

universitetu «Harkivs'kij politehnicnij institut», V. 57, 271–277. 7. Zinovkin, V. V., Kulinich, Je. M. (2009). Modeling of process waste in the production technology of aircrete. Quality strategy for industry and education: Intern. Conf., 6-13 June 2009. Varna, Bulgaria, V. 2, 176 – 179. 8. Dudnikov, E. G. (1987). Automatic control in the chemical industry. Moscow: Himija. 9. Karpuhina, N. V., Pastuhova, K. M., Sviridov, P. A. (2003). Artificial intelligence techniques in problems of operational control and optimization of complex technological systems. Control problems, 3, 21-24. 10. Daffin, R., Paterson, Je., Zener, K. (1972). Geometric programming. Moscow: Mir.

Надійшла (received) 18.12.2014

УДК 044.03; 681.518:061

М. В. ЕВЛАНОВ, канд. техн. наук, доц., ХНУРЭ, Харьков;
Н. В. ВАСИЛЬЦОВА, канд. техн. наук, доц., ХНУРЭ, Харьков;
В. А. НИКИТЮК, аспирант, ХНУРЭ, Харьков

МОДЕЛИ ОПЕРАЦИЙ ИНТЕГРАЦИИ ФУНКЦИОНАЛЬНЫХ СЕРВИСОВ В ИНФОРМАЦИОННОЙ СИСТЕМЕ УПРАВЛЕНИЯ ПРЕДПРИЯТИЕМ

Проведен анализ основных направлений развития сервис-ориентированных информационных систем. Разработаны математические модели операций интеграции описаний отдельных функциональных сервисов в реестр актуальных сервисов информационной системы. Предложен унифицированный метод интеграции отдельных функциональных сервисов, основным преимуществом которого является возможность досрочного прекращения выполнения операции интеграции, если обнаруживается ошибка, вызванная несоответствием описаний сервиса описаниям предметной области

Ключевые слова: информационная система, сервис-ориентированная архитектура, функциональный сервис, интеграция, реестр, онтология

Введение. В соответствии с современными тенденциями в развитии информационных систем (ИС) управления предприятиями и их процессами одним из часто реализуемых подходов к построению ИС корпоративного масштаба является сервис-ориентированная архитектура (Service-Oriented Architecture, SOA). Эта архитектура предполагает построение ИС из набора гетерогенных слабосвязанных компонентов (сервисов) [1]. Поэтому обязательным условием построения и внедрения архитектуры SOA-ИС является использование единой инфраструктуры описания сервисов (репозитория сервисов), разрешенных протоколов доступа и обмена сообщениями, форматов сообщений. Эта инфраструктура образует так называемую интеграционную шину (ИШ) (Enterprise Service Bus – ESB), являющуюся одним из центральных компонентов системы. Она устанавливает единые правила публикации сервисов, управления и информационного взаимодействия между приложениями различных систем, входящих в состав интегрированной системы. Это упрощает управление приложениями и их поддержку, а также снижает риск фрагментации приложений и процессов [2].

Однако на практике руководство предприятия, для которого создается SOA-ИС, склонно забывать о необходимости эффективного управления данными и сервисами SOA-ИС до тех пор, пока не становится слишком поздно [3]. Такое опоздание приводит к неоправданным затратам финансовых и других ресурсов на эксплуатацию отдельных сервисов SOA-ИС предприятия без возможности окупить эти затраты за счет эффекта от эксплуатации SOA-ИС в целом.

© М. В. ЕВЛАНОВ, Н. В. ВАСИЛЬЦОВА, В. А. НИКИТЮК, 2014

Другой, не менее важной проблемой информатизации предприятий, является разнообразие поставщиков и решений на рынке SOA-ИС. Такое разнообразие приводит к тому, что подобные ИС на целом ряде предприятий формируются из разнородных сервисов. Вследствие этого возникает интерес к решению проблемы повышения эффективности использования SOA-ИС в основной деятельности предприятия и к оптимизации затрат расходуемых при этом ресурсов различного рода. Эта задачу не следует считать элементарной или же типовой – есть большое количество практических примеров того, как работы по информатизации предприятия не дают желаемого эффекта или же приводят к излишним затратам. Сказанное позволяет считать проблему управления разнородными сервисами SOA-ИС в значительной степени нерешенной.

Основным направлением решения проблемы управления разнородными сервисами SOA-ИС следует признать модификацию модели реестра сервисов SOA-ИС. В основе современных подходов к описанию реестров сервисов различных производителей находится модель, зафиксированная в стандарте UDDI v. 3.0.2, в которой не предусмотрено хранение структурированных описаний семантики отдельных сервисов [4]. Формализованное описание этой модели приведено одним из авторов в [5, 6]. Поэтому для модификации этой модели в [6] был предложен комплекс моделей семантического представления сервиса, рассмотренный в [7]. Основой для модификации следует избрать агрегаты *bE_description* и *bS_description*. Данные агрегаты, как следует из стандарта UDDI [4], не являются обязательными, а их значения могут определяться создателями сервисов или же специалистами по внедрению и эксплуатации сервисов в рамках SOA-ИС.

Цель работы. Целью работы является разработка моделей операций интеграции функциональных сервисов, основанных на изложенных в [6, 8] результатах модификации модели реестра сервисов SOA-ИС. Под функциональным сервисом следует понимать сервисы, которые обеспечивают реализацию отдельных функций SOA-ИС управления предприятиями и их процессами.

Модели операций интеграции функциональных сервисов. Изложенные в [6, 8] результаты модификации реестра функциональных сервисов и его агрегатов *bE_description* и *bS_description* позволяют значительно модифицировать операции интеграции функциональных сервисов, рассмотренные авторами в [9]. Такая модификация обеспечит целенаправленное изменение содержимого реестра множества актуальных сервисов SOA-ИС путем интеграции в него описаний только тех функциональных сервисов, которые будут необходимы для решения функциональных задач данной SOA-ИС.

