

**УПРАВЛЕНИЕ, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ИНФОРМАТИКА**

УДК 004.9

**РАСПРЕДЕЛЕННАЯ КОРПОРАТИВНАЯ ТРАНСПОРТНАЯ СИСТЕМА  
НА ОСНОВЕ СЕРВИС-ОРИЕНТИРОВАННОЙ АРХИТЕКТУРЫ**

© 2012 г. А.Н. Иванченко, А.А. Масленников

Южно-Российский государственный  
технический университет  
(Новочеркасский политехнический институт)

South-Russian State  
Technical University  
(Novocherkassk Polytechnic Institute)

*Рассматриваются основные принципы сервис-ориентированной архитектуры и её преимущества при создании информационных систем на основе распределенной архитектуры. Приводится обзор разработки компании IBM в области корпоративных транспортных систем. Выполняется анализ основных принципов разработки и сфер применения корпоративных транспортных систем. Описывается разрабатываемая авторами транспортная система и ее функциональные возможности.*

*Ключевые слова:* сервис-ориентированная архитектура; информационная система; транспортная система; веб-сервис; база данных; веб-приложение.

*In article considers the basic principles of service-oriented architecture, the benefits in creating information systems based on distributed architecture. Provides an overview of IBM's development of corporate transportation systems. Examines the basic principles of design and purpose of corporate transportation systems. Considers a developed transportation system, its functionality.*

*Keywords:* service-oriented architecture; information systems; transportation systems; web service; database; web-based application.

В процессе развития информационных систем и технологий предлагались различные варианты построения архитектуры крупных информационных систем. К настоящему времени достаточно широкое распространение получила идея создания информационных систем на основе распределенной архитектуры, и в связи с этим актуальной стала проблема объединения узлов распределенной системы в единое информационное пространство. Для решения этой проблемы используют концепцию сервис-ориентированной архитектуры (*Service Oriented Architecture, SOA*) [1].

Сервис-ориентированная архитектура – это парадигма организации и использования распределенных информационных ресурсов, таких как приложения и данные, находящиеся в сфере ответственности разных владельцев, для достижения желаемых результатов потребителем, которым может быть конечный пользователь или другое приложение [2]. Важно отметить, что *SOA* не имеет какой-либо универсальной реализации (не является коробочным продуктом, не имеет конкретных аналогов), которая подошла бы всем предприятиям. Для каждой компании *SOA* будет представлять собой некую интеграцию информационных ресурсов, служб и услуг. Реализация технологий предоставления подобных услуг может быть различной и зависит от потребностей компании, которая нуждается в подобной концепции.

Одним из основных преимуществ *SOA* является возможность интеграции существующих и вновь соз-

даваемых приложений. Как правило, невозможно реализовать всю бизнес-логику крупной компании готовыми решениями, и для определенных задач создаются уникальные информационные системы. Однако построить новую систему без ее взаимодействия с уже существующими системами практически невозможно (иначе такая система будет содержать не актуальную информацию, и ее внедрение не принесет желаемого успеха).

Важнейшей задачей при разработке распределенной системы является создание транспортной подсистемы передачи данных. Конечно, можно для каждого нового проекта разрабатывать свою транспортную подсистему, создавать свои особые протоколы обмена информацией, но при этом такая система не сможет взаимодействовать с существующими в сети потоками информации, получать актуальные данные о загрузке сети, количестве работающих информационных систем, связи с ними в едином формате.

