP по сети выполняются прокси-классом.

На основе сложившегося алгоритма взаимодействия модулей единого репозитория сервисов, описана схема взаимодействия модулей, представленная на рисунке 5.

Рис. 5. Блок-схема взаимодействия модулей единого репозитория сервисов

МОДЕЛЬ ИНФОРМАЦИОННОЙ СРЕДЫ УНИВЕРСИТЕТА

Сложившуюся модель информационной среды университета можно описать следующим образом:

- Для того, чтобы приложения университета или пользователи могли воспользоваться услугами веб-сервиса, его необходимо зарегистрировать в каталоге UDDI.
- Пользователь или приложение университета получают аутентификационный токен в единой системе аутентификации университета.
- Токен передается во всех запросах взаимодействия с каталогом UDDI.
- Единый каталог служб Университета предоставляет информацию о существующих веб-сервисах и функциях, которые они предоставляют.
- Также пользователь или приложения могут подписаться на тот или иной сервис, в результате чего, они будут извещены о любых изменениях в описании сервиса или его функций.
- После получения информации о веб-сервисе и предоставляемых им функциях (WSDL файл), пользователь или приложение начинают напрямую взаимодействовать с веб-сервисами. Веб-сервис может требовать токен аутентификации, полученный в единой системе аутентификации Университета, для выполнения определенных запросов.
- Все взаимодействия с сервисами проходят через систему мониторинга, которая позволяет определить востребованность того или иного сервиса, а так же получить информацию об ошибках, произошедших в процессе работы.

Модель информационной среды Университета представлена на рисунке 6.