Как показано в [9], основные варианты возможных операций интеграции функциональных сервисов в реестр множества актуальных сервисов SOA-ИС сводятся к следующим операциям:

- а) добавление в реестр множества актуальных сервисов SOA-ИС описания нового функционального сервиса, которое никак не связано с описаниями множества актуальных сервисов, хранимого в реестре;
- б) добавление в реестр множества актуальных сервисов SOA-ИС описания

нового функционального сервиса, который дополняет и/или развивает функциональные возможности одного или нескольких функциональных сервисов, чьи описания уже хранятся в реестре;

в) исключение из реестра множества актуальных сервисов SOA-ИС описания функционального сервиса, чьи функции обработки хранимых данных оказались невостребованными ни одним пользователем SOA-ИС или каким-либо другим сервисом.

Условиями реализации первого варианта интеграции являются условия отсутствия в реестре множества актуальных сервисов SOA-ИС следующих описаний:

а) описаний понятий и терминов предметной области добавляемого функционального сервиса;

б) описаний онтологий добавляемого функционального сервиса;

в) традиционных описаний метаданных добавляемого функционального сервиса.

Первое из указанных условий для нового функционального сервиса N_z можно описать следующим образом:

$$bE_description(M_R) \cap bE_description(N_z) = \emptyset, \quad (1)$$

где $bE_description(M_R)$ - модель агрегата $bE_description$ реестра множества актуальных сервисов SOA-ИС, рассмотренная в [6]; $bE_description(N_z)$ - модель агрегата $bE_description$ нового функционального сервиса N_z , добавляемого в реестр, рассмотренная в [6].

Второе из указанных условий для нового функционального сервиса N_z можно описать следующим образом:

$$bS_description(M_R) \cap bS_description(N_z) = \emptyset, \quad (2)$$

где $bS_description(M_R)$ - модель агрегата $bS_description$ реестра множества актуальных сервисов SOA-ИС, рассмотренная в [6]; $bS_description(N_z)$ - модель агрегата $bS_description$ нового функционального сервиса N_z , добавляемого в реестр, рассмотренная в [6].

Третье из указанных условий для нового функционального сервиса N_z предполагает проверку уникальности традиционных описаний агрегатов этого сервиса. Поэтому в соответствии с предложенной в [5, 6] теоретико-категорной моделью реестра функциональных сервисов введем теоретико-множественные описания хранимого в реестре множества актуальных сервисов SOA-ИС в виде выражения

$$RFS''(M_R) = \left[\begin{array}{l} (bE(M_R) - bE_description(M_R)), \\ (bS(M_R) - bS_description(M_R)), \\ bT(M_R), tM(M_R), pA(M_R) \end{array} \right] \quad (3)$$

и нового функционального сервиса N_z в виде выражения

$$RFS''(N_z) = \begin{bmatrix} (bE(N_z) - bE_description(N_z))_b \\ (bS(N_z) - bS_description(N_z))_b \\ bT(N_z), tM(N_z), pA(N_z) \end{bmatrix}, \quad (4)$$

где $(bE(M_R) - bE_description(M_R))$ - описание элемента `uddi:businessEntity` хранимого в реестре множества актуальных сервисов SOA-ИС за исключением агрегата `bE_description`, имеющее вид

$$(bE(M_R) - bE_description(M_R)) = \begin{bmatrix} businessKey(M_R), \\ bE_discoveryURLs(M_R), \\ bE_name(M_R), \\ bE_description(M_R), \\ bE_contacts(M_R), \\ bE_identifierBag(M_R), \\ bE_categoryBag(M_R), \\ bE_signature(M_R) \end{bmatrix}; \quad (5)$$

$(bS(M_R) - bS_description(M_R))$ - описание элемента `uddi:businessService` хранимого в реестре множества актуальных сервисов SOA-ИС за исключением агрегата `bS_description`, имеющее вид

$$(bS(M_R) - bS_description(M_R)) = \begin{bmatrix} serviceKey(M_R), \\ bS_name(M_R), \\ bS_description(M_R), \\ bS_categoryBag(M_R), \\ bS_signature(M_R) \end{bmatrix}; \quad (6)$$

$bT(M_R)$ - описание элемента `uddi:bindingTemplates` хранимого в реестре множества актуальных сервисов SOA-ИС; $tM(M_R)$ - описание элемента `uddi:tModels` хранимого в реестре множества актуальных сервисов SOA-ИС; $pA(M_R)$ - описание элемента `uddi:publisherAssertion` хранимого в реестре множества актуальных сервисов SOA-ИС; $(bE(N_z) - bE_description(N_z))$ - описание элемента `uddi:businessEntity` нового функционального сервиса N_z за исключением агрегата `bE_description`, имеющее вид

$$(bE(N_z) - bE_description(N_z)) = \begin{bmatrix} businessKey(N_z), \\ bE_discoveryURLs(N_z), \\ bE_name(N_z), \\ bE_description(N_z), \\ bE_contacts(N_z), \\ bE_identifierBag(N_z), \\ bE_categoryBag(N_z), \\ bE_signature(N_z) \end{bmatrix}; \quad (7)$$

$(bS(N_z) - bS_description(N_z))$ - описание элемента `uddi:businessService` нового функционального сервиса N_z за исключением агрегата `bS_description`, имеющее вид

$$(bS(N_z) - bS_description(N_z)) = \begin{bmatrix} serviceKey(N_z), \\ bS_name(N_z), \\ bS_description(N_z), \\ bS_categoryBag(N_z), \\ bS_signature(N_z) \end{bmatrix}; \quad (8)$$

$bT(N_z)$ - описание элемента `uddi:bindingTemplates` нового функционального сервиса N_z ; $tM(N_z)$ - описание элемента `uddi:tModels` нового функционального сервиса N_z ; $pA(N_z)$ - описание элемента `uddi:publisherAssertion` нового функционального сервиса N_z .

