



ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ ИНТЕГРАЦИИ РАЗНОРОДНЫХ ФУНКЦИОНАЛЬНЫХ СЕРВИСОВ

Никитюк Владимир Анатольевич

*Харьковский национальный университет радиоэлектроники, кафедра информационных
управляющих систем*

INFORMATION TECHNOLOGY INTEGRATION OF HETEROGENEOUS FUNCTIONAL SERVICES

Nikityuk Vladimir Anatoljevich

Kharkov national university of Radioelectronics, department of information management systems

Abstract

An approach to the problem of integrating heterogeneous services into a single integrated information system is considered. It is proposed theoretical-categorical model of information technology, which establishes a set of transformations descriptions of services in the course of solving the problem of integration.

Keywords: *service, information system, integration, IDEF3, use case, information technology, UDDI.*

Введение

В настоящее время основным описанием работ, выполняемых в рамках жизненного цикла информационной системы (ИС), следует считать стандарт ISO/IEC 15288:2002 «System engineering – System life cycle processes» и его последующие модификации. Анализ основных процессов, выполняемых в рамках жизненного цикла ИС согласно данному стандарту, показывает, что задача интеграции сервисов как отдельных элементов ИС управления предприятиями и организациями решается в рамках типового процесса комплексирования. Для описания процесса интеграции разнородных сервисов в рамках ИС, основанной на сервис-ориентированной архитектуре (SOA), как частного случая процесса комплексирования системы, прежде всего, следует уточнить цель выполнения этого процесса. Такой целью следует считать получение единой и цельной картины корпоративных бизнес-данных, имеющих семантическую и прагматическую ценность для пользователей различных уровней.

Предложенное определение цели выполнения процесса интеграции разнородных сервисов в рамках ИС, основанной на SOA, требует выделения из всего множества сервисов в первую

очередь тех сервисов, которые обеспечивают реализацию отдельных функций таких ИС. Подобные сервисы предлагается здесь и в дальнейшем называть «функциональные сервисы».

IDEF3-модель процесса интеграции разнородных функциональных сервисов в рамках ИС, основанной на SOA, имеет вид, показанный на рис. 1 [3].

Исходя из данной модели необходимо рассматривать информационную технологию (ИТ) интеграции разнородных функциональных сервисов в рамках ИС, основанной на SOA, как инструментальное средство, используемое для решения задачи оптимизации множества актуальных сервисов после добавления, модификации или удаления одного или нескольких функциональных сервисов. Исходными данными этой ИТ следует считать описания целевой функции процесса интеграции, описания ограничений данной целевой функции, описания исходного множества актуальных сервисов и подмножества сервисов, доступных для интеграции. Результатом выполнения данной ИТ следует считать описания измененного множества актуальных сервисов, которые в ходе принятия решения о степени удовлетворения

персонала объекта автоматизации результатами интеграции могут быть трансформированы либо в записи в реестре функциональных сервисов,

либо в перечень несоответствий между отдельными функциональными сервисами.

