

УДК 004.75, 004.624

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИЙ GRID И ESB ДЛЯ ПОСТРОЕНИЯ ИНТЕГРИРОВАННОЙ СИСТЕМЫ ПОДДЕРЖКИ ЖИЗНЕННОГО ЦИКЛА ВОЗДУШНОГО СУДНА

© 2013 Ю.В. Полянков, В.В. Трясцин, М.М. Дронов

Ульяновский государственный университет

Поступила в редакцию 10.06.2013

Статья посвящена вопросам построения интегрированной системы поддержки жизненного цикла воздушного судна с использованием ESB и GRID технологий, а именно выбору архитектуры системы, описанию модели представления данных, разработке схемы обработки запросов.

Ключевые слова: интеграция приложений, интеграция данных, сервис-ориентированная архитектура, сервисная шина предприятия, GRID, Open Grid Service Architecture – Data Access & Integration.

ВВЕДЕНИЕ

Необходимость обеспечения единого информационного пространства предприятия для обеспечения непрерывной поддержки бизнес-процессов не вызывает сомнений. Другой вопрос, как именно осуществить интеграцию порождает много споров и предложений. Сегодня крупнейшие поставщики программного обеспечения, такие как Microsoft, IBM, Oracle, ведут агрессивную маркетинговую политику, стараясь увеличить свою долю на быстрорастущем рынке интеграции приложений. Международные консорциумы – W3C, WS-I, OASIS – разрабатывают стандарты, призванные упростить и унифицировать процессы интеграции [1].

Авиастроительные предприятия не являются исключением. Для обеспечения информационной поддержки жизненного цикла воздушного судна необходимо объединить множество разнотипных систем от различных производителей, таких как CAD, CAPP, ERP, PDM и т.д. Интегрированная система, объединяющая их, должна обеспечить доступ к функциям и данных друг друга. Поэтому использование только одного из существующих интеграционных подходов [2]: интеграция “каждый с каждым”, интеграция на уровне пользовательских интерфейсов, интеграция на уровне данных, интеграция на уровне корпоративных приложений, интеграция на базе сервис-ориентированной архитектуры – представляется недостаточным и предлагается использовать гибридный.

*Полянков Юрий Вячеславович, доктор технических наук,
профессор, Президент университета.*

E-mail: president@ulsu.ru

*Трясцин Виталий Викторович, аспирант кафедры ТТС.
E-mail: saunintv@gmail.com*

*Дронов Михаил Михайлович, стажер-исследователь
научно-образовательной лаборатории Центра CALS-
технологий. E-mail: m@dronov.net*

АРХИТЕКТУРА ИНТЕГРИРОВАННОЙ СИСТЕМЫ НА БАЗЕ SOA И GRID

Для обеспечения доступа к бизнес-логике приложений в рамках интеграционной системы предлагается использовать сервис-ориентированную архитектуру и сервисную шину предприятия (enterpriseservicebus, ESB), для обеспечения единого пространства данных – технологию GRID, а именно OGSA-DAI (Open Grid Service Architecture – Data Access & Integration).

Системы ERP,PDM,CAD и т.д. будут являться и поставщиками, и потребителями данных. Для их включения в платформу интеграции, поддерживающую SOA, необходимо реализовать для них адаптеры, которые позволят их рассматривать в интегрированном пространстве как сервисы. Общие данные, потребляемые этими системами, будет находиться в едином Grid-хранилище данных.

OGSA и SOA имеют общие стандарты, т.к. обе используют web-сервисы. OGSA направлена на создание, ведение и применение наборов сервисов, поддерживаемых виртуальными организациями. В отличие от традиционных веб-сервисов, которые позволяют обнаруживать инициировать только постоянные (persistent) службы, OGSA также предусматривает поддержку временных (transient) экземпляров сервисов, инициируемых и завершаемых динамически [3].

