

УДК 004.021:004.75

Н.А. Куликовская, ассистент  
Запорожский национальный технический университет, г. Запорожье  
natalya.gontar@gmail.com

## Методы взаимодействия компонент семантической сервис-ориентированной архитектуры

*В данной работе рассматриваются методы взаимодействия элементов семантической сервис-ориентированной архитектуры (ССОА). Данные методы необходимы при проектировании сервис-ориентированных систем. К основным элементам ССОА относятся: множество сервисов, множество онтологий и семантический сервис. Множество онтологий представлено онтологиями сервисов, доменными онтологиями и онтологиями задач. В зависимости от стадии развития системы и ресурсов провайдера сервиса используются нисходящий, восходящий или смешанный метод взаимодействия компонент ССОА. При этом можно получить преимущество либо во времени, либо в трудозатратах на развитие системы в зависимости от выбранного пути.*

**Ключевые слова:** метод, семантическая сервис-ориентированная архитектура, семантический сервис.

### Введение

Ключевым фактором успешной реализации и функционирования семантической сервис-ориентированной архитектуры (ССОА) является управление и организация информационных потоков путем автоматического осмысления семантики передаваемых данных [1]. Рассматривая ССОА, мы формализуем данные сервиса путем описания его онтологии. Таким способом реализуется семантический принцип архитектуры. Сервис-ориентированный принцип ССОА заключается в проектировании сервисов и композитных приложений, представляющих прикладную логику процессов в виде набора услуг.

При правильной организации ССОА получаем такие преимущества [2, 3]:

- интероперабельность сервисов. Взаимодействие между различными компонентами на любой платформе, написанными на любом языке программирования;
- слабосвязность сервисов, которая означает, что взаимодействие между приложениями не нарушается каждый раз, когда меняется дизайн или реализация какого-либо сервиса;
- адаптируемость существующих приложений к меняющимся условиям бизнеса и потребностям заказчика, а также накапливание знания в формализованном виде для дальнейшего их использования;
- семантическая направленность. Данные приобретают новую форму представления, которая позволяет «осмысленную» машинную обработку.

### Постановка задачи

Концепция ССОА основана на архитектурном стиле, который определяет модель взаимодействия между тремя основными сторонами: поставщиком услуг, который публикует описания сервиса и обеспечивает реализацию услуги; потребителем, который может использовать универсальный идентификатор ресурса (URI) и использовать сервис для решения своих задач; семантическим сервисом, который обеспечивает и поддерживает реестр сервисов, и работу с онтологиями [4].

Сервис-ориентированная система (СОС) может быть реализована на основе ССОА. Для этого необходимо существование всех элементов обозначенных на модели ССОА представленной на рис. 1. Human Actor – это пользователь СОС, им могут быть потребитель или поставщик сервиса. Semantic Service / Domain Ontology – семантический сервис (SS) и доменная онтология. Service / Service Description / Service Interface / Application Ontology – элементы сервиса [5].

В связи с необходимостью создания онтологий разных видов возникает вопрос в получении их оптимальным путем для поставщика сервисов и эффективным представлением для пользователя СОС. Целью данной статьи является представление и анализ методов взаимодействия элементов СОС на основе ССОА, в особенности формирование семантических компонент системы.