Построение транспортной системы на основе единого протокола обмена данными открывает множество преимуществ:

– единый протокол обмена данными подразумевает и единый способ управления данным протоколом – т.е. *IT*-служба компании сможет легко управлять ситуацией внутри корпоративной сети (а не изучать транспортную подсистему каждой информационной системы в отдельности, сочетать их настройки таким образом, чтобы они не мешали друг другу функционировать в корпоративной сети и т.д.);

– транспортная система, в которой все сообщения представляются единым форматом, позволяет связать воедино множество различных систем (как правило их в крупных и средних компаниях достаточно много: продукты компании IC, геоинформационные системы и пр.);

– единая транспортная система, в которую заложены идеи управления нагрузкой на сеть, может сама анализировать сложившуюся ситуацию и управлять процессами передачи данных, снижая или повышая загрузку определенных узлов (серверов обработки данных транспортной системы);

– безопасность и сохранность передаваемых данных – в единой транспортной системе искать отправленные или полученные данные (или информацию о результатах получения / передачи данных) гораздо легче, чем обращаться к каждой существующей информационной системе и анализировать журналы транспортных протоколов (единый формат транспортного журнала, единое описание событий);

– распределенная транспортная система сама берет на себя обязанности по синхронизации всех необходимых сведений между узлами;

– возможность предоставления прикладных программных интерфейсов (API) для взаимодействия программистов с существующим протоколом единой транспортной системы, т.е. вновь разрабатываемое приложение может использовать подобные интерфейсы для обмена сообщениями с другими приложениями.

Сегодня одним из основных решений, реализующим описанные выше идеи, является программный продукт *WebSphere MQ* фирмы *IBM*, занимающий 70 % рынка в сегменте корпоративных транспортных систем и ставший стандартом де-факто для систем обмена сообщениями [3].

Решения с использованием *IBM WebSphere MQ* в

качестве транспортного слоя между корпоративными информационными системами внедрены в десятки тысяч предприятий по всему миру. Данный продукт обеспечивает одноразовую гарантированную доставку сообщений, что позволяет решать проблему «плохих» каналов связи.

Основными преимуществами *IBM WebSphere MQ* являются: высокая производительность; более простое управление системами обмена сообщениями, использующими схему «издатель – подписчик» и технологию *Java Message Service (JMS)*; поддержка технологии *Web 2.0*; простой в использовании интерфейс; высокобезопасная система шифрования данных; поддерживаемые операционные системы: *AIX, HP Unix, System i, Linux, Sun Solaris, Windows, z/OS*.

Система *IBM WebSphere MQ* достаточно мощная и надежная, но в то же время обладает множеством функций, которые зачастую просто не используются. Также система требует детальной настройки и привлечения специалистов компании *IBM* либо партнеров данной компании, т.е. помимо закупки самой системы необходимо планировать затраты на поддержку и сопровождение. Если учесть, что крупные компании включают множество филиалов и в каждом филиале необходимы специалисты, управляющие серверами *IBM WebSphere MQ*, то совокупная стоимость владения данной системой значительно увеличивается.

В связи с указанными сложностями появилась идея разработки транспортной подсистемы с интуитивно понятными настройками, работающей на общепринятых принципах, открытой для доработки и надстроек и реализующей принципы *SOA*.

Разрабатываемая транспортная подсистема базируется на трех основных технологиях: *ASP.NET*, веб-сервисы и базы данных (рис. 1).