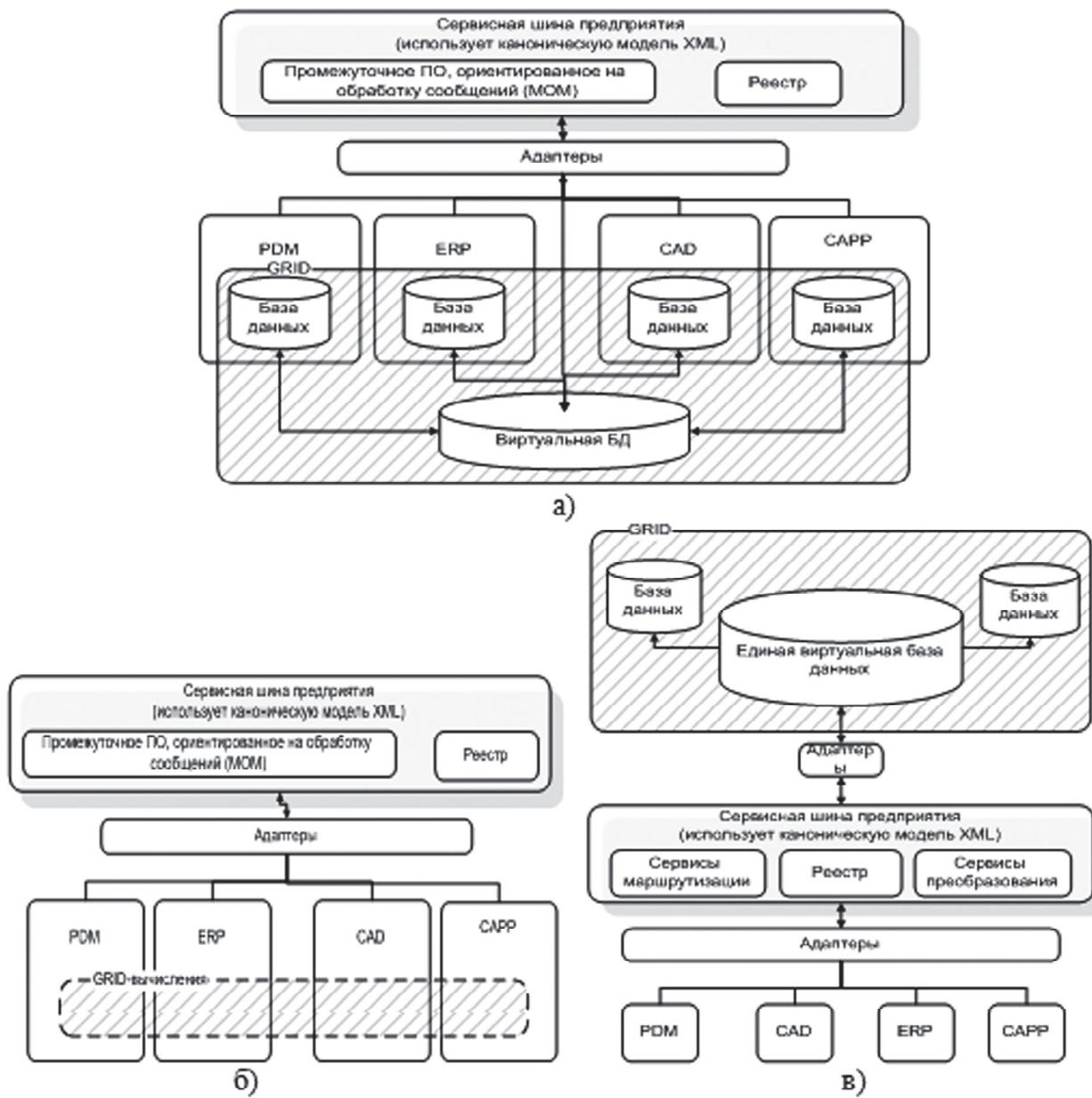
Существующие GRID-системы (проекты) можно разделить на научно-прикладной и коммерческий GRID [4]. Научно-прикладные проекты используют opensource-инструменты построения, коммерческий GRID использует продукты таких производителей как Oracle, HP, IBM. Среди коммерческого Grid, ориентированного на интеграцию данных широкое распространение получила СУБД Oracle, которая позволяет строить grid-систему с поддержкой разнородных БД, применяя средства самонастройки.

В рамках коммерческого GRID интегрированная система может иметь следующую архитектуру (рис. 1а): сервисная шина, объединяя приложения, а grid – данные систем. Такая архитектура возможна, если интегрируемые системы представляются для разработчика платформы «белым» или «серым» ящиком. В такой архитектуре можно использовать СУБД Oracle, данные будут или консолидироваться или федерализироваться.

Другой вариант построения GRID платформы интеграции на базе SOA (рис. 1б) подразумевает интеграцию уровня приложений и использование GRID для вычислительных задач на уровне функциональных систем. Такой подход возможен только при разработке функциональных систем на базе GRID. На практике эти вари-

анты практически не выполнимы, т.к. информационные системы не разрабатываются авиастроительным предприятием, а закупаются у разных вендоров.

Третий вариант (рис. 1в) подразумевает интеграцию приложений и использование grid-хранилища данных, в котором будут храниться общие неизменяемые данные для систем. Такой вариант позволяет использовать как коммерческий, так и некоммерческий GRID, подключать к системе программные продукты от разных производителей и не требует знания устройства подключаемых систем (только стандартов их взаимодействия). Перечисленные свойства позволяют использовать такую архитектуру для построения интегрированной системы поддержки жизненного цикла воздушного судна. В качестве стандар-



а – при модели «белый» ящик; б – построение GRID на функциональном уровне;
в – интеграция приложений и выделенного GRID-хранилища

