

## COMPUTER SCIENCE

**ИНТЕГРАЦИЯ ИНФОРМАЦИОННЫХ СИСТЕМ С  
ИСПОЛЬЗОВАНИЕМ MULE ESB**

*и. т. н. Бакунова О. М.,  
м. т. н. Калитеня И. Л.,  
студент Уласович В. Ю.,  
студент Тимофеев Д. О.,  
Ухналев Р. Ю.,  
Михаленко В. А.*

*Республика Беларусь, г. Минск, Институт информационных технологий  
Белорусского государственного университета информатики и радиоэлектроники*

**ARTICLE INFO**

Received 28 February 2018  
Accepted 15 March 2018  
Published 12 April 2018

**KEYWORDS**

integration,  
enterprise service bus,  
information system

**ABSTRACT**

Today developers often face not with designing or building of new software, but with integration with existing one. And quite often it's need to perform integration not only within an enterprise, but with customers, distributors, governments structures. But according to some points this task might become very complex, and it will require proper approaches and technologies to get scalable and flexible solution.

© 2018 The Authors.

**Введение.** В современном мире разработчики часто сталкиваются с задачами не по разработке нового программного обеспечения, а по интеграции уже существующих модулей или целых систем в единое целое. И довольно часто данная задача не ограничивается рамками предприятия, и типичной является ситуация, когда необходимо провести интеграцию с какими-либо информационными системами партнеров, клиентов, заказчиков или государственных структур.

При проектировании программного средства могут закладываться определенные архитектурные решения, которые в будущем позволят проводить интеграцию с другими системами более просто, например, посредством предоставления специального API. Но на практике чаще случается так, что необходимо провести интеграцию двух систем с большой концептуальной и технической разницей.

Концептуальная разница подразумевает противоречие тех идей и принципов, которые были заложены в основе разных систем. Технологическая же – о различиях в используемых форматах обмена данными, протоколах, технологиях.

Кроме этого, есть ряд других факторов, которые еще больше усложняют задачу по интеграции:

1. Невозможность отказаться от старых программных модулей и технологий, аппаратного обеспечения, даже в случае, если они морально устарели. К тому же, старое программное и аппаратное обеспечение может удовлетворять в полной мере всем требованиям по безопасности, быстродействию и надежности.

2. Для интеграции может потребоваться передачи важной и/или конфиденциальной информации по Сети, что накладывает требования по безопасности и шифрованию. Ранее данное могло и не требоваться, как в случае, если все данные компании передаются внутри частной виртуальной или локальной сети, без возможности доступа извне.

3. В случае интеграции высоконагруженных информационных систем (например, интеграция поставщика со всеми локальными дистрибьюторами крупного региона) требуется оптимальные решения для возможности безотказной обработки большого количества данных.

4. Должна быть обеспечена непрерывность цикла работы и высокая надежность. В противном случае может произойти потеря важных данных, что скорее всего приведет к финансовым убыткам.

#### 5. Прочие факторы.

Учитывая вышеописанные факторы, можно выделить несколько основных способов решения задач по интеграции нескольких информационных систем:

1. Интеграция на уровне конечного пользователя. Самый простой возможный подход, который заключается в ручном переносе необходимых данных из одной системы в другую непосредственно конечными пользователями. Т.к. отсутствует какая-либо автоматизация, то данный подход не потребует выполнения какой-либо автоматизации, что может быть достаточно дешево, что и является его основным преимуществом. Но также данный подход имеет и существенные недостатки, как, например, влияние человеческого фактора на возможность совершения ошибок при переносе данных, низкая скорость обработки данных, возможность относительно эффективного применения только в случае небольших объемов данных.

2. Использование общепринятых стандартных подходов. Данный подход подразумевает использование различных государственных, промышленных или технологических стандартов, что приведет к технологической унификации и позволит провести программную интеграцию двух разных информационных систем. Недостаток данного подхода заключается в необходимости внесения изменений в программные модули информационных систем, в случае если они не соответствуют стандартам. А внесение такого рода изменений может быть нецелесообразно (и даже невозможно) по многим причинам.