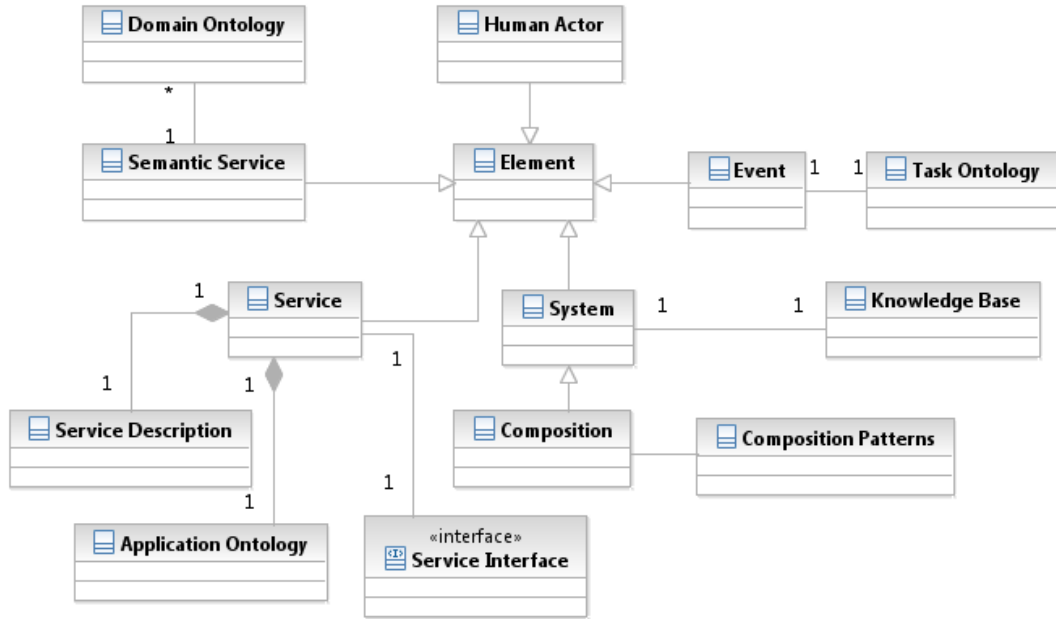


Рисунок 1 – Модель ССОА

**Основная часть**

Формально описать модель ССОА можно в виде кортежа:

$$SSOA = \langle O, S, E, SS, Syst \rangle, \quad (1)$$

где  $O$  – множество онтологий;

$S$  – множество сервисов ССОА;

$E$  – множество событий, которое описывают процессы системы;

$SS$  – семантический сервис;

$Syst$  – содержит множество композитных сервисов и базу знаний.

Множество онтологий состоит из множеств трех видов онтологий:

$$O = \{O_{Task}, O_{Application}, O_{Domain}\}, \quad (2)$$

где  $O_{Task}$  – онтологии событий ССОА;

$O_{Application}$  – онтологии сервисов;

$O_{Domain}$  – онтологии специализированных областей данных систем на основе ССОА.

Семантический сервис включает в себя доменные онтологии и набор модулей работы с сервисами и их онтологическим описанием:

$$SS = \langle O_{Domain}, f_{Service}(S), f_{Ontology}(O_{Application}) \rangle, \quad (3)$$

где  $f_{Service}(S)$  – функции обработки сервисов;

$f_{Ontology}(O_{Application})$  – функции модели-

рования онтологий сервисов.

Множество событий состоит из Task Ontology, времени начала его выполнения ( $t$ ) и длительности ( $d$ ):

$$E = \langle O_{Task}, t, d \rangle. \quad (4)$$

Система состоит из сервисов, правил их композиции, базы знаний:

$$Syst = \langle S, f_{Composition}(S), KB \rangle. \quad (5)$$

Рассмотрим методы взаимодействия компонент СОС на примере системы, состоящей из одного грид-сервиса. Под грид-сервисом будем понимать предоставляемое пользователям компьютерное оборудование и программное обеспечение для вычислений. Рассмотрим данный пример в элементах ССОА. Грид-сервис (Service) предоставляет пользователю определенный программный интерфейс (Service Interface), имеет спецификацию (Service Description) и онтологию (Application Ontology). Семантический сервис, в нашем примере это среда Protégé, состоит из набора модулей, которые отвечают за сравнение онтологий, хранение онтологий, автоматического построения онтологий (плагины для преобразования WSDL/OWL-S) [6, 7]. Доменные онтологии (Domain Ontology) содержат информацию о математических вычислениях, компьютерном оборудовании, функциональных и не функциональных характеристиках грид-сервиса. События (Event) в СОС возможны такие: регистрация сервиса, вызов сервиса, хранение сервиса, построение онтологии сервиса, верификация онтологии, поиск онтологии, обновление онтологии. Онтология событий

(Task Ontology) хранит алгоритмы правильного выполнения событий СОС. Например, событие о регистрации сервиса не может быть раньше, чем осуществлена проверка его онтологического описания.

На рис. 2 показан технологический стек СОС.

QoS	OWL-S
Семантический сервис	JAVA
Онтологии	Web Ontology Language (OWL)
Обмен данными	Simple Object Access Protocol (SOAP)
	XML

Рисунок 2 – Технологический стек ССОА

Применить ССОА можно либо к существующим СОС, либо к только разрабатываемым системам, поэтому мы приводим три различных метода организации взаимодействия между элементами СОС: нисходящий, восходящий и смешанный.