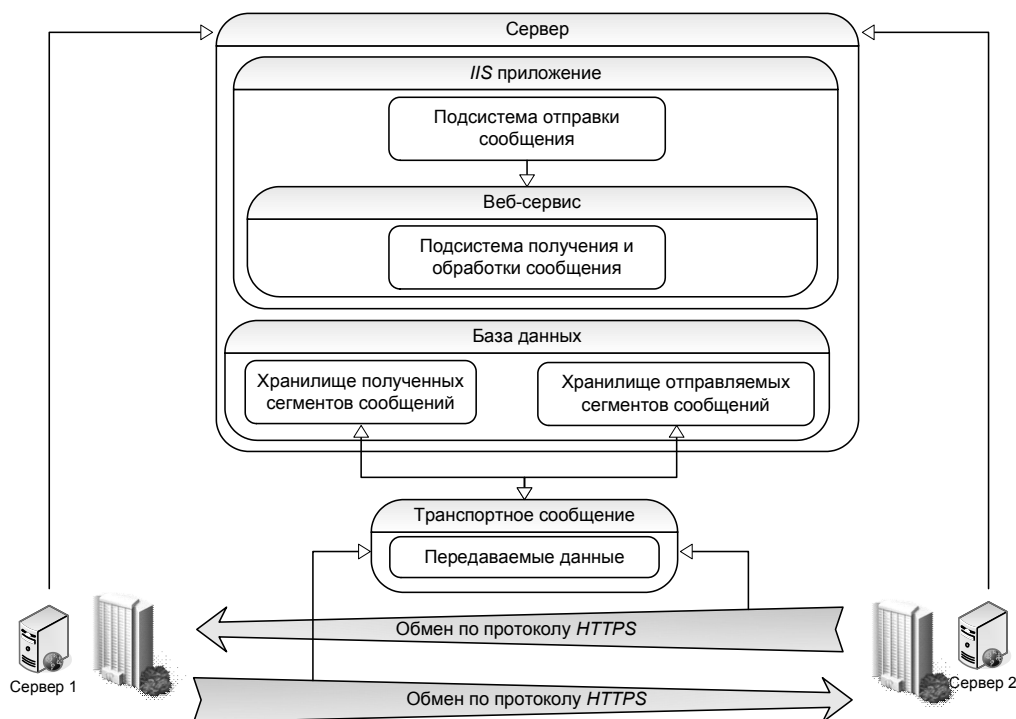


Рис. 1. Структура системы и формат транспортного сообщения

На платформе *ASP.NET* строится веб-приложение, которое осуществляет отправку сообщений, размещение их в хранилища сообщений, анализ состояний системы и выполнение действий по переводу системы из одного состояния в другое в соответствии с системными журналами событий.

Передаваемые между узлами сообщения, в зависимости от размеров, могут подвергаться сегментации. Сегментация позволяет снизить нагрузку на каналы связи между узлами, что важно, например, для «слабых» каналов связи.

Основу транспортной системы составляют веб-сервисы. Обращение к веб-сервису производится по защищенному протоколу *HTTPS* и, кроме того, в каждое сообщение добавляется *SOAP*-заголовок [4], содержащий зашифрованные сведения об имени пользователя и его пароле для доступа к данному веб-сервису, что позволяет дополнительно защитить систему от несанкционированного доступа.

Естественно, что на каждом узле системы должен существовать справочник, содержащий, как минимум, информацию об адресе каждого узла системы (точнее адресе веб-сервиса, к которому будет происходить обращение) и данных пользователя (имя и пароль) для доступа к веб-сервису. Справочник может быть обновлен при получении сообщения с определенным идентификатором (например, когда на одном из узлов произошло изменение информации о данном узле и все остальные узлы должны получить обновленный справочник). Справочник можно экспортировать в файлы формата *XML* для доступа сторонних информационных систем.

Каждый веб-сервис содержит методы по получению информации от узлов системы. Для того чтобы стандартизировать процесс обмена сообщениями между узлами, разработан общий формат транспортного сообщения (рис. 2). Только сообщения данного формата принимаются к рассмотрению веб-сервисом.

Количество веб-сервисов на каждом узле системы ограничивается только аппаратными ресурсами. Каждый веб-сервис согласно концепции *SOA* можно предоставлять другим приложениям или пользователям как услугу передачи данных в транспортной системе.

Таким образом, каждая работающая на данном узле задача (под задачей понимается процесс выполнения какого-либо приложения, веб сайта и т.д.) может использовать свой веб-сервис для отправки и получения данных. При этом каждый веб-сервис, работающий для определенной задачи, находится в постоянной связи с другими – в плане получения актуальной информации о загрузке транспортной системы, свободных ресурсах сервера, текущей производительности и других характеристиках. При этом приложение управления транспортной системой постоянно балансирует нагрузку на сеть и сервер, обрабатывающий запросы.