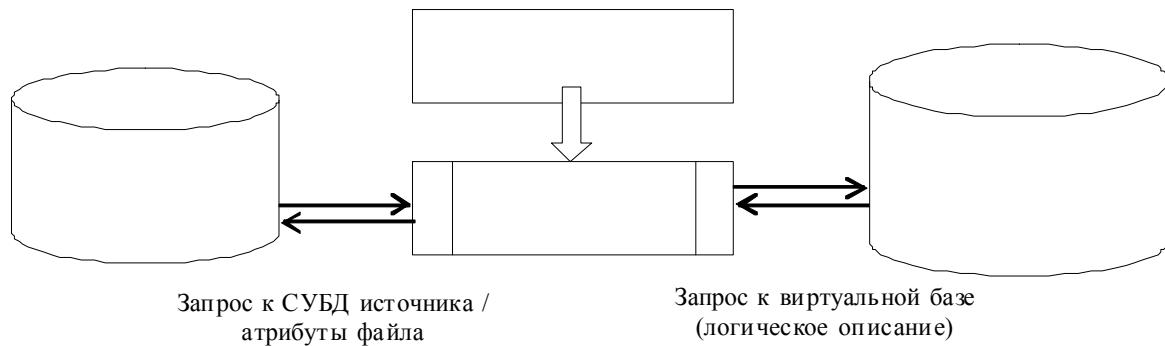


Рис. 2. Схема трансляции запроса

та gridвыбирается OGSA-DAI, так как он позволяет обеспечить распределенное хранение данных в разных СУБД, представить данных в единой виртуальной модели представления данных, и обеспечить поддержку распределенных запросов к хранилищу данных на SQL-подобном языке запросов.

OGSA-DAI может использовать следующие источники данных: MySQL, IBMDB2, Microsoft SQL Server, Oracle, PostgreSQL, eXist, файловая система Unix / Linux, Windows [5].

Технологии ESBи OGSA-DAI являются web-ориентированными и поддерживают модель представления данных в виде xml-документов. Поэтому в рамках интегрированной системы единая виртуальная база данных, включающая и автономные системы (CAPP, CAD, ERP, PDM) и базы данных grid-хранилища (базы могут быть и не реляционными), опирается на xml-модель представления данных. Для поддержки правильности и обеспечения проверки поступающих данных предлагается использовать XSD файлы, содержащие схему модели представления данных.

ОБРАБОТКА ЗАПРОСОВ В ИНТЕГРИРОВАННОЙ СИСТЕМЕ

Для обеспечения взаимодействия между элементами интегрированной системы необходимо, чтобы система обеспечила выполнение запросов к данным виртуальной базы данных. Так как модель представления виртуальной базы данных является XML, логичнее было бы использовать язык представления XQuery, но большинство подключаемых систем (CAPP, CAD, ERP, PDM) используют реляционную модель, и в источниках данных потребуется преобразование запросов. Поэтому представляется целесообразным в качестве языка запросов использовать SQL, а для его поддержки обеспечить согласование XML-модель с моделью «сущность-связь» (сопоставляются теги и их атрибуты атрибутам сущностей).

Запрос к концептуальной модели передается в XML-документе и в источниках преобразовывается в запрос к физическим моделям (рис. 2).

Для данного преобразования используются метаданные, которые описывают взаимосвязь между физической моделью и концептуальной в виде xml-документа:

```
<?xml version="1.0" encoding="utf-8" ?>
<tables><!-- описание таблиц -->
  <table key=“ “ value=""><!-- логическое имя таблицы (key), физическое (value) -->
    <field name=""></field><!-- логическое имя поля (name), физическое -->
    ...
    <field name=""></field>
  </table>
  ...
  <table key=“ “ value="">
    <field name=""></field>
    ...
    <field name=""></field>
  </table>
</tables>
```

Прежде чем попасть в источник данных запрос передается через ESB. По содержанию сообщения, в котором находится запрос, производится маршрутизация сообщений и сервис маршрутизации посыпает сообщение с запросом к поставщикам данных. Поставщики в ответ за запрос посыпают сообщения, которые накапливаются (собираются) в сервисе хранения. Полученные сообщения преобразовываются к необходимому формату единого сообщения в сервисе преобразования и отсыпаются потребителю данных. В зависимости от потребностей поставщика сообщения могут посыпаться в синхронном или асинхронном режиме. Для организации запросов к разнородным источникам данных с помощью ESB предлагаются использовать следующую схему организации запросов (рис. 3).

Grid-хранилище также подключается к ESB, но внутри него запросы обрабатываются средствами OGSA-DAI (рис. 4). Контейнер сервисов OGSA-DAI обеспечивает доступ к ресурсам данных. Чтобы встроить новый сервис данных в контейнер OGSA-DAI, надо реализовать стандартный интерфейс ресурса данных Data Service Resource и набор методов (activities), которые этот