3. Использование общих источников данных. Данный подход заключается в том, что точкой интеграции будет служить, например, общая база данных. Данное решение может оказаться экономически выгодным, что несомненно будет являться преимуществом, но таит в себе некоторые существенные недостатки. Такого рода интеграции может стать причиной появления противоречивости в данных, если на уровне базы данных отсутствуют специальные механизмы по обеспечению ее целостности. А любые изменения в структуре базы данных потребуют внесения изменений в соответствующих программных модулях.

4. Использование общих форматов данных. Заключается оно в том, что во всех системах, участвующих в интеграции, строго фиксируется формат данных и соответствующим образом строится бизнес-логика. Однако в случае изменения каких-либо процессов или структуры данных хотя бы в одной системе, это потребует добавления узкоспециализированных решений.

5. Использование промежуточного программного обеспечения. В данном случае, для решения задачи интеграции нескольких систем используется специальное программное обеспечение, которое станет основной точкой обмена информацией между различными программными системами. Явным преимуществом такого подхода является возможность гибкой настройки форматов входных и выходных данных, возможность выполнения промежуточных преобразований, обеспечения транзакционности всех операций и шифрования данных. К недостаткам же можно отнести необходимость в специалистах, владеющих соответствующими технологиями и инструментами.

В качестве частного примера промежуточного программного обеспечения можно привести сервисную шину предприятия (англ. Enterprise Service Bus, ESB), что является разновидностью сервис-ориентированной архитектуры. Принцип данного программного обеспечения построен на модели событийно-ориентированного обмена сообщениями между различными информационными системами, что позволяет довольно эффективно использовать это для решения задач по интеграции.

Как пример прикладного решения можно привести интеграционную платформу Mule ESB, которая позволяет выполнять интеграцию различных информационных систем. Mule ESB написана на языке программирования Java и является открытым программным обеспечением и распространяется по лицензии CPAL.

Данная платформа обладает рядом особенностей, которые делают ее использование эффективной при интеграции разных информационных систем:

1. Будучи написанной на Java (а если точнее, то на платформе Java EE), Mule ESB является кроссплатформенным и может быть запущен на большом количестве сервере приложений для данной платформы, таких, как, Tomcat, WebSphere, Jetty, JBoss и другие.

2. По умолчанию, Mule ESB поддерживает работу с большим количеством протоколов - HTTP/HTTPS, FTP, SFTP, OAuth2, SOAP, REST, JMS, MQ, JBI, AQ, Caching, Email, IM, JCA, AS400 Data Queues и другие.

3. Доступно большое количество коннекторов для разных прикладных решений. Коннектор - готовый программный модуль, который позволяет работать с целевой системой

посредством использования ее API и/или специальных протоколов. Также имеются возможность написания своих коннекторов с использованием специального набора средств разработки (SDK), или использовать готовые сторонние решения. В официальной библиотеке Anypoint Exchange можно найти большое количество готовых коннекторов для интеграции с различными прикладными программными модулями.

4. Возможность развертывания Mule-приложений в облачной среде. Для этого можно использовать среду Anypoint Platform, которая позволяет запускать приложения в изолированном контейнере на облачной платформе. Также данная платформа предоставляет возможность использования очереди сообщений Anypoint MQ, доступ к журналам всех приложений, создание шлюза для API и портала с документацией, отправка email-оповещений в случае возникновения ошибок.

5. Mule ESB имеет мощные инструменты для обработки данных, такие как Mule Expression Language (MEL) и DataWeave. DataWeave - это шаблонизатор, который позволяет легко преобразовывать данные в большое количество разных форматов. Поддерживаются преобразования в XML, CSV, JSON, в обычные Java-объекты (POJO) и коллекции (List, Map, Set), доступны инструменты для группировки и агрегирования данных. MEL - скриптовый язык, который используется в потоках Mule для работы с данными (доступ к полезной нагрузке сообщения, переменным, свойствам), позволяет быстро и эффективно выполнять фильтрацию, перенаправления или иные действия с объектами сообщения Mule.