Тогда третье из указанных условий для нового функционального сервиса N_z можно описать следующим образом:

$$RFS''(M_R) \cap RFS''(N_z) = \emptyset. \quad (9)$$

По аналогии с описаниями $RFS''(M_R)$ и $RFS''(N_z)$ введем полное теоретико-множественное описание хранимого в реестре множества актуальных сервисов SOA-ИС $RFS(M_R)$ и полное теоретико-множественное описание нового функционального сервиса N_z $RFS(N_z)$. От рассмотренных выше описаний они будут отличаться добавлением агрегата $bE_description$ в выражения (5) и (7), а также агрегата $bS_description$ в выражения (6) и (8) соответственно. Тогда операцию добавления нового функционального сервиса N_z , реализующую первый вариант интеграции, можно представить продукционным правилом следующего вида:

$$\begin{aligned} & \text{если } bE_description(M_R) \cap bE_description(N_z) = \emptyset \\ & \text{и } bS_description(M_R) \cap bS_description(N_z) = \emptyset \\ & \text{и } RFS''(M_R) \cap RFS''(N_z) = \emptyset \\ & \text{то } RFS(M_R) + RFS(N_z) \end{aligned} \quad (10)$$

Как показано выше, операцией реализации второго варианта интеграции является добавление в реестр множества актуальных сервисов SOA-ИС описания нового функционального сервиса, который дополняет и/или развивает функциональные возможности одного или нескольких функциональных сервисов, чьи описания уже хранятся в реестре.

В [9] показано, что выполнение данного варианта интеграции возможно только при выполнении условия, согласно которому в процессе будущей эксплуатации добавляемого/модифицируемого функционального сервиса N_z должно происходить хотя бы одно из следующих событий:

а) сервис N_z принимает данные, формируемые эксплуатируемым сервисом s_i , и использует эти данные в процессе своего функционирования;

б) сервис N_z в результате своего функционирования формирует данные, которые впоследствии передает эксплуатируемому сервису s_i ; полученные таким образом данные сервис s_i использует в процессе своего функционирования.

Таким образом, реализация второго варианта интеграции будет возможной при выполнении следующих условий:

а) выполнение описанного выше условия для описаний понятий и терминов предметной области добавляемого/модифицируемого функционального сервиса и хотя бы одного функционального сервиса, описание которого уже хранится в реестре;

б) выполнение описанного выше условия для описаний онтологий добавляемого/модифицируемого функционального сервиса и хотя бы одного функционального сервиса, описание которого уже хранится в реестре;

в) выполнение описанного выше условия для традиционных описаний метаданных добавляемого/модифицируемого функционального сервиса и хотя бы одного функционального сервиса, описание которого уже хранится в реестре.

Для формализованного описания первого из указанных условий введем дополнительные формализованные описания агрегатов $bE_description$ принимающей и передающей частей добавляемого/модифицируемого функционального сервиса N_z и реестра функциональных сервисов. Эти описания по аналогии с [6] будут иметь вид:

$$bE_description_r(N_z) = \left(\bigcup_j \langle Id_C_{zj}^r, Name_C_{zj}^r, Description_C_{zj}^r, (R_{C_{zk}^r}^{C_{zj}^r}), (\varphi_{C_{zm}^r}^{C_{zj}^r}) \rangle \right), \quad (11)$$

$$bE_description_t(N_z) = \left(\bigcup_p \langle Id_C_{zp}^t, Name_C_{zp}^t, Description_C_{zp}^t, (R_{C_{zik}^t}^{C_{zp}^t}), (\varphi_{C_{zm}^t}^{C_{zp}^t}) \rangle \right), \quad (12)$$

$$bE_description_r(M_R) = \bigcup_i \left(\bigcup_j \langle Id_C_{ij}^r, Name_C_{ij}^r, Description_C_{ij}^r, (R_{C_{ik}^r}^{C_{ij}^r}), (\varphi_{C_{im}^r}^{C_{ij}^r}) \rangle \right), \quad (13)$$

$$i = \overline{1, n},$$

$$bE_description_t(M_R) = \bigcup_i \left(\bigcup_p \langle Id_C_{ip}^t, Name_C_{ip}^t, Description_C_{ip}^t, (R_{C_{ik}^t}^{C_{ip}^t}), (\varphi_{C_{im}^t}^{C_{ip}^t}) \rangle \right), \quad (14)$$

$$i = \overline{1, n},$$

где $bE_description_r(N_z)$ - обозначение части агрегата $bE_description$, описывающего принимающую часть добавляемого/модифицируемого функционального сервиса N_z ; $bE_description_t(N_z)$ - обозначение части агрегата $bE_description$, описывающего передающую часть добавляемого/модифицируемого функционального сервиса N_z ; $bE_description_r(M_R)$ - обозначение части агрегата $bE_description$, описывающего принимающие части хранимого в реестре множества актуальных сервисов SOA-ИС; $bE_description_t(M_R)$ - обозначение части агрегата

$bE_description$, описывающего передающие части хранимого в реестре множества актуальных сервисов SOA-ИС.

Тогда первое из условий выполнения операции добавления/модификации функционального сервиса можно записать в виде выражения

$$\begin{aligned} & [bE_description_r(N_z) \cap bE_description_t(M_R)] \cup \\ & \cup [bE_description_t(N_z) \cap bE_description_r(M_R)] \neq \emptyset \end{aligned} \quad (15)$$

Для формализованного описания второго из указанных условий введем дополнительные формализованные описания агрегатов $bS_description$ принимающей и передающей частей добавляемого/модифицируемого функционального сервиса N_z и реестра функциональных сервисов. Эти описания по аналогии с [6] будут иметь вид:

$$bS_description_r(N_z) = \left(\bigcup_j \langle n_{zj}^r, \{ \langle n_{zj}^{xr}, T_{zj}^{xr} \rangle \}, (R_{O_{zk}^r}^{O_{zj}^r}), (\Phi_{O_{zm}^r}^{O_{zj}^r}) \rangle \right), \quad (16)$$

$$bS_description_t(N_z) = \left(\bigcup_p \langle n_{zp}^t, \{ \langle n_{zp}^{xt}, T_{zp}^{xt} \rangle \}, (R_{O_{zk}^t}^{O_{zp}^t}), (\Phi_{O_{zm}^t}^{O_{zp}^t}) \rangle \right), \quad (17)$$

$$bS_description_r(M_R) = \bigcup_i \left(\left(\bigcup_j \langle n_{ij}^r, \{ \langle n_{ij}^{xr}, T_{ij}^{xr} \rangle \}, (R_{O_{ik}^r}^{O_{ij}^r}), (\Phi_{O_{im}^r}^{O_{ij}^r}) \rangle \right) \right), \quad (18)$$