База данных системы размещается на каждом узле и состоит из двух основных хранилищ:

- хранилище полученных сегментов сообщений – каждое сообщение (не являющееся последним сообщением в группе полученных сообщений) помещается в данное хранилище, чтобы после получения «закрывающего» сообщения (значением поля «Идентификатор положения сообщения в группе» является значение «Последнее сообщение в группе») выбрать данные сообщения и сформировать из них исходное;

- хранилище отправляемых сегментов сообщений – каждое отправляемое сообщение сначала помещается в данное хранилище и уже подсистема отправки сообщений просматривает данное хранилище, после чего отправляет его получателю (адрес получателя хранится в соответствующем поле каждого сообщения).

Транспортную систему можно настроить таким образом, чтобы отправляемые и получаемые сообщения сохранялись в базе данных системы в течение определенного времени. Это может помочь при тестировании и настройке транспортной системы, анализе передаваемых сообщений (изучении транспортных журналов) и т.д.

Упрощенная схема процесса функционирования подсистемы отправки сообщений представлена на рис. 3.

Упрощенная схема процесса функционирования подсистемы обработки входящих сообщений представлена на рис. 4.

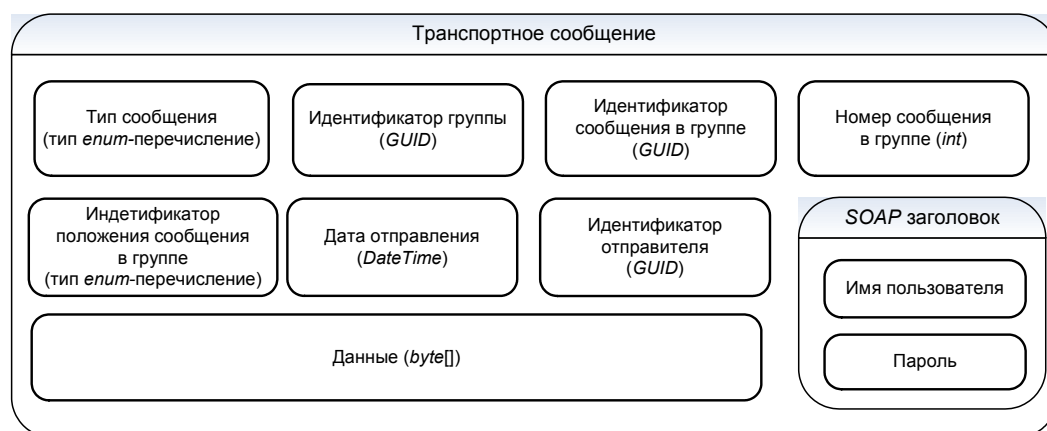


Рис. 2. Формат транспортного сообщения

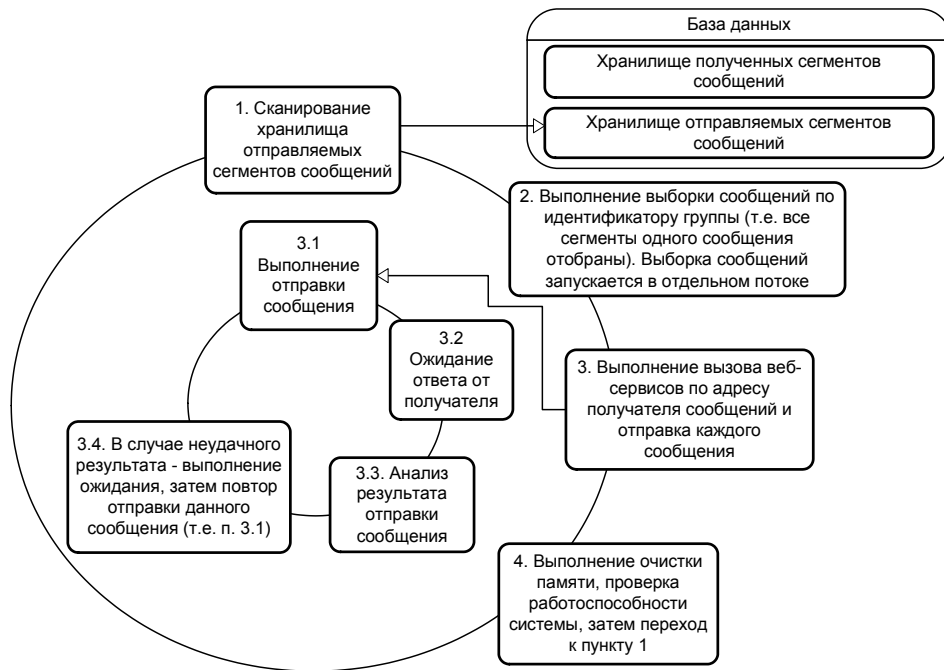


Рис. 3. Функционирование подсистемы отправки сообщений



Рис. 4. Функционирование подсистемы обработки входящих сообщений

Для доступа сторонних разработчиков (приложений) к транспортной подсистеме создаются библиотеки API, содержащие классы, интерфейсы и структуры обмена данными между узлами транспортной системы (филиалами компании). То есть разработчик вновь создаваемого приложения может использовать указанные библиотеки для доступа к транспортной подсистеме, избавив себя от необходимости проработки способов обмена информацией между узлами своей системы в корпоративной сети. На данном этапе разработки такие библиотеки предоставляются только для языка программирования C#. К веб-сервисам, являющимся основой всей транспортной системы, можно получить

доступ с помощью различных языков программирования, создав клиентские классы (с помощью утилит генерирования клиентских классов на основе *wsdl* – *Web Services Description Language* – язык описания веб-сервисов и доступа к ним, основанный на языке XML [5]).