1. Нисходящий метод. Данный метод применяется при переходе СОС на ССОА. Он основан на использовании единых доменных онтологий. Каждый сервис системы должен быть описан в соответствии с доменной онтологией, которая связана в семантическом сервисе. Доменных онтологий может быть несколько для отдельных структур системы.

Начальными условиями такого метода являются наличие множества сервисов, семантиче-

ского сервиса, множества доменных онтологий и системы:

$$\langle S, SS, O_{Domain}, Syst \rangle. \quad (6)$$

Дальнейшее взаимодействие компонент системы, происходит следующим образом: генерируется онтология регистрируемого сервиса с помощью семантического сервиса и соответствующей доменной онтологии.

Приведем последовательность действия при нисходящем методе для нашего примера (рис. 3). Поставщик разработал свой эффективный грид-сервис и обращается к СОС с просьбой его зарегистрировать. Система получает запрос о регистрации сервиса и не имея его онтологии обращается к семантическому сервису с запросом ее сгенерировать. В свою очередь, семантический сервис на основании доменной онтологии генерирует онтологическое описание для грид-сервиса (Application Ontology). Система имея грид-сервис и его онтологию, регистрирует данное приложение. Преимуществами такого подхода является большая вероятность точного и релевантного поиска, т.к. онтологии генерируются одним объектом системы. При автоматической генерации онтологий появляется возможность облегчить и ограничить потребителя от разработки таковых. Так же, при переходе существующей СОС на ССОА все трудозатраты сводятся к разработке доменных онтологий, которые в дальнейшем используются как базис. Затраченное время на процесс регистрации сервиса занимает 10 итераций.

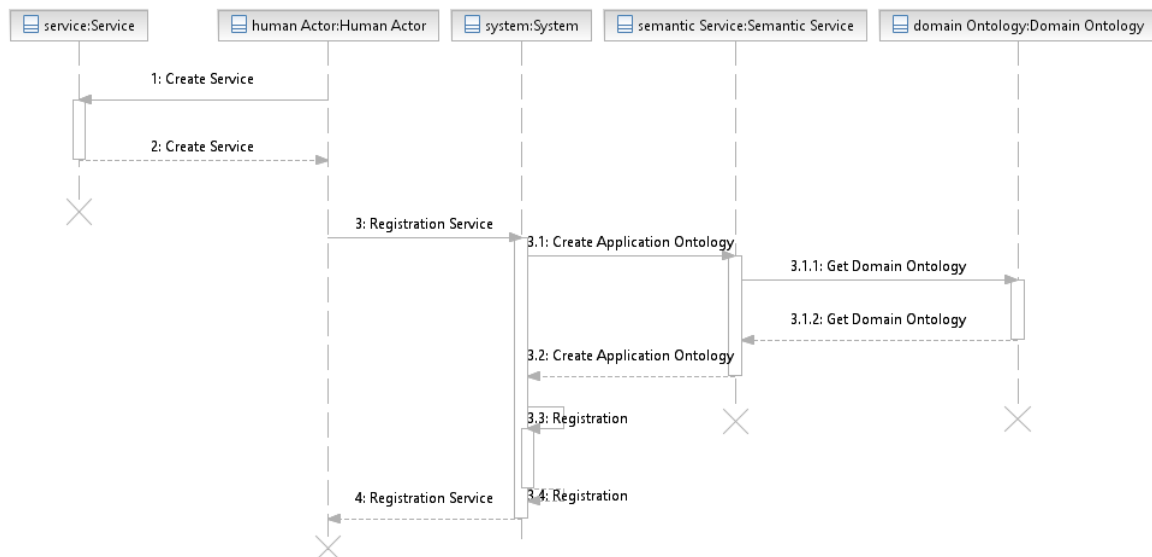


Рисунок 3 – Нисходящий метод организации взаимодействия между элементами ССОА

2. Восходящий метод. При первичной разработке СОС целесообразней уделить внимание созданию онтологий сервисов отдельно от всей системы. Каждый сервис описывается своей собственной онтологией (Application Ontology), кото-

рую реализует провайдер сервиса (рис.4). Данный метод можно описать следующим образом, где начальными условиями являются наличие множества сервисов, семантического сервиса, множества онтологий сервисов и системы:

$$\langle S, SS, O_{Application}, Syst \rangle. \quad (7)$$

Все онтологии услуг попадают в семантический сервис, который из полученных знаний формирует доменные онтологии.