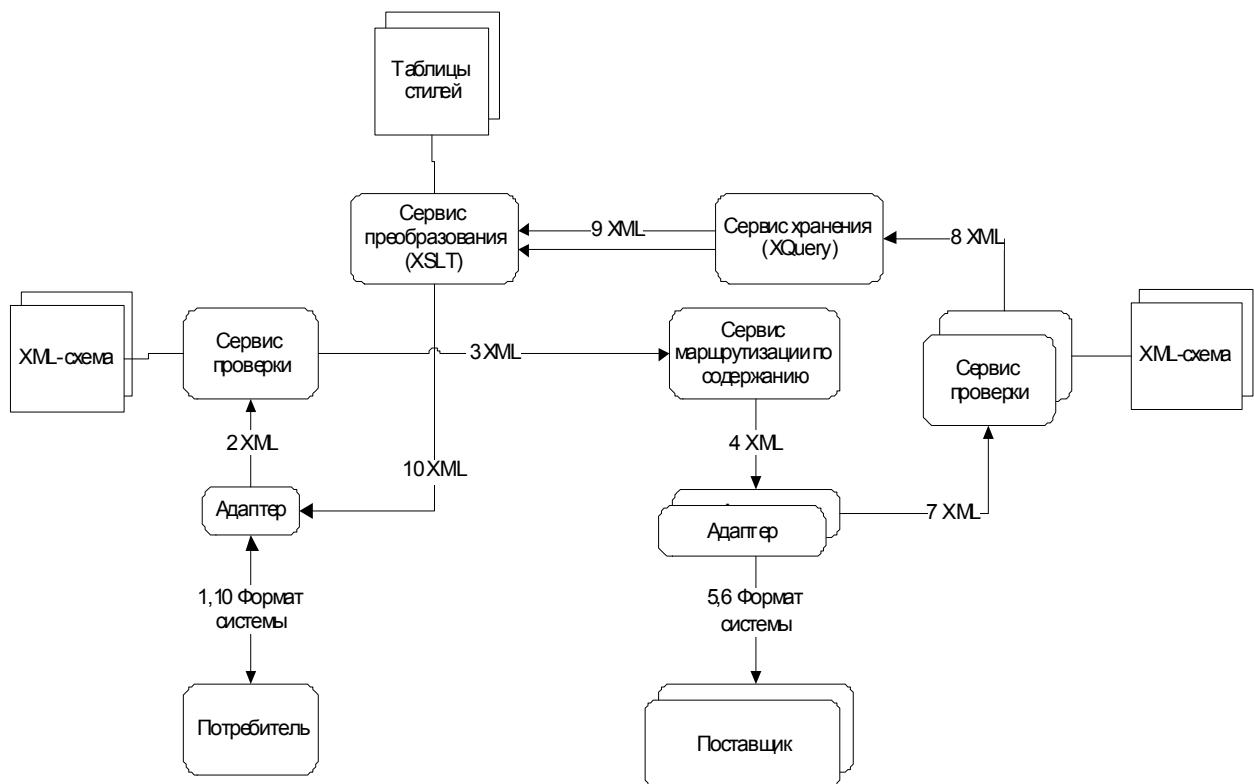


Рис. 3. Схема обработки распределенного запроса

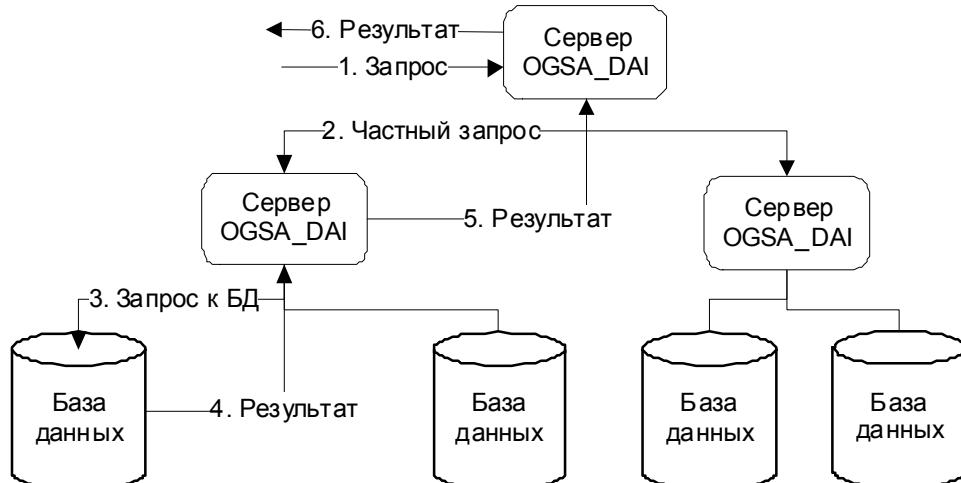


Рис. 4. GRID-архитектура массовой интеграции на базе OGSA-DAI

ресурс будет поддерживать.

В состав комплекса OGSA-DAI входит система OGSA-DQP, реализованная как надстройка над базовой частью и выполняющая интеграцию БД реляционного типа. OGSA-DQP дает более высокий уровень работы с данными: поддерживаются декларативные запросы, соответствующие стандарту SQL-92 с некоторыми ограничениями [7]. Это сервис-ориентированная обработчик запросов, способный распараллеливать запросы к различным ресурсам. OGSA-DQP позволяет работать с данными, полученными из различных источников, как с единой базой БД, эффективно задействует инфраструктуру вычислительных ресурсов, адаптируя методы параллельных БД —

конвейерный параллелизм и параллельное выполнение независимых фрагментов плана. Подобная оптимизация существенно улучшает временные характеристики выполнения запросов [8].