$i = \overline{1, n},$

$$bS_description_t(M_R) = \bigcup_i \left(\left(\bigcup_p \langle n_{ij}^t, \{ \langle n_{ij}^{xt}, T_{ij}^{xt} \rangle \}, (R_{O_{ik}^t}^{O_{ij}^t}), (\Phi_{O_{im}^t}^{O_{ij}^t}) \rangle \right) \right), \quad (19)$$

$i = \overline{1, n},$

где $bS_description_r(N_z)$ - обозначение части агрегата $bS_description$, описывающего принимающую часть добавляемого/модифицируемого функционального сервиса N_z ; $bS_description_t(N_z)$ - обозначение части агрегата $bS_description$, описывающего передающую часть добавляемого/модифицируемого функционального сервиса N_z ; $bS_description_r(M_R)$ - обозначение части агрегата $bS_description$, описывающего принимающие части хранимого в реестре множества актуальных сервисов SOA-ИС; $bS_description_t(M_R)$ - обозначение части агрегата $bS_description$, описывающего передающие части хранимого в реестре множества актуальных сервисов SOA-ИС.

Тогда второе из приведенных выше условий выполнения операции добавления/модификации функционального сервиса можно записать следующим выражением:

$$\begin{aligned} & [bS_description_r(N_z) \cap bS_description_t(M_R)] \cup \\ & \cup [bS_description_t(N_z) \cap bS_description_r(M_R)] \neq \emptyset \end{aligned} \quad (20)$$

Третье из приведенных выше условий предполагает проверку уникальности традиционных описаний агрегатов добавляемого/ модифицируемого функционального сервиса N_z и хранимого в реестре множества актуальных сервисов SOA-ИС. Поэтому по аналогии с рассмотренной выше операцией реализации первого варианта интеграции введем теоретико-множественное описание хранимого в реестре множества актуальных сервисов SOA-ИС $RFS''(M_R)$ (выражение (3)) и теоретико-множественное описание добавляемого/ модифицируемого функционального сервиса N_z (выражение (4)). Поскольку модель реестра множества актуальных сервисов SOA-ИС не предусматривает уникальных агрегатов для описания принимающей и передающей частей функциональных сервисов, возможно разделение данных описаний на следующие варианты:

а) описание хранимого в реестре множества принимающих частей актуальных сервисов SOA-ИС $RFS_R''(M_R)$;

б) описание хранимого в реестре множества передающих частей актуальных сервисов SOA-ИС $RFS_T''(M_R)$;

в) описание принимающей части добавляемого/модифицируемого функционального сервиса N_z $RFS_R''(N_z)$;

г) описание передающей части добавляемого/модифицируемого функционального сервиса N_z $RFS_T''(N_z)$.

При этом данные описания будут в общем случае аналогичны выражениям (3) и (4) соответственно.

Тогда третье из приведенных выше условий выполнения операции добавления/модификации функционального сервиса можно записать следующим выражением:

$$[RFS_R''(N_z) \cap RFS_T''(M_R)] \cup [RFS_T''(N_z) \cap RFS_R''(M_R)] \neq \emptyset. \quad (21)$$

Исходя из выражений (15), (20) и (21), операцию добавления/модификации функционального сервиса N_z , реализующую второй вариант интеграции, можно представить продукционным правилом следующего вида:

$$\begin{aligned} & \text{если } [bE_description_r(N_z) \cap bE_description_t(M_R)] \cup \\ & \cup [bE_description_t(N_z) \cap bE_description_r(M_R)] \neq \emptyset \\ & \text{и } [bS_description_r(N_z) \cap bS_description_t(M_R)] \cup \\ & \cup [bS_description_t(N_z) \cap bS_description_r(M_R)] \neq \emptyset \\ & \text{и } [RFS_R''(N_z) \cap RFS_T''(M_R)] \cup [RFS_T''(N_z) \cap RFS_R''(M_R)] \neq \emptyset \\ & \text{то } RFS(M_R) + RFS(N_z) \end{aligned} \quad (22)$$

Операцией реализации третьего варианта интеграции является удаление из реестра множества актуальных сервисов SOA-ИС описания функционального сервиса, чьи функции обработки хранимых данных оказались не востребованными каким-либо другим сервисом SOA-ИС. В [9] показано, что выполнение данного варианта интеграции возможно только при выполнении условий, согласно

которым решение об исключении функционального сервиса O_z должно приниматься, если происходят следующие события:

а) в реестре существует описание хотя бы одного функционального сервиса, который передает те же данные, что и исключаемый функциональный сервис O_z ;

б) в реестре не существует описаний функциональных сервисов, которые принимают данные, передаваемые исключаемым функциональным сервисом O_z .

Таким образом, реализация третьего варианта интеграции будет возможной при выполнении следующих условий:

а) выполнение описанных в [9] условий для описаний понятий и терминов предметной области исключаемого функционального сервиса и функциональных сервисов, описания которых уже хранятся в реестре;

б) выполнение описанных в [9] условий для описаний онтологий исключаемого функционального сервиса и функциональных сервисов, описания которых уже хранятся в реестре;

в) выполнение описанных в [9] условий для традиционных описаний метаданных исключаемого функционального сервиса и функциональных сервисов, описания которых уже хранятся в реестре.

Для формализованного описания первого из указанных условий используем введенные выше дополнительные формализованные описания агрегатов $bE_description$. На их основе сформируем:

а) дополнительное формализованное описание агрегата $bE_description$ принимающей части исключаемого функционального сервиса O_z $bE_description_r(O_z)$ (описывается выражением, аналогичным выражению (11));

б) дополнительное формализованное описание агрегата $bE_description$ передающей части исключаемого функционального сервиса O_z $bE_description_t(O_z)$ (описывается выражением, аналогичным выражению (12));

в) дополнительное формализованное описание агрегата $bE_description$ передающей части актуального функционального сервиса s_i $bE_description_t(s_i)$ (описывается выражением, аналогичным выражению (12)).

