

ПРОБЛЕМЫ ВЗАИМОДЕЙСТВИЯ И ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В ИНТЕГРИРОВАННОЙ АСУ

УДК 004.415.2

А.А. Смагин, С.В. Липатова, П.И. Смикун, А.С. Мельниченко

МЕТОДИКА ПОСТРОЕНИЯ СПЕЦИАЛИЗИРОВАННЫХ АС НА БАЗЕ SOA

Смагин Алексей Аркадьевич, доктор технических наук, окончил радиотехнический факультет Ульяновского политехнического института. Заведующий кафедрой телекоммуникационных технологий и сетей Ульяновского государственного университета. Имеет свыше 100 статей, изобретения, монографию в области разработки информационных систем различного значения [E-mail: SmaginAA1@mail.ru].

Липатова Светлана Валерьевна, кандидат технических наук, окончила факультет информационных телекоммуникационных технологий. Ассистент кафедры телекоммуникационных технологий и сетей Ульяновского государственного университета. Специализируется в области создания экспертных систем [E-mail: dassége@mail.ru].

Смикун Петр Иванович, окончил факультет автоматики и вычислительной техники Таганрогского радиотехнического института. Заместитель главного конструктора, начальник научно-исследовательского отделения ФНПЦ ОАО «НПО «Марс». Специализируется в области информационных технологий и разработки программного обеспечения для АСУ. [E-mail: smikun@mail.ru].

Мельниченко Анатолий Степанович, окончил факультет автоматики и вычислительной техники Московского института инженеров транспорта. Старший преподаватель кафедры телекоммуникационных технологий и сетей Ульяновского государственного университета. Специализируется в области моделирования процессов технологической подготовки создания программных продуктов и разработки экспертных систем. [E-mail: masulgu@yandex.ru].

Аннотация

В статье рассматривается методика разработки автоматизированных систем (АС) на базе SOA, базирующаяся на принципах системного подхода, обладающая универсальностью, независимостью от программно-аппаратной платформы заказчика и возможностью адаптации к предметной области, и позволяющий обеспечить качество программного продукта АС, соответствующего потребностям пользователя, в заданные сроки и в пределах заранее составленной сметы.

Ключевые слова: сервис-ориентированная архитектура, методика, специализированная автоматизированная система, информационные технологии, жизненный цикл систем.

Abstract

The article deals with a procedure for development of computer-aided SOA-based systems, which results from system approach principles, is characterized by universality, independence from Customer's hardware and software platform and feasible adaptation to subject domain, and ensures quality of computer-aided

system software in accordance with user's requirements within a given time period and a pre-calculated estimate.

Key words: service-oriented architecture, procedure, special-purpose computer-aided system, information technology, system life cycle.

ВВЕДЕНИЕ

Ряд известных зарубежных фирм сегодня предлагают инструменты и методики создания и реорганизации бизнес-процессов на основе SOA (например, платформы IBM WebSphere, BEA Aqualogic, SAP NetWeaver®, ИБК Юпитер, Tibco, Diasoft). Среди названных инструментов и методик существуют две основные, наиболее распространённые платформы и методики построения SOA-систем: J2EE и .NET, множество IDE для разработки web-сервисов, различные реализации сервисных шин предприятия (коммерческие и open source). Вместе с тем, протоколы SOA, несмотря на наличие стандартов, не полностью согласованы, множество сервисов от различных производителей не всегда соответствуют SOA - все это ведет к необходимости выработки подхода, учитывающего специфику и технологическую подготовку предприятия, поставленные задачи и возможности, которые предоставляют современные организационные и информационные технологии SOA.

Разрабатываемая методика должна удовлетворять современным требованиям интеграции информационных процессов и ресурсов предприятия, обеспечивать выполнение общепринятых стандартов SOA и решать поставленные задачи.

Основные принципы методики: движение извне, отталкиваясь от требований пользователя; движение внутрь, отталкиваясь от основных абстракций предметной области; спуск вниз от высокоуровневых моделей к детальному проекту.

Предлагаемая методика позволяет:

- учитывать принципы системного подхода, SOA, интеграции GERA, методологии GERAM, а также сетевые стандарты и стандарты программной инженерии;
- неявно учитывать риск на каждом этапе разработки;
- обеспечивать универсальность, независимость от программно-аппаратной платформы заказчика и возможность адаптации к предметной области;
- обеспечить качество программного продукта, соответствующее потребностям пользователя, в заданные сроки его создания и в пределах заранее составленной сметы;
- использовать жизненный цикл информационных систем (ИС), определяемый стандартом ISO 12207;
- указывать средства для поддержки и выполнения работ по фазам и этапам.