Рассмотрим наш пример относительно данного метода. Поставщик грид-сервиса самостоятельно разрабатывает онтологию приложения и обращается к системе с просьбой о регистрации сервиса. Система направляет онтологическое описание в семантический сервис, который проверяет его и формирует доменную онтологию. В таком

случае затраченное время будет состоять из 11 итераций.

В данном методе сервис может развиваться независимо от других ресурсов или их онтологий, поэтому при таком построении архитектуры сравнительно легко вносить изменения в сервисы, добавлять новые и т.д. (улучшаются качественные характеристики архитектуры: расширяемость, масштабируемость, способность к эволюционным изменениям). Однако процесс добавления и выделения доменных онтологий является очень затруднительным.

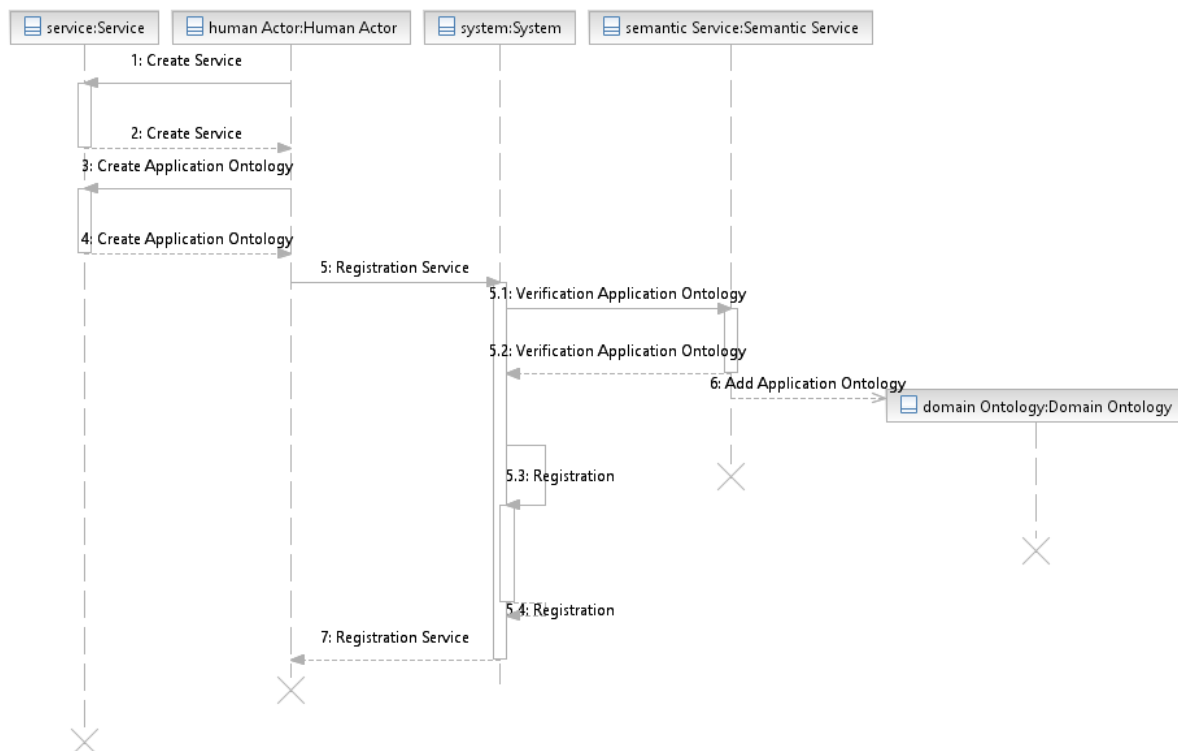


Рисунок 4 – Восходящий метод организации взаимодействия между элементами ССОА

3. Смешанный метод - аналогично предыдущему, семантика каждого сервиса (Application Ontology) не генерируется средствами семантического сервиса. Но для того, чтобы онтологии ресурсов были сравнимы друг с другом в семантическом плане, они строятся на основе доменной онтологии (Domain Ontology), которую получает провайдер (рис.5).

Третий метод описывается следующими кортежем:

$$\langle S, SS, O_{Domain}, O_{Application}, Syst \rangle. \quad (8)$$

Приведем последовательность действий для нашего примера. Провайдер грид-сервиса запрашивает доменную онтологию, на основе которой сам строит онтологию приложения и далее регистрирует свой сервис. В таком случае

метод выполняется в 17 итераций. Но при таком увеличении количества итераций, он наследует все преимущества двух описанных ранее методов.