Для осуществления взаимодействия ESB и OGSA необходимо разработать веб-сервисы, выполняемые на сервиснойшине предприятия. Данные веб-сервисы подразделяются на сервисы проверки, преобразования, хранения и маршрутизации поисковых запросов, получаемых от OGSA и конкретных подсистем, которые можно представить в виде обобщенного сервиса обработки и трансляции внешних поисковых запросов (рис. 6). Средствами самой ESB задается возможность настройки последовательности и пра-

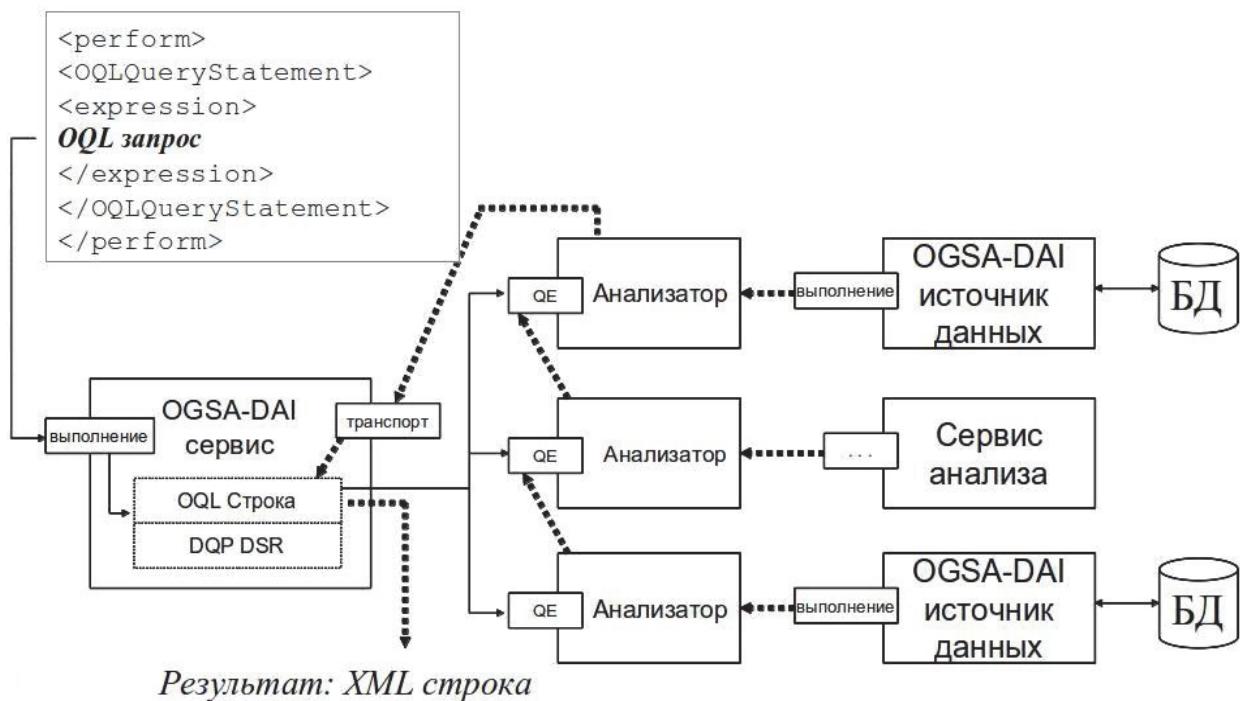


Рис. 5. Схема распределенного запроса в OGSA-DQP

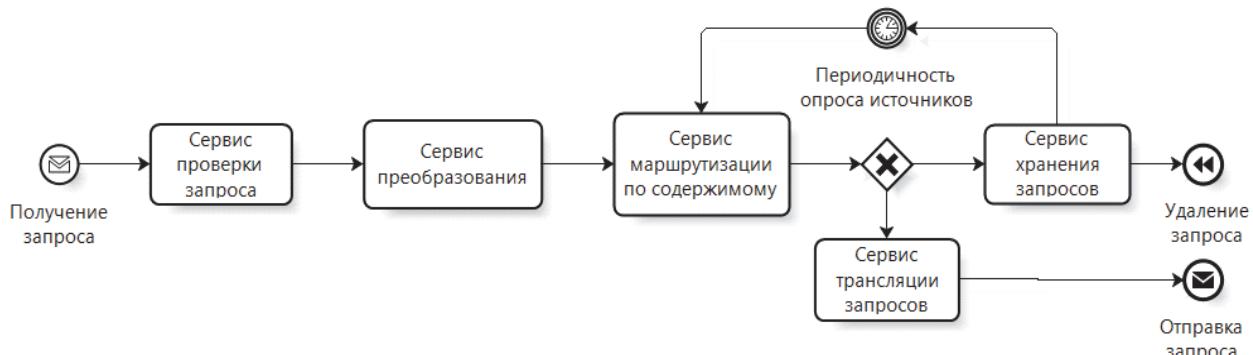


Рис. 6. Бизнес-процесс сервиса обработки и трансляции поисковых запросов

вил выполнения данных сервисов, а также определяются параметры и атрибуты подключаемых источников данных (то есть системы OGSA, PDM, ERP и т.п.).