Тогда первое из приведенных выше условий выполнения операции исключения функционального сервиса можно записать следующими выражениями:

$$\begin{aligned} & (bE_description_t(s_i) \cap bE_description_t(O_z)) \cap \\ & \cap bE_description(M_R) \neq \emptyset; \end{aligned} \quad (23)$$

$$i, z = \overline{1, n}; i \neq z;$$

$$\begin{aligned} & bE_description(M_R) \cap \\ & \cap (bE_description_t(s_i) \cap bE_description_r(O_z)) = \emptyset; \end{aligned} \quad (24)$$

$$i, z = \overline{1, n}; i \neq z.$$

Для формализованного описания второго из указанных условий используем введенные выше дополнительные формализованные описания агрегатов $bS_description$. На их основе сформируем:

а) дополнительное формализованное описание агрегата $bS_description$ принимающей части исключаемого функционального сервиса O_z $bS_description_r(O_z)$ (описывается выражением, аналогичным выражению (16));

б) дополнительное формализованное описание агрегата $bS_description$ передающей части исключаемого функционального сервиса O_z $bS_description_t(O_z)$ (описывается выражением, аналогичным выражению (17));

в) дополнительное формализованное описание агрегата $bS_description$ передающей части актуального функционального сервиса s_i $bS_description_t(s_i)$ (описывается выражением, аналогичным выражению (17)).

Тогда второе из приведенных выше условий выполнения операции исключения функционального сервиса можно записать следующими выражениями:

$$\begin{aligned} & (bS_description_t(s_i) \cap bS_description_t(O_z)) \cap \\ & \cap bS_description(M_R) \neq \emptyset; \end{aligned} \quad (25)$$

$$i, z = \overline{1, n}; i \neq z;$$

$$\begin{aligned} & bS_description(M_R) \cap \\ & \cap (bS_description_t(s_i) \cap bS_description_r(O_z)) = \emptyset; \end{aligned} \quad (26)$$

$$i, z = \overline{1, n}; i \neq z.$$

Для формализованного описания третьего из указанных условий используем следующие, аналогичные выражению (4), теоретико-множественные описания:

- описание принимающей части исключаемого функционального сервиса O_z $RFS_R''(O_z)$;

- описание передающей части исключаемого функционального сервиса O_z $RFS_T''(O_z)$;

- описание передающей части актуального функционального сервиса s_i $RFS_T''(s_i)$.

Тогда третье из приведенных выше условий выполнения операции исключения функционального сервиса можно записать следующими выражениями:

$$\begin{aligned} & (RFS_T''(s_i) \cap RFS_T''(O_z)) \cap RFS''(M_R) \neq \emptyset; \end{aligned} \quad (27)$$

$$i, z = \overline{1, n}; i \neq z;$$

$$\begin{aligned} & RFS''(M_R) \cap (RFS_T''(s_i) \cap RFS_R''(O_z)) = \emptyset; \end{aligned} \quad (28)$$

$$i, z = \overline{1, n}; i \neq z.$$

Тогда операцию исключения функционального сервиса O_z из реестра, реализующую третий вариант интеграции, можно представить продукционным правилом следующего вида:

$$\begin{aligned}
& \text{если } (bE_description_t(s_i) \cap bE_description_t(O_z)) \cap \\
& \cap bE_description(M_R) \neq \emptyset \\
& u bE_description(M_R) \cap \\
& \cap (bE_description_t(s_i) \cap bE_description_r(O_z)) = \emptyset \\
& u (bS_description_t(s_i) \cap bS_description_t(O_z)) \cap \quad . \quad (29) \\
& \cap bS_description(M_R) \neq \emptyset \\
& u bS_description(M_R) \cap \\
& \cap (bS_description_t(s_i) \cap bS_description_r(O_z)) = \emptyset \\
& u (RFS_T''(s_i) \cap RFS_T''(O_z)) \cap RFS''(M_R) \neq \emptyset \\
& u RFS''(M_R) \cap (RFS_T''(s_i) \cap RFS_R''(O_z)) = \emptyset \\
& \text{то } RFS(M_R) - RFS(O_z).
\end{aligned}$$

Унифицированный метод интеграции функциональных сервисов.

Модели операций интеграции функциональных сервисов (10), (22) и (29) позволяют формализованно описать процесс формирования реестра измененного множества актуальных сервисов как этапа рассмотренной в [10] информационной технологии (ИТ) интеграции разнородных функциональных сервисов как совокупность отображений следующего вида:

$$F_{M_{R'}}^{(M_R, N_z)} : (M_R, N_z) \xrightarrow{(10)} M_{R'}, \quad (30)$$

$$F_{M_{R'}}^{(M_R, N_z)} : (M_R, N_z) \xrightarrow{(22)} M_{R'}, \quad (31)$$

$$F_{M_{R'}}^{(M_R, O_z)} : (M_R, O_z) \xrightarrow{(29)} M_{R'}. \quad (32)$$

Однако такое представление данного этапа требует в ходе реализации ИТ принятия решения о выборе конкретной операции интеграции перед началом обработки описаний каждого отдельного сервиса, поступившего в SOA-ИС на момент начала выполнения процесса интеграции. Кроме того, каждая из моделей операций (10), (22) и (29) по отдельности довольно трудоемка для выполнения. Поэтому становится необходимым создание специального метода, позволяющего сократить количество операций по обработке хранимых в реестре данных и описаний отдельных сервисов.

В основе данного метода находится описанная в [6] модель реестра функциональных сервисов, которая используется для описания реестра множества актуальных сервисов SOA-ИС, отдельных сервисов, поступивших в SOA-ИС на момент начала выполнения процесса интеграции, а также реестра измененного множества актуальных сервисов SOA-ИС. Такая унификация позволяет ввести термин «интегрируемый сервис» S_z , который может описывать:

а) новый сервис N_z , для которого следует выполнить операцию добавления нового функционального сервиса (10);

б) новый сервис N_z , для которого следует выполнить операцию добавления/модификации функционального сервиса (22);

в) исключаемый сервис O_z , для которого следует выполнить операцию исключения функционального сервиса из реестра (29).