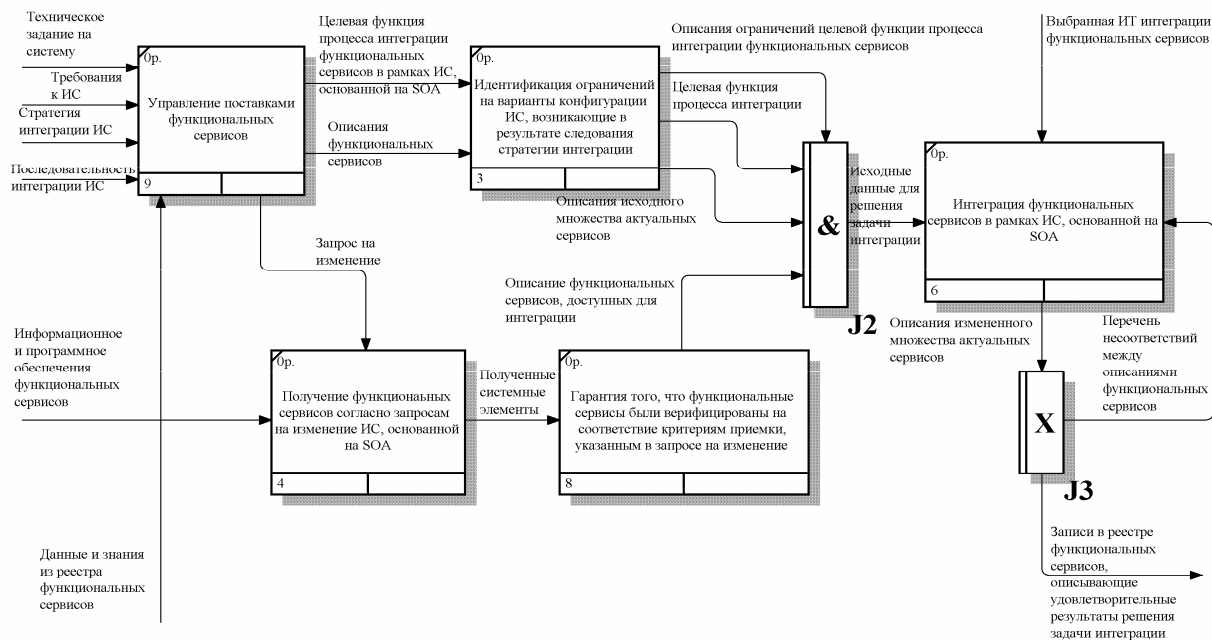


Рис. 1. IDEF3-модель процесса интеграции разнородных функциональных сервисов в рамках информационной системы, основанной на сервис-ориентированной архитектуре

Материалы и методы

Как следует из [1], ИТ – это приемы, способы и методы применения средств вычислительной техники при выполнении функций сбора, хранения, обработки, передачи и использования данных. Поэтому любую ИТ можно рассматривать как последовательность этапов выполнения некоего множества работ. Для описания результатов выполнения этих работ используются множества структур данных. Собственно результаты выполнения этих работ представляют собой значения элементов соответствующих структур данных, полученные непосредственно в ходе выполнения этих работ.

Основываясь на предложенной IDEF3-модели процесса интеграции разнородных функциональных сервисов в рамках ИС, основанной на SOA, ИТ, автоматизирующую выполнение данного процесса, стоит рассматривать как последовательность таких этапов.

Этап 0 (предварительный). Принятие решения об инициации процесса интеграции разнородных функциональных сервисов в рамках ИС, основанной на SOA, на основе одного или нескольких событий из следующего перечня:

а) поступление в ИС одного или нескольких новых функциональных сервисов или же модифицированных версий уже эксплуатируемых функциональных сервисов;

б) принятие руководством предприятия, на котором эксплуатируется ИС, основанная на SOA, решения о расширении или изменении множества актуальных сервисов данной ИС;

в) принятие руководством предприятия, на котором эксплуатируется ИС, основанная на SOA, решения об изменении стратегии интеграции функциональных сервисов и, как следствие, об изменении вида целевой функции и ограничений задачи оптимизации измененного множества актуальных сервисов после выполнения операций добавления, модификации или удаления отдельных функциональных сервисов.

В случае принятия отрицательного решения об инициации процесса завершить работу ИТ. В случае принятия положительного решения об инициации процесса перейти к Этапу 1.

Этап 1. Формирование модели реестра множества актуальных сервисов ИС, основанной на SOA, на момент начала выполнения процесса интеграции разнородных функциональных

сервисов.

Этап 2. Формирование моделей отдельных сервисов, поступивших в ИС, основанную на SOA, на момент начала выполнения процесса интеграции разнородных функциональных сервисов.

Этап 3. Формирование описания новой или корректировка описания существующей целевой функции задачи оптимизации реестра измененного множества актуальных сервисов ИС, основанной на SOA.

Этап 4. Формирование описаний новых или корректировка описаний существующих ограничений задачи оптимизации реестра измененного множества актуальных сервисов ИС, основанной на SOA.

Этап 5. Формирование модели реестра измененного множества актуальных сервисов ИС, основанной на SOA, по результатам выполнения множества операций интеграции [2] над моделью реестра множества актуальных сервисов ИС, основанной на SOA, сформированной в результате выполнения Этапа 1, и моделями отдельных сервисов, поступивших в ИС, основанную на SOA, сформированными в результате выполнения Этапа 2.

Этап 6. Решение задачи оптимизации измененного множества актуальных сервисов ИС, основанной на SOA, с учетом результатов выполнения Этапов 3 – 5.