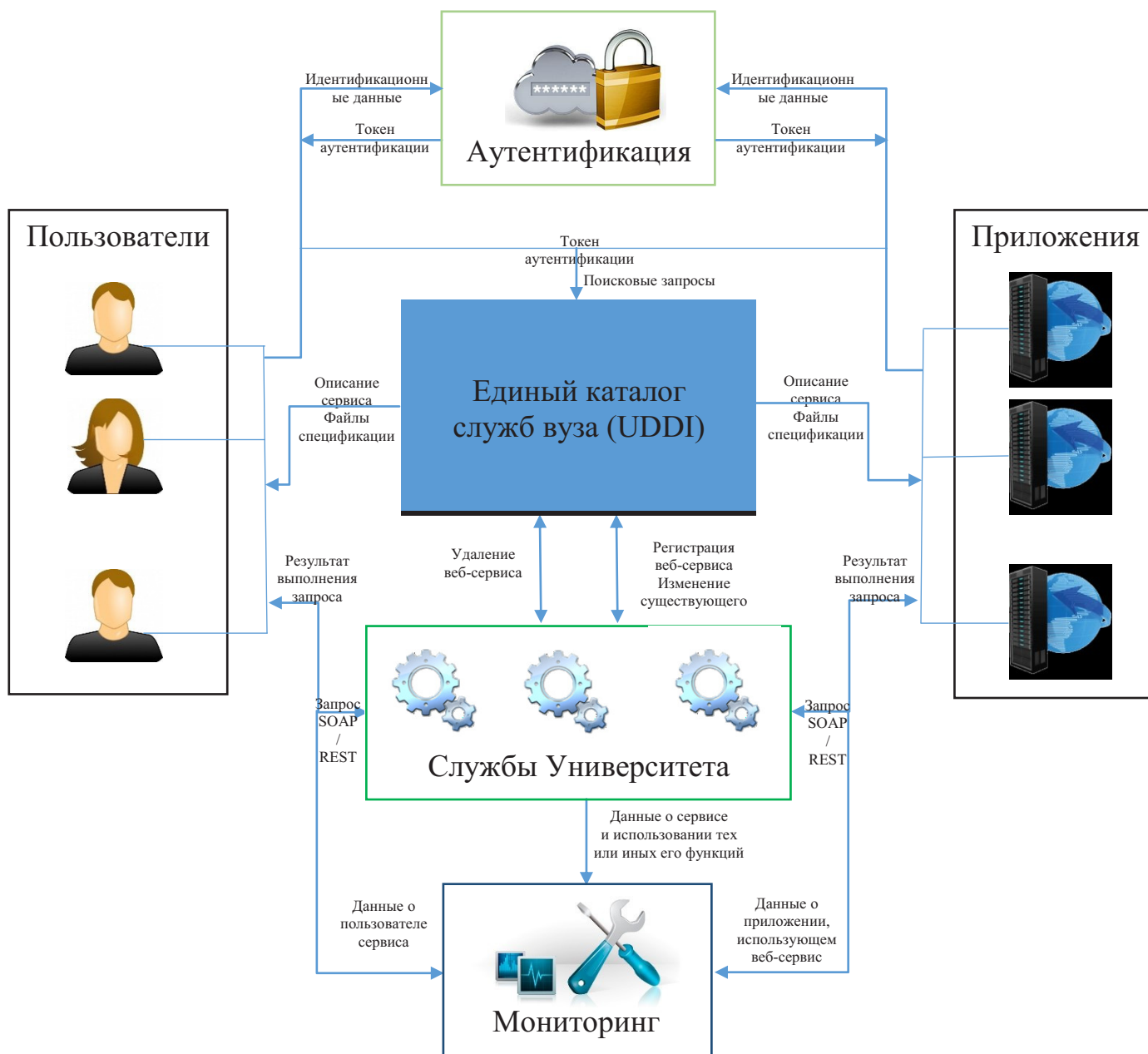


Рис. 6. Модель информационной среды университета

ВЫВОДЫ

Взаимодействие различных приложений университета через веб-сервисы определяет соответствие сервисно-ориентированной архитектуре. Для облегчения обращения приложений Университета и пользователей к веб-сервисам, необходимо воспользоваться единым каталогом сервисов UDDI.

Для того, чтобы UDDI выполняла возложенные на нее задачи, необходима адаптация сложившегося в Университете алгоритма создания веб-сервисов. Жизненно необходимо, в этом случае, предоставлять некоторое описание интерфейсов доступа к веб-сервису и предоставляемых им услугах. Данное описание должно быть в формате WSDL. Который стандартизированным образом описывают веб-сервисы и механизм доступа к ним.

В рамках Университета, подключение к единому каталогу сервисов позволит оптимизировать работу информационных отделов, которые занимаются разработкой сервисов, предоставляющих информацию об учебном процессе и людях участвующих в нем. К одному из таких сервисов относится «Контингент студентов», предоставляющий информацию о студентах, разбитую по кафедрам и факультетам. Данную информацию можно использовать, к примеру, для автоматической генерации паролей к Wi-Fi, на основе нового списка студентов в данном семестре. Данный сервис предоставляет WADL файл, описывающий все необходимые интерфейсы взаимодействия для автоматической генерации паролей к Wi-Fi.

Подключение к каталогу UDDI должно производиться через единую систему аутентификации пользователей, для обеспечения требуемого уровня безопасности.

После обнаружения сервиса и получения информации о предоставляемых им функциях, приложение или пользователь может напрямую обращаться к веб-сервису, минуя единый каталог UDDI.

Все взаимодействия с сервисами проходят через систему мониторинга, которая позволяет определить востребованность того или иного сервиса, а так же получить информацию об ошибках, произошедших в процессе работы.

Список литературы

1. David Sprott and Lawrence Wilkes, Understanding Service-Oriented Architecture, 2004, URL: <http://msdn.microsoft.com/en-us/library/aa480021.aspx> (дата обращения 01.10.2013)
2. Liam Quin, Extensible Markup Language, URL: <http://www.w3.org/XML/> (дата обращения 01.10.2013)

3. Ананд Раджарам, Обзор стандарта UDDI, ЖУРНАЛ BPM WORLD, 2001, URL: <http://www.iso.ru/rus/document5889.phtml> (дата обращения 01.10.2013)
4. Microsoft Corporation, XLANG/s Language, URL: <http://msdn.microsoft.com/en-us/library/aa577463.aspx> (дата обращения 01.10.2013)
5. IBM, Web Services Flow Language, URL: <http://www.ibm.com/developerworks/library/ws-ref4/> (дата обращения 01.10.2013)
6. IBM, WebSphere Application Server Toolkit Feature Pack for Web Services, URL: <http://www-01.ibm.com/support/docview.wss?uid=swg24016184> (дата обращения 08.10.2013)
7. Юйчан Цзяо, Управление и мониторинг сервисов во время исполнения, 2012, URL: <http://www.ibm.com/developerworks/ru/library/ws-servicemonitoring/> (дата обращения 08.10.2013)
8. Анатолев А.Г., Цыганенко В.Н., Юдин Е.Б., Веб-сервисы как средство интеграции приложений в WWW, URL: <http://www.4stud.info/networking/web-services.html> (дата обращения 20.10.2013)
9. ZEND, Zend Studio - The Leading PHP IDE, URL: <http://www.zend.com/en/products/studio/> (дата обращения 20.10.2013)
10. Stefh, WSDLGenerator, URL: <http://wsdlgenerator.codeplex.com/> (дата обращения 20.10.2013)
11. Adnanmasood, Jbergman, WADL - RESTful WCF Services Client Code Generation Utility, URL: <http://wadl.codeplex.com/> (дата обращения 20.10.2013)
12. Ньюкоммер Э, Веб-сервисы. XML, WSDL, SOAP и UDDI. Для профессионалов, Издательский дом "Питер", 2003 г.
13. WC3, XSL Transformations, URL: <http://www.w3.org/TR/xslt20/> (дата обращения 07.11.2013)
14. Wikipedia, SOAP, URL: <http://ru.wikipedia.org/wiki/SOAP> (дата обращения 07.11.2013)
15. Арулази Десиасилан, Что нового в WSDL 2.0, URL: http://citforum.ru/internet/xml/wsdl2_0/ (дата обращения 07.11.2013)
16. Enartemy, Архитектура REST, URL: <http://habrahabr.ru/post/38730/> (дата обращения 07.11.2013)