В связи с этим она носит многоэтапный характер, причем на каждом этапе решаются от-

дельные задачи и выделяются шаги, которые детализируют процесс создания SOA-системы в зависимости от подзадач этапа. Каждый этап является точкой опоры построения SOA-системы с заданными свойствами и не позволяет потеряться среди множества средств, методов и рекомендаций специалистов, имеющих опыт разработки и внедрения SOA-систем.

Построенная методика была создана в процессе анализа существующих подходов к проектированию SOA-систем, разработки пилотного проекта данной системы по заказу предприятия, компьютерного моделирования и апробации на множестве бизнес-сервисов.

Ниже приведены состав и содержание этапов предлагаемой методики, которая носит канонический характер и может быть дополнена без существенных изменений при проектировании, разработке и реализации в зависимости от используемой модели жизненного цикла информационной системы.

Последовательность этапов методики

Этап 1. Анализ информационных ресурсов организации.

- 1.1 Определение задач SOA-системы.
- 1.2 Создание информационной модели предприятия.
- 1.3 Определение имеющихся аппаратно-программных и информационных ресурсов.

Этап 2. Моделирование SOA-системы.

- 2.1 Выбор платформы реализации (J2EE / .NET).
- 2.2 Выбор среды разработки (ECLIPSE / NetBeans / Idea / MVS).
- 2.3 Выбор способа реализации реестра (UDDI / SQL).
- 2.4 Создание модели интерфейсов системы.
- 2.5 Создание модели системы бизнес-сервисов (BPEL).
- 2.6 Создание общей модели SOA-системы.

Этап 3. Программная реализация.

- 3.1 Реализация системы web-сервисов (IDE).
- 3.2 Реализация системы бизнес-сервисов (BPEL).
- 3.3 Создание описаний сервисов (WSDL).
- 3.4 Реализация реестра (UDDI / SQL).
- 3.5 Установка и настройка сервисной шины предприятия.
- 3.6 Реализация интерфейсов (IDE).

Этап 4. Внедрение системы.

- 4.1 Развертывание SOA-системы.
- 4.2 Пополнение реестра web-сервисами.
- 4.3 Сопровождение SOA-системы.