Этап 6. Формирование и отображение результатов решения задачи оптимизации измененного множества актуальных сервисов ИС, основанной на SOA, в терминах, понятных пользователю.

Данное представление ИТ интеграции разнородных функциональных сервисов в рамках ИС, основанной на SOA, позволяет представить концептуальную модель данной технологии в виде диаграммы прецедентов, выполненной с использованием языка моделирования UML (рис. 2).

Основываясь на предложенной концептуальной модели и рассмотренных выше особенностях представления промежуточных и основных результатов работы ИТ, формализованное описание ИТ интеграции разнородных функциональных сервисов в рамках ИС, основанной на SOA, можно представить как совокупность связанных отображениями следующих моделей:

а) модель реестра множества актуальных сервисов ИС, основанной на SOA;

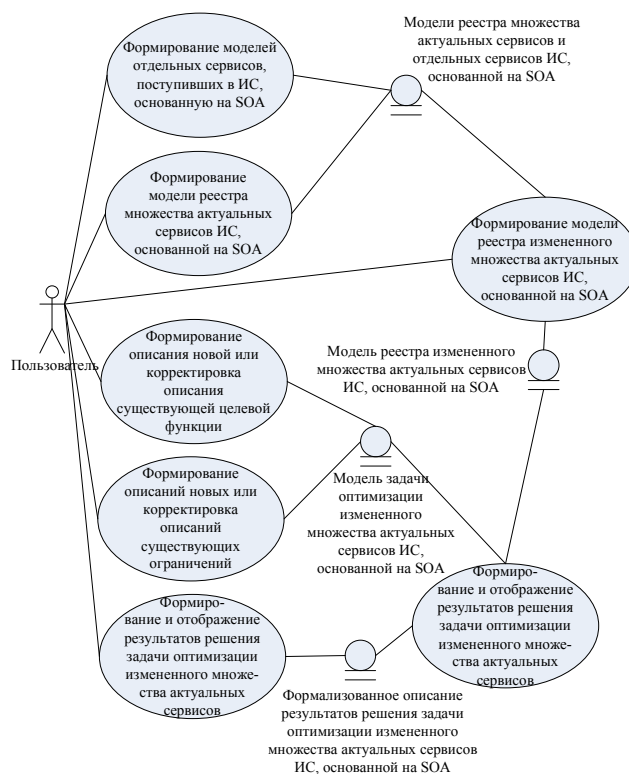


Рис. 2. Диаграмма прецедентов информационной технологии интеграции разнородных функциональных сервисов в рамках информационной системы, основанной на сервис-ориентированной архитектуре

б) модели отдельных сервисов, поступивших в ИС, основанную на SOA;

в) модель реестра измененного множества актуальных сервисов ИС, основанной на SOA;

г) модель задачи оптимизации реестра измененного множества актуальных сервисов ИС, основанной на SOA;

д) модель результатов решения задачи оптимизации измененного множества актуальных сервисов ИС, основанной на SOA.

В общем случае такую модель можно представить как категорию L_{IT} следующего вида:

$$L_{IT} = [L_{M_R}, L_{M_{Ser}}, L_{M_{R'}}, L_{M_{Opt}}, L_{M_{Res}}, F_{L_{M_R}^{L_{M_{R'}}}, L_{M_{Ser}}^{L_{M_{Opt}}}, F_{L_{M_{R'}}^{L_{M_{Opt}}}, F_{L_{M_{Res}}^{L_{M_{Opt}}}], \quad (1)$$

где L_{M_R} - теоретико-категорное описание модели реестра множества актуальных сервисов ИС, основанной на SOA; $L_{M_{Ser}}$ - теоретико-категорное описание моделей отдельных сервисов, поступивших в ИС, основанную на SOA; $L_{M_{R'}}$ - теоретико-категорное описание