Основываясь на данном термине, введем множество $(S_{z_1}, \dots, S_{z_i}, \dots, S_{z_m})$ отдельных сервисов, поступивших в SOA-ИС на момент начала выполнения процесса интеграции. Данное множество позволяет представить операции (10), (22) и (29) следующим образом:

$$\begin{aligned} & \text{если } bE_description(M_R) \cap bE_description(S_{z_i}) = \emptyset \\ & u \ bS_description(M_R) \cap bS_description(S_{z_i}) = \emptyset \\ & u \ RFS''(M_R) \cap RFS''(S_{z_i}) = \emptyset \\ & \text{то } RFS(M_R) + RFS(S_{z_i}) \end{aligned} \quad , \quad (33)$$

$$\begin{aligned} & \text{если } [bE_description_t(M_R) \cap bE_description_r(S_{z_i})] \cup \\ & \cup [bE_description_r(M_R) \cap bE_description_t(S_{z_i})] \neq \emptyset \\ & u [bS_description_t(M_R) \cap bS_description_r(S_{z_i})] \cup \\ & \cup [bS_description_r(M_R) \cap bS_description_t(S_{z_i})] \neq \emptyset \\ & u [RFS_T''(M_R) \cap RFS_R''(S_{z_i})] \cup [RFS_R''(M_R) \cap RFS_T''(S_{z_i})] \neq \emptyset \\ & \text{то } RFS(M_R) + RFS(S_{z_i}) \end{aligned} \quad , (34)$$

$$\begin{aligned} & \text{если } bE_description(M_R) \cap \\ & \cap (bE_description_t(s_i) \cap bE_description_t(S_{z_i})) \neq \emptyset \\ & u \ bE_description(M_R) \cap \\ & \cap (bE_description_t(s_i) \cap bE_description_r(S_{z_i})) = \emptyset \\ & u (bS_description_t(s_i) \cap bS_description_t(S_{z_i})) \cap \\ & \cap bS_description(M_R) \neq \emptyset \\ & u \ bS_description(M_R) \cap \\ & \cap (bS_description_t(s_i) \cap bS_description_r(S_{z_i})) = \emptyset \\ & u (RFS_T''(s_i) \cap RFS_T''(S_{z_i})) \cap RFS''(M_R) \neq \emptyset \\ & u \ RFS''(M_R) \cap (RFS_T''(s_i) \cap RFS_R''(S_{z_i})) = \emptyset \\ & \text{то } RFS(M_R) - RFS(S_{z_i}). \end{aligned} \quad (35)$$

Следует учесть, что выполнение условия $bE_description_t(M_R) \cap bE_description_r(S_{z_i}) \neq \emptyset$ возможно только в случае, если существует хотя бы один сервис $s_i \in M_R$, для которого выполняется это условие. Исходя из этого, процесс принятия решения о выполнении операций интеграции разнородных функциональных сервисов может быть представлен как последовательное применение деревьев решений, показанных на рис. 1 – 3.

Тогда применение унифицированного метода интеграции функциональных сервисов позволит представить отображения (3.51)-(3.53) как частные случаи следующего отображения:

$$F_{M_R'}^{(M_R, (S_{z_1}, \dots, S_{z_m}))} : (M_R, (S_{z_1}, \dots, S_{z_m})) \xrightarrow{\text{(УМИФС)}} M_R', \quad (36)$$

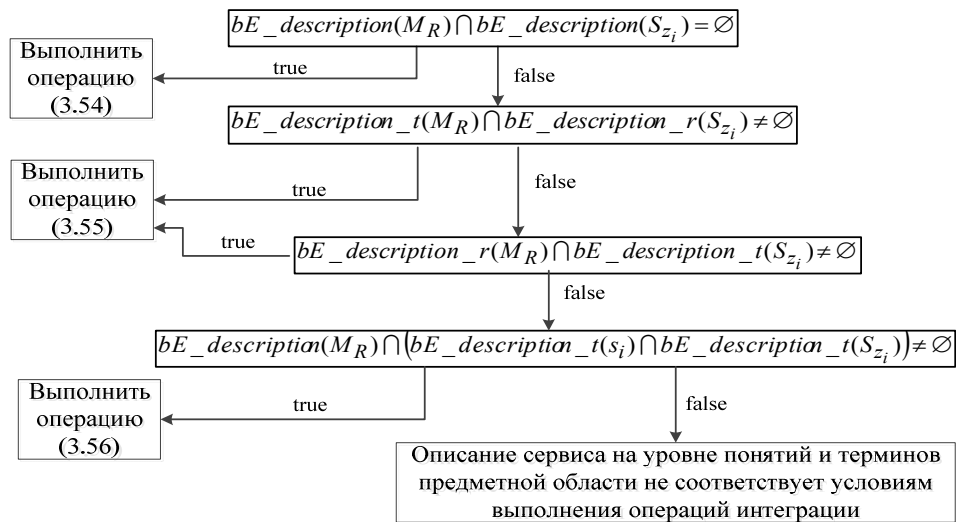


Рис. 1 – Дерево решений по выбору операции интеграции на уровне описаний понятий и терминов предметной области

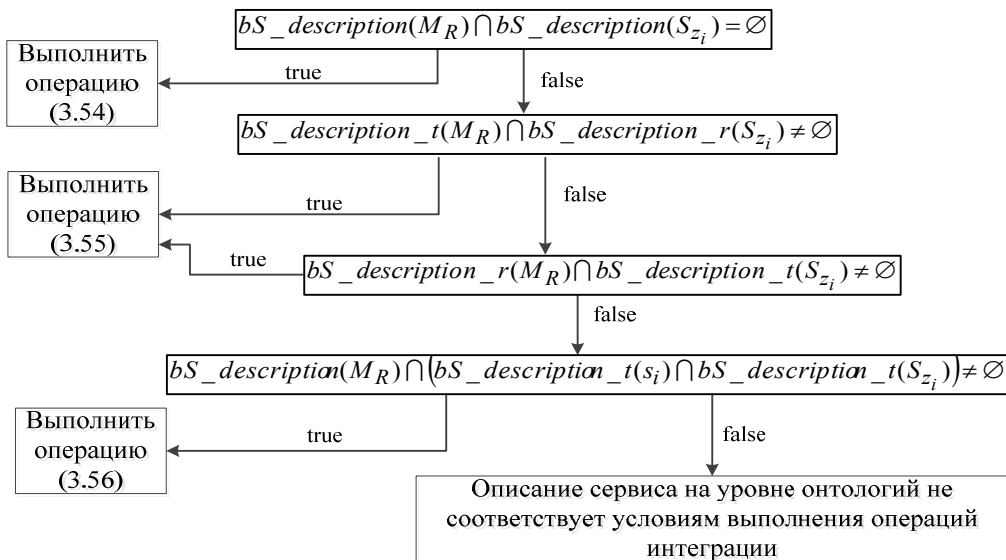


Рис. 2 – Дерево решений по выбору операции интеграции на уровне описаний онтологий