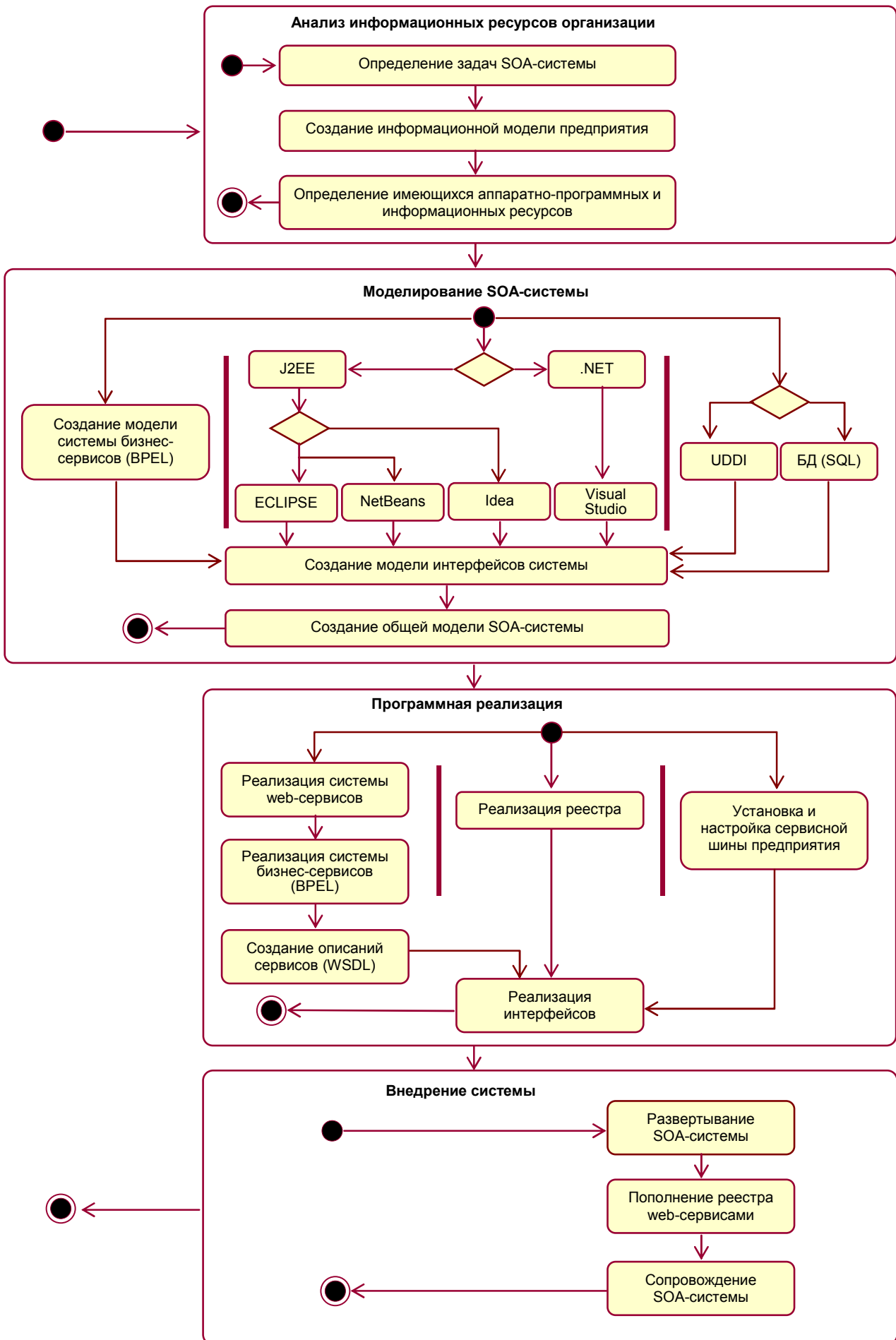


Рис. 1. Алгоритмическое представление последовательности этапов и шагов методики

Предлагаемая многоэтапность, определенность, завершенность предлагаемой методики позволяют представить ее в виде алгоритма, поддерживающего весь жизненный цикл SOA-системы.

Алгоритм отображается в виде диаграммы действий UML (рис. 1), причем некоторые шаги различных этапов можно осуществлять параллельно и взаимосвязанно, например, шаги «Реализация реестра» и «Установка и настройка сервисной шины» на этапе «Программная реализация».

Рассмотрим подробно содержание предлагаемой методики.

Этап 1. Анализ информационных ресурсов организации

На первых этапах, кроме формулирования требований, выделения задач и функций системы, определения необходимых ресурсов, оценки выполнимости проекта в полном объеме, необходимо определить методологию управления проектом, составить план, определить обязанности и сроки (оба этапа носят аналитический характер).

В зависимости от опыта разработчиков, имеющих в наличии информационных и финансовых средств, объема работы, размера проекта и требований, предъявляемых к системе, выбирается та или иная методология ведения информационного проекта (RAD, RUP, ICONIX и т.д.).

Шаг 1. Определение задач SOA-системы.

Задачи определяются заказчиком и логикой работы SOA-приложений. Из них вытекает набор функций системы, они влияют на формирование модели предприятия и системы бизнес-процессов.

Шаг 2. Создание информационной модели предприятия.

Здесь можно использовать методы структурного анализа, представляя задачи и процессы предприятия в виде DFD- и SADT-диаграмм, одним из его результатов станет иерархическая структура программной системы. Кроме того, для отображения производственных процессов рекомендуется использовать UML-диаграммы, которые поддерживаются методологиями управления IT-проектами RUP и ICONIX.

Модель предприятия строится с использованием средств, поддерживающих выбранную методологию и модель жизненного цикла (если RUP или ICONIX, то UML-диаграммы; если RAD, то DFD- и SADT-диаграммы и т.д.).

Шаг 3. Определение имеющихся аппаратно-программных и информационных ресурсов.

Аппаратно-программные средства могут влиять на выбор средств разработки. Эти средства могут быть приобретены или могут быть использованы средства *open source*, в зависимости от имеющихся финансовых возможностей. Ин-

формационные ресурсы - имеющиеся базы данных, хранилища данных, разработанные сервисы и программное обеспечение (ПО) – определяются в процессе построения модели предприятия и создания системы бизнес-сервисов.

Этап 2. Моделирование SOA-системы

Шаг 1. Выбор платформы реализации (J2EE / .NET).