модели реестра измененного множества актуальных сервисов ИС, основанной на SOA; $L_{M_{Opt}}$ - теоретико-категорное описание модели задачи оптимизации реестра измененного множества актуальных сервисов ИС, основанной на SOA; $L_{M_{Res}}$ - теоретико-категорное описание модели результатов решения задачи оптимизации измененного множества актуальных сервисов ИС, основанной на SOA; $F_{L_{M_{R'}}}^{L_{M_{Res}}}$ - отображение теоретико-категорных описаний модели реестра множества актуальных сервисов и моделей отдельных сервисов в теоретико-категорное описание модели реестра измененного множества актуальных сервисов ИС, основанной на SOA; $F_{L_{M_{Opt}}}^{L_{M_{R'}}$ - отображение теоретико-категорного описания модели реестра измененного множества актуальных сервисов в теоретико-категорное описание модели задачи оптимизации реестра измененного множества актуальных сервисов ИС, основанной на SOA; $F_{L_{M_{Res}}}^{L_{M_{Opt}}}$ - отображение теоретико-категорного описания модели задачи оптимизации реестра измененного множества актуальных сервисов в теоретико-категорное описание модели результатов решения задачи оптимизации измененного множества актуальных сервисов ИС, основанной на SOA.

Результаты и обсуждение

Изложенное выше представление ИТ интеграции разнородных функциональных сервисов позволяет определить структуру модели реестра функциональных сервисов ИС, основанной на SOA. В отличие от традиционных представлений реестра сервисов как базы описаний, разрабатываемая модель реестра функциональных сервисов должна иметь два четко выделенных и взаимосвязанных компонента [4]:

а) компонент модели реестра функциональных сервисов, описывающий хранение сведений о семантическом представлении множества актуальных сервисов и функциональных сервисов, доступных для интеграции;

б) компонент модели реестра функциональных сервисов, описывающий хранение сведений о синтаксических описаниях метаданных множества актуальных сервисов и

функциональных сервисов, доступных для интеграции.

Первый компонент модели реестра функциональных сервисов предлагается рассматривать в общем случае как базу знаний о семантических представлениях отдельных функциональных сервисов с точки зрения конкретных предметных областей и объектов автоматизации. Второй компонент модели реестра функциональных сервисов предлагается рассматривать в общем случае как базу данных, в которой хранятся описания метаданных отдельных функциональных сервисов. При этом все изменения множества актуальных сервисов по результатам выполнения операций интеграции, выполненные в рамках первого компонента модели реестра функциональных сервисов, должны автоматически отображаться в содержание и структуру второго компонента этой же модели.

Таким образом, схема структуры модели реестра функциональных сервисов и взаимосвязей этой модели с другими моделями будет иметь вид, показанный на рис. 3 [4].



Рис. 3. Схема структуры модели реестра функциональных сервисов и взаимосвязей этой модели с другими моделями

В основу описания реестра функциональных сервисов положен стандарт UDDI. Поэтому структура модели реестра функциональных сервисов L_{rfs} будет определяться принятой в стандарте UDDI структурой модели данных используемой для описания сервисов. Тогда в общем случае модель реестра функциональных сервисов L_{rfs} будет иметь следующий вид [4]:

$$L_{rfs} = [L_{bE}^{uddi}, L_{bS}^{uddi}, L_{bT}^{uddi}, L_{tM}^{uddi}, L_{pA}^{uddi}, F_{L_{bS}^{uddi}}^{L_{bE}^{uddi}}, F_{L_{bE}^{uddi}}^{L_{bS}^{uddi}}, F_{L_{bT}^{uddi}}^{L_{bS}^{uddi}}, F_{L_{bT}^{uddi}}^{L_{bE}^{uddi}}, F_{L_{pA}^{uddi}}^{L_{bS}^{uddi}}, F_{L_{pA}^{uddi}}^{L_{bE}^{uddi}}], \quad (2)$$