Доступ к данным сервисам можно осуществить несколькими способами. Во-первых, доступ можно осуществить путем включения клиентской части сервиса маршрутизации по содержанию в состав адаптеров источников данных и адаптера OGSA. Данный способ эффективнее всего работает в том случае, когда адаптер и веб-сервис маршрутизации данных реализованы на одном языке программирования, так как при этом снижается количество методов преобразования данных, а значит итоговая вероятность появления ошибки такого взаимодействия значительно ниже. Так же этот подход наиболее соответствует логике работы с ESB, так как адаптер в данном случае воспринимается как еще один веб-сер-

вис, а значит его можно включить в состав шины (рис. 7), прописав для него соответствующие оркестровку и хореографию. Но, также, стоит учитывать, что такая возможность есть не всегда, так как на сами адаптеры накладываются ограничения по выбору языка программирования средствами самого источника данных, а также по принципу его работы. При этом адаптер источника данных либо целиком хранится и выполняется на сервиснойшине предприятия, либо его серверная часть, что также не всегда является возможным и целесообразным. Данный подход также хорош тем, что позволяет аккумулировать запросы и результаты их выполнения в самом адаптере и передавать их источнику данных в асинхронном режиме, то есть даже в том случае, если каналы связи работают с перебоями.

Во втором случае, адаптеры обращаются к веб-сервису маршрутизации данных по содержа-

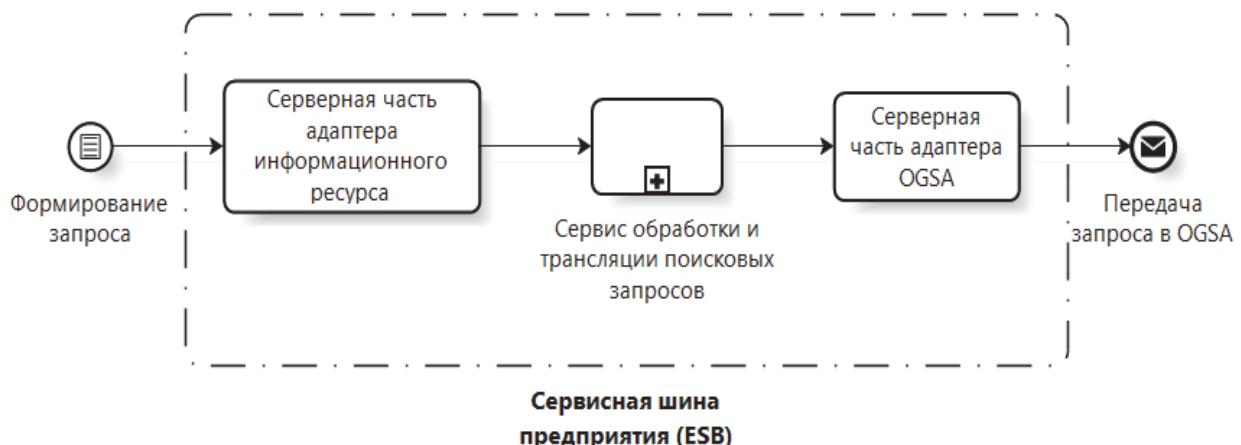


Рис. 7. Подключение OGSAк ESB путем подключения серверной части адаптера OGSA к сервисам ESB

мому путем удаленного вызова его методов через WSDL-файл веб-сервиса. Данный способ позволяет использовать язык программирования адаптера отличный от того, на котором был реализован веб-сервис. При этом расположение самого адаптера может быть как на сервиснойшине предприятия, так и на стороне источника данных (рисунок 8). Данный подход позволяет организовать эффективную работу сервиса с адаптерами в синхронном режиме, но при возникновении перебоев в работе каналов связи между ESB и источниками данных возможны потери данных. Это, в конечном счете, приводит к усложнению самого сервиса маршрутизации данных, возникает необходимость хранения запросов и результатов запроса в самом сервисе, что позволяет эмулировать асинхронную передачу данных. Также сложности возникают в том, как именно сервис возвращает данные адаптеру. Например, если адаптер представляет собой некий исполняемый файл, то ESB должна обязательно знать его точное расположение, что приводит к ограничению масштабируемости интегрированной системы поддержки жизненного цикла изделия. Если же адаптер - это некий плагин, входящий в состав информационного ресурса, то данные этого ресурса будут доступны только тогда, когда он выполняется.

Третий способ предполагает включение клиентской части сервиса маршрутизации в состав службы сервиса маршрутизации, выполняемой на той же платформе, на которой запущена ESB. При этом достигаются преимущества обоих описанных выше методов. Адаптеры, в зависимости от их реализации, при этом могут обмениваться с веб-сервисом данными как напрямую, так и через службу сервиса маршрутизации (рис. 9). При этом запросы и результаты их выполнения передаются источникам как в синхронном, так и в асинхронном режимах, а сами адаптеры могут быть реализованы на любом языке программирования.