Выбор платформы определяет набор средств реализации SOA-системы. Платформы J2EE и .Net являются конкурирующими технологиями, каждая из которых позволяет создавать web-службы.

На сегодняшний день существуют две основных платформы для создания web-служб. Платформа J2EE (Java 2 Enterprise Edition) позволяет создавать web-службы на основе Java-приложений (Java-класс, EJB-компонент) для любых архитектур и операционных систем. Практически любой класс Java можно объявить как web-службу. Отметим достоинства платформ.

Microsoft .NET:

- множество языков программирования;
- сильные рыночные позиции и мощная маркетинговая команда;
- сложившееся сообщество Windows-программистов;
- законченность решения, обусловленная наличием в продуктовой линейке компании всего спектра серверного ПО, от ОС до СУБД и веб-серверов;
- наличие лишь одного поставщика гарантирует нефрагментированность платформы в будущем;
- платформа технически совершеннее (.NET появился почти на пять лет позже);
- стандартизация.

Sun Java:

- реальная кроссплатформность;
- конкурентность рынка поставщиков;
- сложившееся сообщество Java-программистов.

Выбор платформы определяется уже используемыми на предприятии программными средствами и зависит от опыта разработчиков предприятия.

Согласно данным Gartner Group Research, поляризация между Microsoft .NET и J2EE, с которой в данный момент приходится иметь дело IT-менеджерам, приведет к тому, что эти две платформы в ближайшее время окажутся доминирующими при создании новых корпоративных приложений, при этом 45% всех вновь разрабатываемых проектов будут так или иначе иметь дело с обеими платформами, с вероятностью 70% широко применяться будут обе платформы. При использовании подхода *open source* рекомендуется выбрать J2EE.

Шаг 2. Выбор среды разработки.

Выбор среды разработки зависит от выбранной платформы и от опыта программистов. На сегодняшний день существует множество инструментов разработки. Лидерами среди IDE-средств разработки на Java являются IntelliJ IDEA, ECLIPSE и NetBeans. Набор выполняемых ими функций примерно одинаков. Поэтому при выборе средства нужно ориентироваться на IDE, используемое на предприятии, или выбирать доступное в рамках имеющихся ресурсов.

Определяющее значение имеют следующие критерии:

- стоимость среды разработки;
- открытая лицензия;
- наличие инструментов для работы с проектами SOA;
- наличие документации по работе в IDE с проектами SOA.

Наиболее удовлетворяет этим критериям NetBeans, так как в ней есть инструменты для работы с проектами SOA, встроенный сервер приложений GlassFish и достаточно хорошее сопровождение, например на официальном сайте проекта можно ознакомиться с методическим обеспечением по работе в IDE.

Шаг 3. Выбор способа реализации реестра (UDDI / SQL).

Классическая модель SOA-системы представляет реестр через стандарт UDDI. Третья версия UDDI предусматривает:

- разделение регистра на произвольное множество узлов, представляющих собой физически разделенные и потенциально географически распределенные системы, наполнение которых идентично;
- перенос регистром данных в другой регистр;
- публикацию данных в разных регистрах.

В качестве альтернативного средства построения реестра можно выбрать базы данных. Они являются более предпочтительными с точки зрения создания гибкой системы поиска web-сервисов в реестре и представления информации о сервисах. Кроме того, использование баз данных не требует дополнительного времени по освоению средств разработки.

Шаг 4. Создание модели интерфейсов системы.

Интерфейс может предполагать одну или несколько точек входа. Например, сервис по обработке записей о клиентах поддерживает три услуги: создание записи о клиенте, удаление записи и получение информации по клиенту.

Типичный процесс разработки SOA-приложения начинается с определения интерфейса и дальше идет к стадии развертывания программного обеспечения. Большинство традиционных средств разработки скрывают язык описания интерфейсов, но для методологии SOA эти

описания являются центральными на протяжении всего процесса разработки.

Каждое SOA-приложение может иметь свою собственную внутреннюю архитектуру, требовать для своей поддержки различных знаний и инструментов.

Применение сервисов в различных динамически изменяющихся контекстах предполагает изоляцию сервисов, что облегчает их повторное использование, однако увеличивает сложность и вероятность конфликтов - требуется дополнительная технология интеграции.

SOA нельзя рекомендовать для приложений, использующих асинхронное интерактивное взаимодействие.