где L_{bE}^{uddi} - подкатегория, описывающая элемент `uddi:businessEntity` модели данных стандарта UDDI; L_{bS}^{uddi} - подкатегория, описывающая элемент `uddi:businessService` модели данных стандарта UDDI; L_{bT}^{uddi} - подкатегория, описывающая элемент `uddi:bindingTemplates` модели данных стандарта UDDI; L_{tM}^{uddi} - подкатегория, описывающая элемент `uddi:tModels` модели данных стандарта UDDI; L_{pA}^{uddi} - подкатегория, описывающая дополнительный элемент `uddi:publisherAssertion` модели данных стандарта UDDI; $F_{L_{bE}^{uddi}}^{L_{bS}^{uddi}}$ - отображение, устанавливающее иерархию подчинения вида «один – ко многим» для элементов категорий L_{bE}^{uddi} и L_{bS}^{uddi} соответственно; $F_{L_{bE}^{uddi}}^{L_{bS}^{uddi}}$ - отображение, устанавливающее иерархию подчинения вида «один – ко многим» для элементов категорий L_{bS}^{uddi} и L_{bE}^{uddi} соответственно; $F_{L_{bS}^{uddi}}^{L_{bT}^{uddi}}$ - отображение, устанавливающее иерархию подчинения вида «один – ко многим» для элементов категорий L_{bS}^{uddi} и L_{bT}^{uddi} соответственно; $F_{L_{bS}^{uddi}}^{L_{bT}^{uddi}}$ - отображение, устанавливающее иерархию подчинения вида «один – ко многим» для элементов категорий L_{bT}^{uddi} и L_{bS}^{uddi} соответственно; $F_{L_{bS}^{uddi}}^{L_{pA}^{uddi}}$ - отображение, устанавливающее иерархию подчинения вида «один – ко многим» для элементов категорий L_{bS}^{uddi} и L_{pA}^{uddi} соответственно.

Элемент `uddi:businessEntity` модели данных стандарта UDDI представляет собой верхний уровень структуры данных, который содержит описательную информацию об организации и ее бизнесе [5]. Кроме того, этот элемент объединяет в себе информацию, содержащуюся в элементах `uddi:businessService` и `uddi:bindingTemplates`. Поэтому в общем случае подкатегория L_{bE}^{uddi} будет иметь следующий вид:

$$L_{bE}^{uddi} = [businessKey, bE_discoveryURLs, bE_name, bE_description, bE_contacts, bE_identifierBag, bE_categoryBag, bE_signature, h_{bE_discoveryURLs}^{businessKey}, h_{bE_name}^{businessKey}, h_{bE_description}^{businessKey}, h_{bE_contacts}^{businessKey}, h_{bE_identifierBag}^{businessKey}, h_{bE_categoryBag}^{businessKey}, h_{bE_signature}^{businessKey}], \quad (3)$$

где *businessKey* - идентификатор экземпляра элемента `uddi:businessEntity`, публикуемого в реестре функциональных сервисов; *bE_discoveryURLs* - агрегат данных, представляющий собой список указателей на адреса документов или механизмов, открывающих сервисы, используемые для данного экземпляра элемента `uddi:businessEntity`; *bE_name* - агрегат данных, представляющий собой список имен данного экземпляра элемента `uddi:businessEntity`; *bE_description* - агрегат данных, представляющий собой список описаний данного экземпляра элемента `uddi:businessEntity`; *bE_contacts* - агрегат данных, представляющий собой список сведений для осуществления контактов либо с конкретными людьми, имеющими отношение к данному экземпляру элемента `uddi:businessEntity`, либо же с людьми, должности которых указывают на непосредственное участие в данном экземпляре элемента `uddi:businessEntity`; *bE_identifierBag* - агрегат данных, представляющий собой список идентификаторов данного экземпляра элемента `uddi:businessEntity`, отличных от *businessKey* и используемых в других системах, в которых может требоваться идентификация бизнес-сущностей; *bE_categoryBag* - агрегат данных, представляющий собой совокупность структур данных, позволяющих классифицировать данный экземпляр элемента `uddi:businessEntity` с помощью существующих систем классификации аспектов бизнеса; *bE_signature* - агрегат данных, представляющий собой список цифровых подписей, удостоверяющих подлинность и достоверность данного экземпляра элемента `uddi:businessEntity`; $h_{bE_discoveryURLs}^{businessKey}$ - коконус морфизмов, устанавливающий иерархическую связь вида