### Заключение

Таким образом, главным отличием СОС, построенных на основе ССОА является эффективное использование онтологий. В таком случае можно использовать три метода организации взаимодействия между компонентами СОС. Нисходящий метод полностью основан на функциях семантического сервиса и больше подходит для существующих систем, которые решили перейти на ССОА. Восходящий метод опирается на работу провайдера сервиса, который самостоятельно разрабатывает онтологии приложений. Смешанное проектирование является синтезом

работы семантического сервиса и провайдера сервиса. Затраченное время на публикацию сервиса в каждом методе разное, но и трудоемкость провайдера тоже отличается. Предполагается,

что при выборе какого-либо из этих методов необходимо руководствоваться возможностями провайдера сервисов.

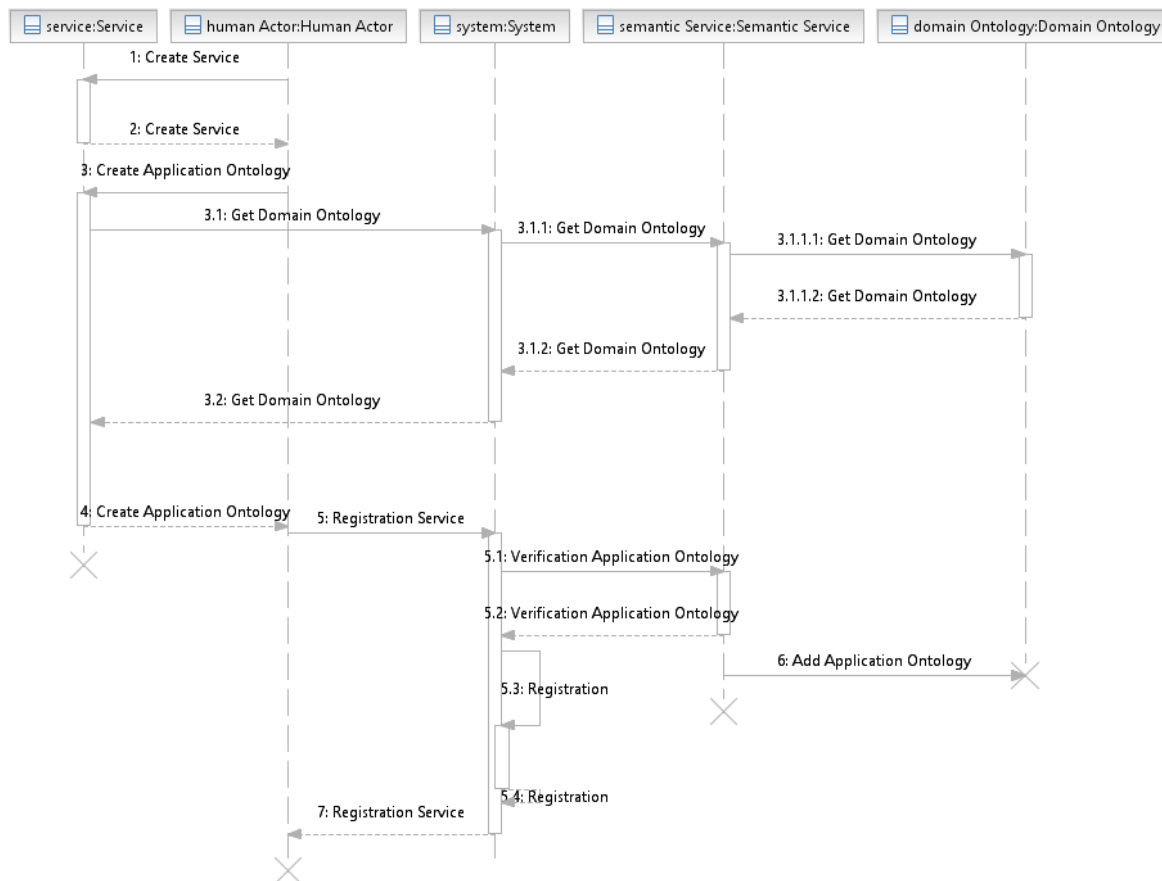


Рисунок 5 – Смешанный метод организации взаимодействия между элементами ССОА

### Список использованной литературы

1. Востряков О.В., Гребешкова О.М. Концепція управління знаннями в стратегічному процесі сучасного підприємства [Текст] / О.В. Востряков, О.М. Гребешкова // Стратегія розвитку України. Економіка, соціологія, право: наук. вісн. / Ін-т міжнар. відн. Нац. авіац. ун-ту. - 2009. - Вип. 1-2. - С. 419-427.
2. SOA и web-сервисы для новичков [Электронный ресурс]. – Режим доступа: <http://www.ibm.com/developerworks/ru/webservices/newto/>.
3. Гонтарь Н.А., Кудерметов Р.К. Организация семантической сервис-ориентированной архитектуры [Текст] / Н.А. Гонтарь, Р.К. Кудерметов // «Системный анализ и информационные технологии»: 14-й Международной научно-технической конференции SAIT 2012, Киев, 24 апреля 2012 г. / УНК «ИПСА» НТУУ «КПИ». – К. : УНК «ИПСА» НТУУ «КПИ», 2012. – С. 337-339.
4. Гонтарь, Н.А. Семантичний аспект сервис-орієнтованої архітектури [Текст] / Н.А. Гонтарь // Вісник Кременчуцького національного університету імені Михайла Остроградського. – 2013. – Вип. № 4(81) - С53-58.
5. Гонтарь Н.А. Моделирование семантической сервис-ориентированной архитектуры [Текст] / Гонтарь Н.А. // Наукові праці Донецького національного технічного університету Серія "Інформатика, кібернетика та обчислювальна техніка". - 2013. – Вип. №1(17). – С. 68-73.
6. Allemang D. Semantic web for the working ontologist modeling in RDF, RDFS and OWL [Text] / Dean Allemang, James A. Hendler. - Elsevier Inc., 2008 – 330 p.

Надійшла до редакції 20.03.2015

**Н.А. КУЛИКОВСЬКА**

Запорізький національний технічний університет, м. Запоріжжя

**МЕТОДИ ВЗАЇМОДІЇ КОМПОНЕНТ СЕМАНТИЧНОЇ СЕРВІС-ОРІЄНТОВАНОЇ  
АРХІТЕКТУРИ**