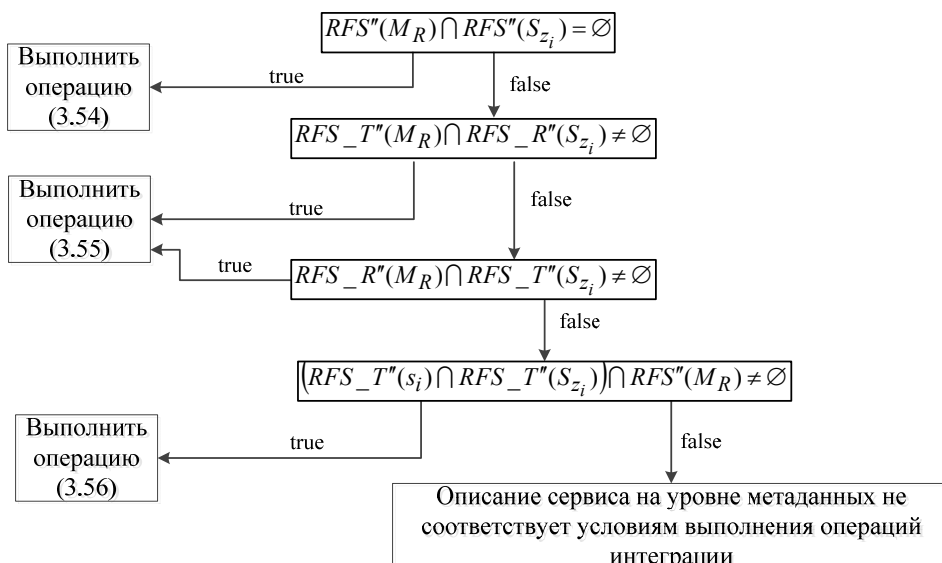


Рис. 3 – Дерево решений по выбору операции интеграции на уровне описаний метаданных

На рис. 3: УМИФС - набор этапов унифицированного метода интеграции функциональных сервисов, выполнение которого приводит к формированию реестра измененного множества актуальных сервисов SOA-ИС.

Подобное представление процесса формирования реестра измененного множества актуальных сервисов SOA-ИС, а также описанные выше деревья решений позволяют описать унифицированный метод интеграции функциональных сервисов как последовательность таких этапов.

Этап 1. Проверка условия $RFS(S_{z_1}, \dots, S_{z_i}, \dots, S_{z_m}) = \emptyset$. Если условие истинно – признать невозможность выполнения операций интеграции вследствие отсутствия описаний интегрируемых функциональных сервисов и прекратить выполнение метода. В противном случае – принять $i = 1$ и перейти к Этапу 2.

Этап 2. Проверка условия $bE_description(S_{z_i}) = \emptyset$. Если условие истинно – признать невозможность выполнения операций интеграции, поскольку сервис не связан с предметной областью объекта автоматизации, и прекратить выполнение метода. В противном случае – перейти к Этапу 3.

Этап 3. Проверка условия $bS_description(S_{z_i}) = \emptyset$. Если условие истинно – признать невозможность выполнения операций интеграции, поскольку входные данные и результаты работы сервиса не связаны с предметной областью объекта автоматизации, и прекратить выполнение метода. В противном случае – перейти к Этапу 4.

Этап 4. Формирование по результатам проверки условий дерева решений, показанного на рис. 1, описания решения о выполнении операции интеграции над сервисом S_{z_i} в виде кортежа

$$Op(S_{z_i}) = \langle op_{bE} \rangle, \quad (37)$$

где $op_{bE} = [1, 2, 3, e_{bE}]$ - идентификатор операции интеграции или сообщения об ошибке описания интегрируемого сервиса на уровне понятий и терминов предметной области.

Этап 5. Модификация по результатам проверки условий дерева решений, показанного на рис. 2, описания $Op(S_{z_i})$ путем его приведения к виду

$$Op(S_{z_i}) = \langle op_{bE}, op_{bS} \rangle, \quad (38)$$

где $op_{bS} = [1, 2, 3, e_{bS}]$ - идентификатор операции интеграции или сообщения об ошибке описания интегрируемого сервиса на уровне онтологий.

Этап 6. Модификация по результатам проверки условий дерева решений, показанного на рис. 3, описания $Op(S_{z_i})$ путем его приведения к виду

$$Op(S_{z_i}) = \langle op_{bE}, op_{bS}, op_{RFS} \rangle, \quad (39)$$

где $op_{RFS} = [1, 2, 3, e_{RFS}]$ - идентификатор операции интеграции или сообщения об ошибке описания интегрируемого сервиса на уровне метаданных.

Этап 7. Проверка для сформированного в ходе выполнения Этапов 4-6 описания $Op(S_{z_i})$ условия $op_{bE} = op_{bS} = op_{RFS}$. Если это условие выполняется –

выполнить операцию интеграции, идентификатор которой зафиксирован в описании $Op(S_{z_i})$ и перейти к Этапу 10. В противном случае – перейти к этапу 8.

Этап 8. Если описание $Op(S_{z_i})$ содержит идентификаторы ошибок, выявленных в ходе проверки условий деревьев решений, то признать невозможность выполнения операций интеграции для данного сервиса с указанием выявленных ошибок и перейти к Этапу 10. В противном случае – перейти к Этапу 9.

Этап 9. Признать невозможность выполнения операций интеграции для данного сервиса по причине несоответствия описаний сервиса S_{z_i} на различных уровнях и перейти к Этапу 10.

Этап 10. Исключить описание сервиса S_{z_i} из рассматриваемого множества описаний $(S_{z_1}, \dots, S_{z_i}, \dots, S_{z_m})$ и принять $i = i + 1$. Если условие $i > m$ не выполняется, то перейти к Этапу 2, иначе признать множество описаний $(S_{z_1}, \dots, S_{z_i}, \dots, S_{z_m})$ рассмотренным полностью и завершить работу метода.