«один – ко многим» между родительским экземпляром объекта $businessKey$ подкатегории L_{bE}^{uddi} и дочерними экземплярами объекта $bE_discoveryURLs$ подкатегории L_{bE}^{uddi} , мощность этой связи находится в диапазоне $[0; \infty]$; $h_{bE_name}^{businessKey}$ - коконус морфизмов, устанавливающий иерархическую связь вида «один – ко многим» между родительским экземпляром объекта $businessKey$ подкатегории L_{bE}^{uddi} и дочерними экземплярами объекта bE_name подкатегории L_{bE}^{uddi} , мощность этой связи находится в диапазоне $[1; \infty]$; $h_{bE_description}^{businessKey}$ - коконус морфизмов, устанавливающий иерархическую связь вида «один – ко многим» между родительским экземпляром объекта $businessKey$ подкатегории L_{bE}^{uddi} и дочерними экземплярами объекта $bE_description$ подкатегории L_{bE}^{uddi} , мощность этой связи находится в диапазоне $[0; \infty]$; $h_{bE_contacts}^{businessKey}$ - коконус морфизмов, устанавливающий иерархическую связь вида «один – ко многим» между родительским экземпляром объекта $businessKey$ подкатегории L_{bE}^{uddi} и дочерними экземплярами объекта $bE_contacts$ подкатегории L_{bE}^{uddi} , мощность этой связи находится в диапазоне $[0; \infty]$; $h_{bE_identifierBag}^{businessKey}$ - коконус морфизмов, устанавливающий иерархическую связь вида «один – ко многим» между родительским экземпляром объекта $businessKey$ подкатегории L_{bE}^{uddi} и дочерними экземплярами объекта $bE_identifierBag$ подкатегории L_{bE}^{uddi} , мощность этой связи находится в диапазоне $[0; \infty]$; $h_{bE_categoryBag}^{businessKey}$ - коконус морфизмов, устанавливающий иерархическую связь вида «один – ко многим» между родительским экземпляром объекта $businessKey$ подкатегории L_{bE}^{uddi} и дочерними экземплярами объекта $bE_categoryBag$ подкатегории L_{bE}^{uddi} , мощность этой связи находится в диапазоне $[0; \infty]$;

$h_{bE_signature}^{businessKey}$ - коконус морфизмов, устанавливающий иерархическую связь вида «один – ко многим» между родительским экземпляром объекта $businessKey$ подкатегории L_{bE}^{uddi} и дочерними экземплярами объекта $bE_signature$ подкатегории L_{bE}^{uddi} , мощность этой связи находится в диапазоне $[0; \infty]$.

Элемент `uddi:businessService` модели данных стандарта UDDI представляет собой логическое описание сервисов, предоставляемых экземпляром модели элемента `uddi:businessEntity` [5]. Кроме того, этот элемент объединяет в себе информацию, содержащуюся в элементе `uddi:bindingTemplates`. Поэтому в общем случае подкатегория L_{bS}^{uddi} будет иметь следующий вид:

$$L_{bS}^{uddi} = [serviceKey, bS_name, bS_description, bS_categoryBag, bS_signature, h_{bS_name}^{serviceKey}, h_{bS_description}^{serviceKey}, h_{bS_categoryBag}^{serviceKey}, h_{bS_signature}^{serviceKey}], \quad (4)$$

где $serviceKey$ - идентификатор экземпляра элемента `uddi:businessService`, публикуемого в реестре функциональных сервисов; bS_name - агрегат данных, представляющий собой список имен данного экземпляра элемента `uddi:businessService`; $bS_description$ - агрегат данных, представляющий собой список описаний данного экземпляра элемента `uddi:businessService`; $bS_categoryBag$ - агрегат данных, представляющий собой совокупность структур данных, позволяющих классифицировать данный экземпляр элемента `uddi:businessService` с помощью существующих систем классификации аспектов бизнеса; $bS_signature$ - агрегат данных, представляющий собой список цифровых подписей, удостоверяющих подлинность и достоверность данного экземпляра элемента `uddi:businessService`; $h_{bS_name}^{serviceKey}$ - коконус морфизмов, устанавливающий иерархическую связь вида «один – ко многим» между родительским экземпляром объекта $serviceKey$ подкатегории L_{bS}^{uddi} и дочерними экземплярами объекта bS_name подкатегории L_{bS}^{uddi} , мощность этой связи находится в диапазоне $[0; \infty]$; $h_{bS_description}^{serviceKey}$ - коконус морфизмов,



устанавливающий иерархическую связь вида «один – ко многим» между родительским экземпляром объекта *serviceKey* подкатегории L_{bS}^{uddi} и дочерними экземплярами объекта *bS_description* подкатегории L_{bS}^{uddi} , мощность этой связи находится в диапазоне $[0; \infty]$; $h_{bS_categoryBag}^{serviceKey}$ - коконус морфизмов, устанавливающий иерархическую связь вида «один – ко многим» между родительским экземпляром объекта *serviceKey* подкатегории L_{bS}^{uddi} и дочерними экземплярами объекта *bS_categoryBag* подкатегории L_{bS}^{uddi} , мощность этой связи находится в диапазоне $[0; \infty]$; $h_{bS_signature}^{serviceKey}$ - коконус морфизмов, устанавливающий иерархическую связь вида «один – ко многим» между родительским экземпляром объекта *serviceKey* подкатегории L_{bS}^{uddi} и дочерними экземплярами объекта *bS_signature* подкатегории L_{bS}^{uddi} , мощность этой связи находится в диапазоне $[0; \infty]$.