В даній роботі розглядаються методи взаємодії елементів семантичної сервіс-орієнтованої архітектури (ССОА). Дані методи необхідні при проектуванні сервіс-орієнтованих систем. До основних елементів ССОА відносять: множину сервісів, множину онтологій і семантичний сервіс. Множину онтологій представлено онтологіями сервісу, доменними онтологіями і онтологіями завдань. В залежності від стадії розвитку системи і ресурсів провайдера сервісу використовуються спадний, висхідний або змішаний метод взаємодії компонент ССОА. При цьому можна отримати перевагу або в часі, або в трудовитратах на розвиток системи залежно від обраного шляху.

**Ключові слова:** *метод, семантична сервіс-орієнтована архітектура, семантичний сервіс.*

**N.A. KULIKOVSKA**

Zaporizhzhya National Technical University, Zaporizhzhya

**METHODS OF INTERACTION OF THE COMPONENTS OF SEMANTIC SERVICE-ORIENTED  
ARCHITECTURE**

This paper discusses methods of interaction of the elements of semantic service-oriented architecture (SSOA). These methods are needed in the design of service-oriented systems. The main elements of SSOA include a set of services, a set of ontologies and semantic service. A set of ontologies represent ontologies of services, domain-ontologies and ontologies of tasks. Depending on the stage of system development and provider's resources a top-down, bottom-up or mixed method of component SSOA interaction can be used. The top-down method is entirely based on semantic service functions and is more suitable for the existing systems that have decided to go on SSOA. The advantage of this approach is more accurate and relevant search, because ontologies generate one aspect of the system. Time spent on registration service takes 10 iterations. Bottom-up approach is based on the work of the service provider, which develops its own application ontology. In this method, the service can be developed independently from other resources or their ontologies, in this architecture, in this construction it is relatively easy to make changes to the services, add new, etc. (improving quality characteristics of architecture: scalability, scalable bridge, the ability for evolutionary change). In this case, the elapsed time will be 11 iterations. Mixed design is a synthesis of the work and the failure of semantic service and provider service. The method is performed in 17 iterations. Thus, the method chosen can take advantage of either the time or effort required to develop a system.

**Keywords:** *method, semantic service-oriented architecture, semantic service.*