Модель интерфейса должна отражать все возможные точки доступа ко всем функциям системы. Расположение компонентов интерфейса адаптируется для решаемых задач и под требования заказчика.

Интерфейс должен предоставлять возможность разграничения прав пользователей. Основными окнами интерфейса должны быть следующие: администрирование сервисов, реестр, поиск сервиса, регистрация сервисов. При этом интерфейс должен поддерживать две группы пользователей: администраторов системы и пользователей. Пользователи имеют право регистрировать, искать и запускать сервисы на выполнение. Администраторы имеют право заводить новых пользователей системы любой группы и создавать новые категории сервисов. Рекомендуется создавать модель интерфейсов с помощью наиболее распространенного на практике языка моделирования — UML.

Шаг 5. Создание модели системы бизнес-сервисов (BPEL).

На данном этапе разрабатывается система сервисов, предоставляющих все необходимые функции для поддержки информационной модели предприятия. Сервис может содержать набор услуг — методов, которые рассматриваются в рамках модели как подфункции. Разработанные web-сервисы используются для формирования бизнес-сервисов, которые отражают процессы, протекающие в информационной модели. Бизнес-сервисы решают задачи, поставленные перед SOA-системой пользователем, и формируются с использованием средств BPEL.

Модель может быть представлена с помощью средств UML и включать в свой состав диаграммы состояний, вариантов использования, диаграммы действий или могут быть представлены с помощью диаграмм потоков данных (DFD).

Шаг 6. Создание общей модели SOA-системы.

Общая модель SOA-системы должна отражать логический и физический уровни реализации и развертывания системы в сети предприятия; определять точки и интерфейс доступа

к web-сервисам, способ построения реестра и системы поиска в реестре (в случае использования UDDI — страницы, БД — системы запросов). Общая модель SOA-системы должна содержать в том или ином виде триаду «потребитель — реестр — поставщик». Модель системы может базироваться на эталонной модели OASIS и должна учитывать существующие стандарты.

Процесс создания модели SOA-системы можно представить в виде следующей цепочки действий: выработка требований к системе — определение функциональности системы в целом и ее элементов — выбор средств реализации — отображение архитектуры и состава системы в виде диаграмм и схем (например, UML), развертывание SOA-системы на аппаратной платформе предприятия.

Этап 3. Программная реализация

На этапе реализации выбирается одно из средств программирования. Выбор конкретного языка программирования зависит от требований к системе, программно-аппаратного окружения, предпочтений заказчика и программиста, выбранной концепции программирования.

Шаг 1. Реализация системы web-сервисов.

Web-сервисы могут реализовываться независимо друг от друга различными программистами с использованием разных средств, хотя рекомендуется использовать одну, выбранную IDE. Они должны соответствовать требованиям, предъявляемым к web-сервисам SOA, т.е. они должны поддерживать следующие стандарты и протоколы: WSDL, SOAP.

Шаг 2. Реализация системы бизнес-сервисов (BPEL).

Современные IDE, поддерживающие стандарты SOA, предоставляют удобные средства (например, графический интерфейс моделирования BPEL-процесса) для создания бизнес-процессов. У разработчиков web-сервисов имеются все необходимые средства для реализации различных бизнес-сервисов.

Шаг 3. Создание описаний сервисов (WSDL).

Существует множество средств генерации WSDL-описаний, в том числе встроенных в IDE, которые и рекомендуется использовать. Следует следить за обновлением WSDL-описаний при изменении сервисов. В некоторых случаях WSDL-описание следует корректировать вручную.

Шаг 4. Реализация реестра UDDI/SQL.

При выборе базы данных как средства организации реестра следует использовать уже установленный web-сервер, если он поддерживает выбранную СУБД, или установить дополнительный и на нем развернуть базу с заданной схемой, отражающей все элементы UDDI, и дополнительными полями, позволяющими оптимизировать поиск и эффективно описать сервисы с

точки зрения поставленных перед специализированной АС задач. Рекомендуется использовать СУБД, для которых предоставлены средства доступа в выбранной IDE, и объекты в языке программирования, которые используются для написания интерфейса системы.

Шаг 5. Установка и настройка сервисной шины предприятия.

Сервисную шину предприятия можно разрабатывать с нуля, а можно использовать готовые сервисные шины от различных разработчиков, например, использовать шины *open source*. Необходимо выбрать шину в соответствии с потребностями и имеющимися ресурсами. Готовую сервисную шину нужно установить на сервере и адаптировать к поставленным в ТЗ задачам.