6. Возможность создания API с использованием специального языка RAML (RESTful API Modeling Language). Данный язык позволяет подробно описать API для приложения в удобном для чтения формате, похожем на JSON и автоматически создать соответствующие потоки, после чего останется лишь описать бизнес-логику у соответствующих методов. При развертывании такого API в облачной среде Anypoint Platform есть возможность подробного мониторинга всех запросов, настройка политик безопасности, автоматическое создание документации.

7. Также данная платформа предоставляет удобный инструмент для тестирования MUnit, которая позволяет разработчику легко создавать автоматизированные тесты для интеграций и API. Данный фреймворк полностью совместим со сборщиком проектов Maven и плагином Surefire, что позволяет использовать его вместе с методологией непрерывной интеграции.

Учитывая вышеописанные особенности, с использованием платформы Mule ESB разработчик получает легковесный, но в то же время мощный и гибкий инструмент, при помощи которого можно выполнить интеграцию двух информационных систем.

Большое количество поддерживаемых протоколов и богатая библиотека коннекторов дает возможность работать с большим количеством прикладных программных решений (например, сервера баз данных, CRM-системы, FTP-сервера и др.) без установки и написания дополнительного программного обеспечения; мощные инструменты по работе с данными позволяют выполнять необходимые преобразования входных и выходных данных, чтобы обеспечить совместимость со всеми системами, участвующими в интеграции. Также для Mule приложения мы можем стандартными средствами обеспечить шифрование данных и использовать различные методы аутентификации.

В случае, когда стандартных средств языка недостаточно, можно использовать все возможности языка программирования Java для расширения возможностей самой платформы, так и для описания специфической бизнес-логики для конкретной интеграции.

**Выводы.** Учитывая все вышесказанное можно сделать вывод, что выполнение задачи по интеграции информационных систем с использованием решений на ESB может стать оптимальным решением, поскольку это не требует внесения изменений в сами существующие системы, и вся задача по интеграции ложится на специальное промежуточное программное обеспечение. Как частный пример такого программного обеспечения, Mule ESB может оказаться очень эффективным при решении задач подобного рода.

**ЛИТЕРАТУРА**

1. David Dossot, John D'Emic, Victor Romero Mule in Action, Second Edition. Manning Publications, 2014.
2. Gregor Hohpe, Bobby Woolf Enterprise Integration Patterns: Designing, Building and Deploying Messaging Solutions. Addison-Wesley, 2004.
3. Мартин Фаулер Шаблоны корпоративных приложений: [пер. с англ.]. Вильямс, 2010.
4. О. Н. Образцова, О. М. Бакунова, Д. М. Кугач, А. В. Хомяков Практико-ориентированное обучение в сфере информационных технологий в БГУИР и сотрудничество вуза с ведущими компаниями IT // Проблемы современного образования: материалы VIII международной научной конференции, 10-11 сентября 2017. – Прага: Vědecko vydavatelské centrum «Sociosféra-CZ», 2017 - С.38-41
5. Бакунова О. М., Калитеня И. Л., Бакунов А. М., Малиновская Т. И. Применение ИКТ для оказания образовательных услуг лицам с особыми потребностями на примере изучения системы 1С дистанционно // Непрерывное профессиональное образование лиц с особыми потребностями: сборник статей международной науч.- практической конференции (Минск, 14 - 15 декабря 2017 года). – Минск: БГУИР, 2017. – С. 41 – 43.
6. Бакунова О. М., Калитеня И. Л., Бакунов А. М., Антонов Е. Д., Мелешкевич Д. В. Информационные компьютерные сети и системы в сфере образования // Непрерывное профессиональное образование лиц с особыми потребностями: сборник статей международной науч.- практической конференции (Минск, 14 - 15 декабря 2017 года). – Минск: БГУИР, 2017. – С. 39 – 41.
7. Бакунова О. М., Хмелевская А. Л., Беликов А. С., Мирончик А. Н., Агапкин Л. М., Чучвал А. Ю. Использование современных подходов и нейронных сетей для качественного образования в ВУЗах // I Международный симпозиум "Гуманитарные и общественные науки в Европе: достижения и перспективы"– Вена, Австрия 2018 г.