После того как из полученных сегментов будет сформировано отправленное сообщение, система анализирует значение поля «Тип сообщения», и в зависимости от значения данного поля выполняет определенные действия. На основе рассмотренной транспортной системы можно создать подсистему устранения сбоев и слежения за состояниями системы во всей корпоративной сети.

Под состояниями системы можно понимать некие точки состояния информации, содержащейся в базах данных, файловой системе всех приложений, использующих транспортную подсистему, т.е. при записи или удалении информации сохраняются ссылки на записи о данной информации. Тогда процесс восстановления может просто учитывать необходимые точки, информация о которых размещается в системной таблице, а в свою очередь данные точки будут указывать на определенные данные, и процесс перевода системы из одного состояния в другое будет заключаться в удалении или восстановлении точек (и соответствующих записей базы данных или файловой системы). Тогда в случае какого-либо сбоя, системному администратору компании не нужно изучать множество различных приложений, работающих в сети компании (разбираться в форматах транспортных журналов каждого из них), затем восстанавливать базы данных каждого такого приложения вручную. Достаточно обратиться к Системе, которая просматривает все ранее сделанные точки восстановления и предложит варианты восстановления.

Помимо узлов, обменивающихся информацией между собой, должен существовать центральный узел, анализирующий сеть, в которой функционируют узлы, и посылающий управляющие воздействия.

Например, в фоновом режиме на центральном узле может работать подсистема проверки функционирования каждого узла, работающая по следующему алгоритму: происходит обращение к определенному узлу, если узел не отвечает, то обращение к приложению, представляющему системный уровень узла, и отправка ему управляющего воздействия. То есть каждое веб-приложение на каждом сервере снабжается службой, функционирующей в среде операционной системы и ожидающей получения управляющих воздействий от центрального узла. Когда такая служба получает сообщение, происходит обращение к базе данных основного веб-приложения и анализ точек восстановления. Затем в автоматическом режиме удаляются или восстанавливаются определенные данные, приведшие к выходу из строя системы.