Главной предпосылкой, определяющей возможность применения данного метода, является условие $RFS(M_R) \neq \emptyset$. Это условие определяет существование реестра множества актуальных сервисов SOA-ИС на момент начала выполнения процесса интеграции разнородных функциональных сервисов, в котором присутствуют описания каждого из актуальных сервисов на уровне понятий и терминов предметной области, а также на уровне онтологий.

Выводы. Предлагаемые модели операций интеграции функциональных сервисов (10), (22) и (29) позволяют рассматривать реализацию этих операций путем выполнения запросов к базе данных, реализующей реестр функциональных сервисов SOA-ИС на одном из стандартных языков манипулирования данными. Это, в свою очередь, позволяет реализовать разработанные в данном подразделе модели операций в подавляющем большинстве разрабатываемых или уже эксплуатируемых SOA-ИС без значительного изменения их системного программного обеспечения (в частности СУБД).

Основным преимуществом разработанного унифицированного метода интеграции функциональных сервисов является возможность досрочного прекращения выполнения операции интеграции, если обнаруживается ошибка, вызванная несоответствием описаний сервиса описаниям предметной области. Такая возможность позволит сократить время выполнения метода для неправильно описанных сервисов за счет обработки меньшего объема данных. Кроме того, такая возможность позволит повысить защищенность реестра функциональных сервисов SOA-ИС от несанкционированной записи сервисов различных производителей, назначение которых неизвестно администраторам ИС.

Список литературы: 1. Деревянко А. С. Технологии и средства консолидации информации: учебн. пос. [Текст] / А. С. Деревянко, М. Н. Солощук – Харьков: НТУ "ХПИ", 2008. – 432с. 2. Шаховська Н. Б. Сховища та простори даних: монографія [Текст] / Н. Б. Шаховська, В. В. Пасічник. – Львів: Видавництво Національного університету «Львівська політехніка», 2009. – 244 с. 3. Parikh Ash. SOA в реальности [Электронный ресурс] / Ash Parikh, Murty Gurajada //

Сайт ERPNews. – Режим доступа: <http://erpnews.ru/doc2610.html>. – Заголовок с экрана. **4.** 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. **5.** Никитюк В. А. Модель реестра функциональных сервисов информационной системы [Текст] / В.А. Никитюк // «Застосування інформаційних технологій у підготовці та діяльності сил охорони правопорядку». Збірник тез доповідей науково-практичної конференції 20-21 березня 2013 року. – Харків: Академія внутрішніх військ МВС України, 2013. – С. 80-81. **6.** Никитюк В. А. Усовершенствование модели реестра web-сервисов [Текст] / В. А. Никитюк // Вісник Кременчуцького національного університету імені Михайла Остроградського. – 2013. – Випуск 5/2014 (88). – С. 98 – 104. **7.** Никитюк В. А. Модель семантического представления функциональных сервисов [Текст] / В. А. Никитюк // Вісник національного технічного університету «ХПІ». Збірник наукових праць. Серія: «Нові рішення в сучасних технологіях». – Харків: НТУ «ХПІ». – 2012. – № 68. – С. 54-62. **8.** Никитюк В. А. Модель описаний терминов предметной области в сервис-ориентированной информационной системе [Текст] / В. А. Никитюк // Информационные системы и технологии: материалы 3-й Международ. науч.-техн. конф., Харьков, 15-21 сентября 2014 г.: тезисы докладов. - Харьков: ХНУРЕ, 2014. – С. 68-69. **9.** Евланов М. В. Формализованное описание условий интеграции IT-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. **10.** Никитюк В. А. Усовершенствование модели процессов интеграции разнородных функциональных сервисов [Текст] / В. А. Никитюк // Системний аналіз. Інформатика. Управління (САІУ-2013): матеріали IV Міжнародної науково-практичної конференції (м. Запоріжжя, 13-16 березня 2013 р.). – Запоріжжя: КПУ, 2013. –С. 180-182.

Bibliography (transliterated): **1.** Derevjanko A. S., Soloschuk M. N. (2008). Technologies and means of consolidating information: tutorial. Kharkov: NTU “KhPI”, 432. **2.** Shakhovska N. B., Pasichnyk V.V. (2009). Data store and data spaces: monograph. Vydavnytstvo natsyonalnoho unyversyteta «Lvivska politehnika», Lviv, Ukraine. 244. **3.** Parikh A. and Gurajada M. (2014). "SOA is a reality", available at: <http://erpnews.ru/doc2610.html>. **4.** Sayt OASIS (Advancing open standards for the information society) / "UDDI Version 3.0.2" (2014). Available at: <http://erpnews.ru/doc2610.html>. **5.** Nykytyuk, V. A. (2013). Registry model of functional services information system. The use of information technology in the preparation of forces and law enforcement. Abstracts of scientific and practical conference. Kharkiv, Akademiya vnutrennykh viysk MVS Ukrayiny, March 20-21, 2013, 80-81. **6.** Nykytyuk, V. A. (2014). Web-service registry model improvements. Transactions of Kremenchuk Mykhailo Ostrohradskyi National University. Issue 5/2014 (88). 98-104. **7.** Nykytyuk V. A. (2012). Model of semantic representation of functional services. Visnyk natsyonalnoho tekhnicheskoho unyversyteta «KHPI». Zbirnyk naukovykh prats. Seriya: «Novi reshenye v suchasnikh tekhnolohiyakh. Kharkiv: NTU «KHPI», no. 68. 54-62. **8.** Nykytyuk V. A. (2014). Model descriptions domain terms in service-oriented information system. Information Systems and Technologies: Proceedings of the 3-th Intern. sci.-techn. conf. Kharkov. 15-21 september 2014: abstracts. Kharkov. KhNURE. 68-69. **9.** Vasytsova N. V., Ievlanov M. V., Nykytyuk V. A. (2011). A formal description of the conditions of integration IT-services in enterprise information management systems. Visnyk Akademyy mytnoyi sluzhby Ukrayiny. Seriya «Tekhnichni nauky». No. 2 (46). 87-96. **10.** Nykytyuk V. A. (2013). Improvement process model integration of heterogeneous functional services. System analysis. Informatics. Management (SAIU 2013): Proceedings of the IV International Scientific Conference (m. Zaporizhzhya, 13-16 March 2013). Zaporizhzhya. KPU. 180-182.

Надійшла (received) 18.12.2014