В зависимости от выбранного способа осуществляется синхронное, асинхронное или смешанное взаимодействие ESB и источников данных. OGSA, с позиции ESB, является таким же источником данных как и системы PDM, ERP, CAD и т.п. А значит может быть подключена любым из этих способов.

Предлагаемая схема построения интегрированной системы поддержки жизненного цикла воздушного судна (рисунок 10) позволяет использовать одновременно два подхода к интеграции: интеграцию приложений на базе сервис-ориентированной архитектуры и интеграцию данных, обеспечивая масштабируемость и расши-

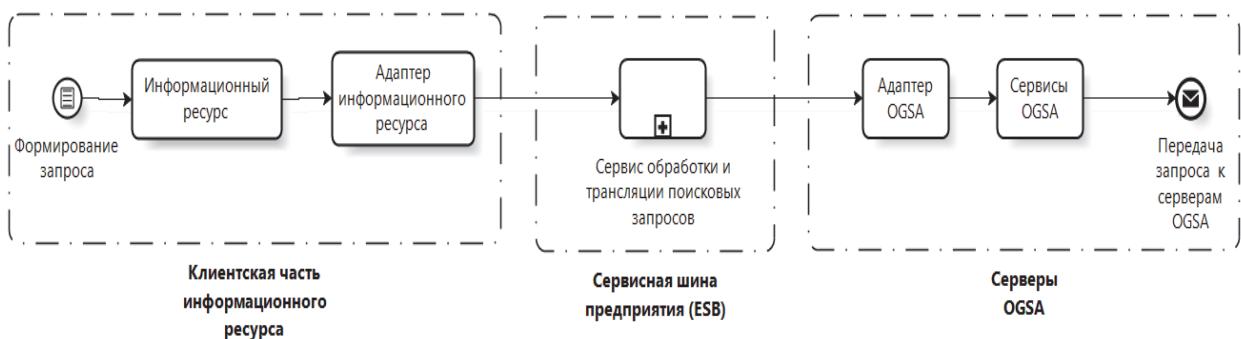


Рис. 8. Подключение OGSAк ESB путем удаленного подключения адаптеров OGSA к WSDLфайлам сервисов ESB