Элемент *uddi:bindingTemplates* модели данных стандарта UDDI представляет собой описание основных технических параметров сервиса, опубликованного в реестре [5]. Элемент *uddi:tModel* модели данных стандарта UDDI представляет собой описание структур, спецификаций, понятий, ролей и архитектур сервиса, позволяющее сравнивать сервисы для определения их совместимости или же устанавливать ссылки на пространство имен сущностей и атрибутов данных, используемых различными сервисами [5]. Эти элементы не имеют прямого отношения к особенностям предметной области, определяющим семантику сервисов. Поэтому в работе в дальнейшем структура и содержание этих элементов считаются полностью соответствующими стандарту UDDI v. 3.0.2 и подробно рассматриваться не будут.

Элемент *uddi:publisherAssertion* модели данных стандарта UDDI представляет собой описание структур метаданных и спецификаций связи различных экземпляров элемента *uddi:businessEntity* между собой [5]. Поэтому в общем случае подкатегория L_{pA}^{uddi} будет иметь следующий вид:

$$L_{pA}^{uddi} = [pa_fromKey, pa_toKey, (\varphi_{pA_keyed Reference})], \quad (5)$$

где *pa_fromKey* - идентификатор экземпляра элемента *uddi:businessEntity*, от которого передаются данные, полученные в результате выполнения сервиса как экземпляра элемента *uddi:businessService*; *pa_toKey* - идентификатор экземпляра элемента *uddi:businessEntity*, которому передаются данные, полученные в результате выполнения сервиса как экземпляра элемента *uddi:businessService*; $(\varphi_{pA_keyed Reference})$ - множество морфизмов, каждый элемент которого описывает связь между двумя экземплярами элемента *uddi:businessEntity* следующим образом:

$$\varphi_{pA_keyed Reference} = (tModelKey, KeyName, KeyValue), \quad (6)$$

tModelKey - идентификатор параметра, значение которого передается между двумя экземплярами элемента *uddi:businessEntity* сервисом, описанным как конкретный экземпляр элемента *uddi:businessService*; *KeyName* - наименование параметра, значение которого передается между двумя экземплярами элемента *uddi:businessEntity* сервисом, описанным как конкретный экземпляр элемента *uddi:businessService*; *KeyValue* - значение параметра, которое передается между двумя экземплярами элемента *uddi:businessEntity* сервисом, описанным как конкретный экземпляр элемента *uddi:businessService*.

Отображение $F_{L_{bS}^{uddi}}^{L_{bE}^{uddi}}$ в общем случае будет иметь следующий вид:

$$F_{L_{bS}^{uddi}}^{L_{bE}^{uddi}} : businessKey \rightarrow (serviceKey) \quad (7)$$

и представляет собой описание множества идентификаторов сервисов как экземпляров элемента *uddi:businessService*, поставленных в соответствие идентификатору одного экземпляра элемента *uddi:businessEntity*. Физический смысл данного отображения заключается в описании множества сервисов, эксплуатируемых в рамках некоего бизнес-процесса или же отдельной работы этого бизнес-процесса.

Отображение $F_{L_{bE}^{uddi}}^{L_{pA}^{uddi}}$ в общем случае будет иметь следующий вид:



$$F_{L_{bE}}^{L_{bS}^{uddi}} : serviceKey \rightarrow (businessKey) \quad (8)$$

и представляет собой описание множества идентификаторов различных экземпляров элемента `uddi:businessEntity`, поставленных в соответствие идентификатору одного экземпляра элемента `uddi:businessService`. Физический смысл данного отображения заключается в описании множества бизнес-процессов или же отдельных работ этих процессов, в рамках которых эксплуатируется конкретный сервис. Подобное повторное использование сервиса может быть целесообразно как с точки зрения управления различными предприятиями, объединенными в одну корпорацию [5], так и с точки зрения управления различными бизнес-процессами одного предприятия, в которых используются одни и те же ресурсы или же механизмы.