Разрабатываемая система для каждой из основных типовых проблем разработки корпоративных приложений имеет достаточно удобное, простое в использо-

вании и надежное решение. Рассмотрим эти проблемы и ответы системы на них:

– проблемы безопасности – вся информация передается по защищенному протоколу *HTTPS*, каждое сообщение веб-сервиса передается вместе с зашифрованным *SOAP*-заголовком;

– проблемы масштабируемости – при расширении бизнеса компании, а вместе с тем и объемов обрабатываемой информации, не нужно докупать какие-либо компоненты, система предполагает обрабатывать большие объемы информации; способы обращения и настройки менять не нужно (система сама выполняет распределение нагрузки);

– проблемы переносимости между различными операционными системами – клиентские библиотеки (т.е. классы для доступа сторонних разработчиков к транспортной системе) могут быть использованы на различных платформах и операционных системах (операционные системы семейства *Windows*; операционные системы *Linux*, *Mac OS X*, *iPhone OS*, *Android*, *Sun Solaris*, *OpenBSD*, *FreeBSD*, *NetBSD* и т.д.);

– проблемы взаимодействия с гетерогенными источниками данных, почтовыми системами, *Web*-серверами – так как транспортная система построена на основе веб-сервисов, то при желании на каждой операционной системе и для желаемого языка программирования на основе *wsdl*-данных можно сгенерировать классы для обращения к данному веб-сервису и организовать свою надстройку над транспортной системой, получать и загружать данные.

#### Литература

1. Сервис-ориентированная архитектура // CITForum.ru: онлайн библиотека свободно доступных материалов по информационным технологиям, 2005. URL: <http://citforum.ru/internet/webservice/soa/> (дата обращения: 20.01.2012).
2. Сервис-ориентированная архитектура // Википедия: электронная энциклопедия, 2009. URL: [http://ru.wikipedia.org/wiki/Сервис-ориентированная\\_архитектура](http://ru.wikipedia.org/wiki/Сервис-ориентированная_архитектура)
3. Инновационные решения в технологиях и бизнесе // IBM Developer works: электронная библиотека документов компании IBM, 2009. URL: [http://public.dhe.ibm.com/software/dw/ru/download/2009\\_ibm2-1.pdf](http://public.dhe.ibm.com/software/dw/ru/download/2009_ibm2-1.pdf)
4. SOAP // Википедия: электронная энциклопедия, 2012. URL: <http://ru.wikipedia.org/wiki/SOAP> (дата обращения: 23.01.2012).
5. WSDL // Википедия: электронная энциклопедия, 2011. URL: <http://ru.wikipedia.org/wiki/WSDL>

Поступила в редакцию

27 января 2011 г.

**Иванченко Александр Николаевич** – канд. техн. наук, профессор, заведующий кафедрой «Программное обеспечение вычислительной техники», Южно-Российский государственный технический университет (Новочеркасский политехнический институт). Тел. +7-918-552-92-72. E-mail: [ian2008.52@mail.ru](mailto:ian2008.52@mail.ru)

**Масленников Алексей Александрович** – аспирант, кафедра «Программное обеспечение вычислительной техники», Южно-Российский государственный технический университет (Новочеркасский политехнический институт). Тел. +7-918-594-19-97. E-mail: [newsky98@mail.ru](mailto:newsky98@mail.ru)

**Ivanchenko Alexander Nikolaevich** – Candidate of Technical Sciences, professor, head of department «Software computer engineering» South Russia State Technical University (Novocherkassk Polytechnic Institute). Ph. +7-918-552-92-72. E-mail: [ian2008.52@mail.ru](mailto:ian2008.52@mail.ru)

**Maslennikov Alexsey Alexandrovich** – post-graduate student, department «Software computer engineering» South Russia State Technical University (Novocherkassk Polytechnic Institute). Ph. +7-918-594-19-97. E-mail: [newsky98@mail.ru](mailto:newsky98@mail.ru)