Шаг 6. Реализация интерфейсов (IDE).

Интерфейс SOA-системы разрабатывается в соответствии с заданными задачами, отражающими различные варианты доступа к функциям системы, реализуется как web-интерфейс. Для его создания могут использоваться различные языки программирования: Java, PHP, ASP и т.д.

Этап 4. Внедрение системы

Шаг 1. Развертывание SOA-системы.

Развертывание системы производится в соответствии с общей моделью SOA-системы, логикой использования SOA-системы, предусматривает установку системы в сети предприятия, серверов приложений и реестра, а также интерфейса системы на рабочие места и/или размещение его как web-сервиса.

Шаг 2. Пополнение реестра web-сервисами.

Пополнение реестра производится через разработанную и установленную SOA-систему (через интерфейс пользователя). Web-сервисы могут быть разработаны как на предприятии, так и сторонним поставщиком сервиса.

Шаг 3. Сопровождение SOA-системы.

Выбор средства сопровождения и поддержки напрямую зависит от требований заказчика и выделенных ресурсов. Мероприятия по поддержке и сопровождению могут ограничиваться разработкой документации и справочной системы или включать в себя комплекс мероприятий, осуществляющих обучение разработчиков сервисов и пользователей системы и круглосуточную службу поддержки. Основными направлениями работы на этапе являются создание средств обучения для пользователей системы, организация обучающих мероприятий, создание службы поддержки системы и т.д.

ЗАКЛЮЧЕНИЕ

Предложенная методика наиболее реально отображает разработку SOA-системы как системы интеграции предприятия, позволяет

учитывать принципы SOA, интеграции GERA, методологий GERAM, IBM, Sonic, а также сетевые стандарты и стандарты программной инженерии. Методика, план и временной график по всем этапам проекта упорядочивают ход конструирования SOA-системы, позволяет неявно учитывать риск на каждом этапе разработки, базируются на принципах системного подхода, обладают универсальностью, независимостью от программно-аппаратной платформы заказчика и возможностью адаптации к предметной области, позволяют обеспечить качество программного продукта, соответствующего потребностям пользователя, в заданные сроки и в пределах заранее составленной сметы.

Полученные результаты подтвердили корректность выбранных этапов и составляющих их шагов.

В заключение следует отметить, что многовариантность и многоплановость возможных действий на отдельных шагах методики существенно не влияют на качество получаемого результата, а решают частично проблему ограниченных средств разработки, проектирования и внедрения SOA-систем.

СПИСОК ЛИТЕРАТУРЫ

1. Binildas C. A. Service Oriented Java Business Integration. - Birmingham-Mumbai: Packt Publishing, 2008. - 414 с.
2. Salter D., Jennings F. Building SOA-Based Composite Applications Using NetBeans IDE 6. - Birmingham: Packt Publishing, 2008. - 288 с.
3. Биберштейн Н., Боуз С., Джонс К., Фаммант М., Ша Р. Компас в мире сервис-ориентированной архитектуры (SOA): ценность для бизнеса, планирование и план развития предприятия. - М.: Кудиц-пресс, 2007. - 256 с.
4. Драница А. Java vs .NET // Компьютера.- 2003.- № 41 (516).
5. Елманова Н. Платформы для создания корпоративных решений: Microsoft .NET или J2EE? // КомпьютерПресс. - 2003. - № 1.
6. Ларман К. Применение UML и шаблонов проектирования, введение в объектно-ориентированный анализ, проектирование и унифицированный процесс RUP / Пер. с англ. - М.: Вильямс, 2004.- 624 с.
7. Ньюкомер Э. Веб-сервисы: XML, WSDL, SOAP и UDDI. - СПб.: Питер, 2003. - 256 с.
8. Орлов С.А. Технологии разработки программного обеспечения. - СПб.:Питер, 2002. - 464 с.
9. Поллис Г., Огастин Л., Лоу К., Мадхар Д. Разработка программных проектов на основе Rational Unified Process (RUP). - М.: ООО «Бином-Пресс», 2005. - 256 с.
10. Стефан Бергстрем, Лотта Роберг. Rational Unified Process — путь к успеху. Руководство по внедрению RUP. - М: КУДИЦ-Образ, 2004. - 256 с.
11. Хабибуллин И.Ш. Разработка web-служб средствами Java. - СПб.: БХВ-Петербург, 2003. - 409 с.