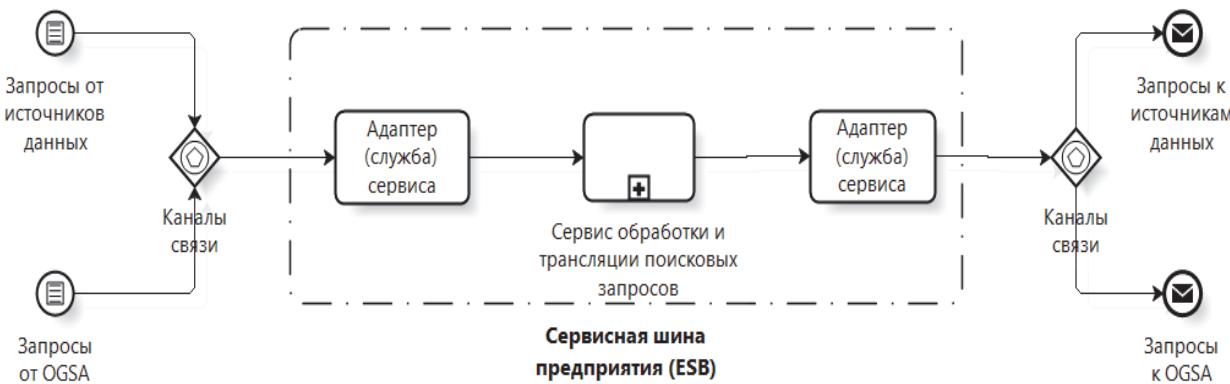


Рис. 9. Подключение OGSA к ESB путем включения в состав ESB службы маршрутизации

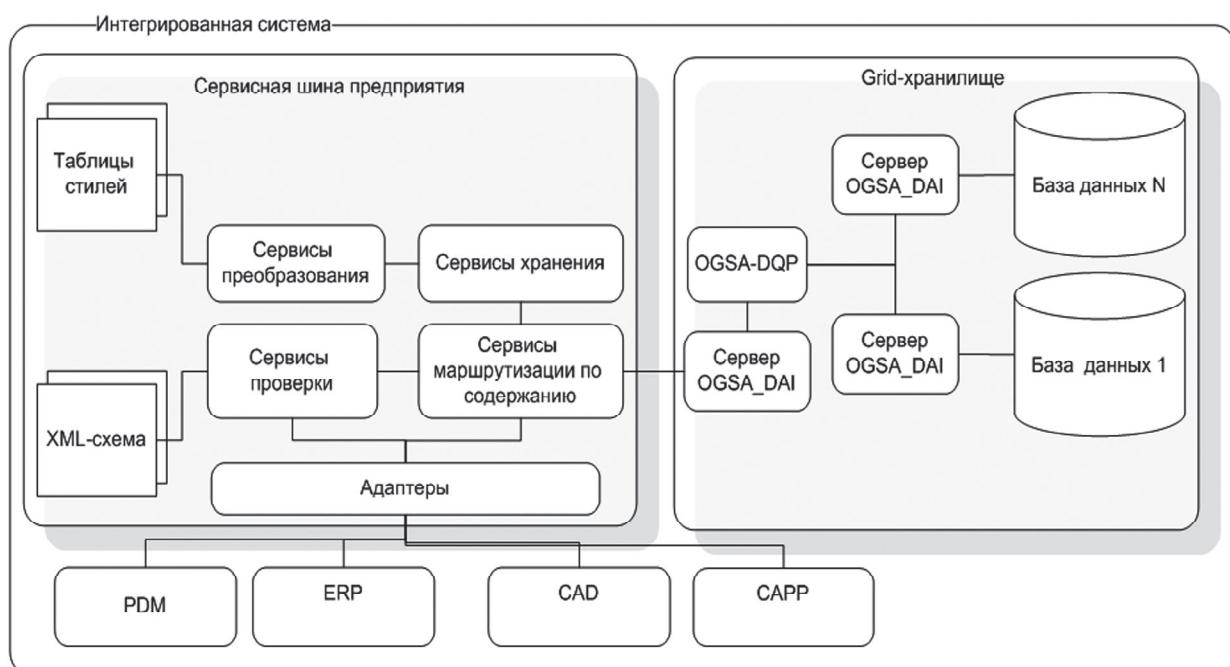


Рис. 10. Гибридная архитектура интегрированной системы поддержки жизненного цикла воздушного судна

ряемость системы, поддержку непрерывности потоков работ (средствами можно выполнять оркестровку и хореографию сервисов).

Интегрированная система обеспечивает повышение качества изготовления воздушного судна, сокращение времени на подготовку производства и выпуск продукции, повышение эффективности послепродажного обслуживания за счет единого информационного пространства конструкторско-технологической подготовки производства и изготовления воздушного судна. Может использоваться на авиастроительных предприятиях.

Экспериментальный образец интегрированной системы поддержки жизненного цикла воздушного судна разработан и протестирован в Ульяновском государственном университете, результаты экспериментальных исследований под-

тверждают применимость на практике предлагаемых решений.

Работа выполнена при частичной финансовой поддержке Минобрнауки России в рамках Государственного контракта № 07.514.11.4131.

СПИСОК ЛИТЕРАТУРЫ

1. Нефедкин Д. Прагматический подход к интеграции приложений // CONNECT! Мир Связи. 2007. №10. С. 114-116.
2. Кусов А.А. Проблемы интеграции корпоративных информационных систем // Управление экономическими системами: электронный научный журнал. 2011. № 28. С. 103-109.
3. Богданов А.В., Станкова Е.Н., Мареев В.В. Сервис-ориентированная архитектура: новые возможности в свете развития GRID-технологий. URL: <http://www.ict.edu.ru/ft/005639/62316e1-st03.pdf> (дата об-

- ращения: 07.10.2012).
4. Тарасов Я. Коммерческий grid // Открытые системы. 2008. № 07. С. 40-43.
 5. ДокументацияOGSA-DAI.Chapter 27. Data resource products. URL: <http://ogsadai.sourceforge.net/documentation/ogsadai3.1/ogsadai3.1-axis/DataResourceProducts.html> (дата обращения: 08.10.2012).
 6. Бежин А.А., Коваль Е.О. Доступ к базам данных с помощью OGSA-DAI и OGSA-DQP. // Материалы XVI Всероссийской научно-методической конференции «Телематика2009». URL: <http://tm.ifmo.ru/> tm2009/db/doc/get_thes.php?id=72 (дата обращения: 08.09.2012).
 7. Дорошенко А.В. Доступ к базам данных с помощью OGSA-DAI и OGSA-DQP // Междисциплинарные исследования в науке и образовании. 2012. №1 Sp; URL: www.es.rae.ru/mino/157-670 (дата обращения: 07.10.2012).
 8. Steven Lynden. Data Access and integration with OGSA-DAI: OGSA-DQP // Materials of «Open Grid Forum 2007». URL: <http://www.ggf.org/GGF17/materials/326/OGSA-DQP-ggf17.ppt> (дата обращения: 05.10.2012).

USING OF GRID AND ESB TECHNOLOGIES FOR BUILDING INTEGRATED SUPPORT LIFE CYCLE SYSTEM OF AIRCRAFT

© 2013 Yu.V. Polyanskov, V.V. Tryascyn, M.M. Dronov

Ulyanovsk State University

The article is devoted to building an integrated system to support the life cycle of the aircraft using the ESB and GRID technologies, namely the choice of system architecture description model of data, schema development process requests.

Keywords: application integration, data integration, service-oriented architecture, enterprise service bus, GRID, Open Grid Service Architecture - Data Access & Integration.

Yuriy Polyanskov, Doctor of Technics, Professor, President of University, Director of Competence Center “Aviation Technology and Aviation Mobility”.

E-mail: president@ulsu.ru

Vitaliy Tryascyn, Graduate Student at the TTS Department, E-mail: sauninvv@gmail.com

Mikhail Dronov, researcher of Researching and Education Laboratory of Center CALS Technologies.

E-mail: m@dronov.net