Отображение $F_{L_{pA}}^{L_{bS}^{uddi}}$ в общем случае будет иметь следующий вид:

$$F_{L_{pA}}^{L_{bS}^{uddi}} : (businessKey, serviceKey) \rightarrow \begin{cases} pa_fromKey; \\ pa_toKey; \end{cases} \quad (9)$$

причем выбор между двумя атрибутами элемента `uddi:publisherAssertion`, в один из которых будут записаны идентификаторы экземпляров элементов `uddi:businessEntity` и `uddi:businessService`, определяется человеком, выполняющим это отображение.

Отображения $F_{L_{bT}}^{L_{bS}^{uddi}}$ и $F_{L_{tM}}^{L_{bT}^{uddi}}$ устанавливают связи между элементами `uddi:businessService` и `uddi:bindingTemplates`, а также `uddi:bindingTemplates` и `uddi:tModel` соответственно. Эти отображения, как и соответствующие элементы, не имеют прямого отношения к особенностям предметной области, определяющим семантику сервисов. Поэтому в работе в дальнейшем структура и содержание этих отображений считаются полностью соответствующими стандарту UDDI v. 3.0.2 и подробно рассматриваться не будут.

Из выражений (2) – (9) следует, что в стандарте UDDI хранение структурированных описаний семантики отдельных сервисов не предусмотрено. Поэтому предлагается усовершенствовать модель данных UDDI, используя разработанный ранее комплекс

моделей семантического представления функционального сервиса. Основой для модификации следует избрать агрегаты `bE_description` и `bS_description`. Данные агрегаты, как следует из стандарта UDDI [5], не являются обязательными при описании экземпляров элементов `uddi:businessEntity` и `uddi:businessService`, а их значения могут определяться создателями сервисов или же специалистами по внедрению и эксплуатации сервисов в рамках ИС, основанной на SOA.

Заключение

Предлагаемое формализованное описание ИТ интеграции разнородных функциональных сервисов в рамках ИС, основанной на SOA, определяет основные элементы, разработка и апробация которых позволят значительно снизить технические риски, а также риски процессов и процедур управления проектом, возникающие в ходе разработки данной ИТ. Данные риски определяются требованиями и особенностями разрабатываемой ИТ, а также применяемыми при управлении проектом разработки данной ИТ методами, моделями и методиками. Снижение указанных рисков позволит значительно сократить возможные затраты на создание рассматриваемой ИТ.

Литература

- [1] ГОСТ 34.003–90 Автоматизированные системы. Требования и определения [Электронный ресурс]. – Введ. 01.01.1992. – Режим доступа : [www. URL: http://www.vashdom.ru/gost/34.003-90/](http://www.vashdom.ru/gost/34.003-90/).
- [2] Евланов, М.В. Формализованное описание условий интеграции ИТ-сервисов в информационную систему управления предприятием [Текст] / М.В. Евланов, Н.В. Васильцова, В.А. Никитюк // Вісник Академії митної служби України. Серія «Технічні науки». – 2011. – № 2 (46). – С. 87-96.
- [3] Никитюк, В.А. Усовершенствование модели процессов интеграции разнородных функциональных сервисов [Текст] / В.А. Никитюк // Системний аналіз. Інформатика. Управління (САІУ-2013): матеріали ІV Міжнародної науково-практичної конференції (м. Запоріжжя, 13-16 березня 2013 р.). – Запоріжжя: КПУ, 2013. – С. 180-182.
- [4] Никитюк, В.А. Модель реестра функциональных сервисов информационной системы [Текст] / В.А. Никитюк // Збірник тез доповідей науково-практичної конференції «Застосування інформаційних технологій у підготовці та діяльності сил охорони правопорядку». – Харків: Академія Внутрішніх військ МВС України, 2013. – С. 90-92.
- [5] UDDI Version 3.0.2. [Электронный ресурс] // Сайт OASIS (Advancing open standards for the information society). – Режим доступа: http://uddi.org/pubs/uddi_v3.htm#_Toc85907977 – Заголовок с экрана.