

---

---

## ОНТОЛОГИЧЕСКИЙ АНАЛИЗ WEB-СЕРВИСОВ В ИНТЕЛЛЕКТУАЛЬНЫХ СЕТЯХ

Анатолий Гладун, Юлия Рогушина, Виктор Штонда

**Аннотация:** *Определены подходы к анализу семантики Web-сервисов на основе их онтологий, ориентированные на автоматизацию их поиска и компоновки в интеллектуальных сетях. Проанализированы средства создания онтологических описаний Web-сервисов. Предложены алгоритмы формирования и усовершенствования таких онтологий, а также методы их сравнения.*

**Ключевые слова:** *Web-сервис, онтология, тезаурус, поиск информации.*

**ACM Classification Keywords:** *1.2.4 Knowledge Representation Formalisms and Methods*

---

### Введение

Центральным элементом зарождающейся сервис-ориентированной архитектуры (SOA) является способность разрабатывать новые приложения посредством композиции функциональных возможностей предприятия, представленных в форме сервисов - в пределах одной организации или же среди множества предприятий. Семантические описания сервисов, включая аннотации функциональных и не функциональных атрибутов, направлены на автоматизацию этого процесса и создание более качественных решений. Необходимо описать возможности сервисов на однозначно интерпретируемом, поддающемся компьютерной обработке языке и улучшают качество и устойчивость существующих задач, таких, как Web-сервисы и вызовы процедур. Одна из целей проекта Semantic Web – создать средства, позволяющие пользователям выбирать, использовать, компоновать и отслеживать такие Web-сервисы автоматически. Онтологии, в общем виде определяемые как совместно используемые формальные концепции конкретных предметных областей (ПрО), в состоянии сыграть критически важную роль в организации обработки знаний на базе Web, их совместного использования и обмена ими между приложениями.

---

### Концепция Web-сервисов

*Сервис-ориентированная архитектура (COA) – это парадигма проектирования, разработки и управления функциональных модулей (сервисов), каждый из которых доступен через сеть и способен выполнять определенные действия. COA создает коммуникационную среду для модулей, реализующих прикладную бизнес-логику. Под сервисом понимают некоторый Web-сайт, который предоставляет не просто статичную информацию, но также позволяет выполнять некоторые действия или изменять мир, например, продавать какие-либо продукты или управлять физическим устройством [1]. Web-сервисы обеспечивают высокоуровневые абстракции для того, чтобы организовать применение их в крупномасштабных, открытых средах. Web-сервис – это автономное приложения, предоставляющее средства доступа к информации внешним клиентам через набор предоставляемых им услуг. В [2] Web-сервис определяется как самоуправляемый модуль, способный обрабатывать данные и оперировать ими; он имеет возможность взаимодействовать с окружающей средой с помощью сообщений. В одном модуле соединены функции хранения данных и их обработки; сервис создается и внедряется независимо от других, поэтому может быть изменен без влияния на партнерские сервисы; сервис имеет четко*

определенные границы и представлен извне только через сообщения. Иначе говоря, сервис может быть определен как компонент ИТ, который обеспечивает выполнение взаимозависимых функций и общается с внешней инфраструктурой с помощью обмена сообщениями. Основная задача заключается в группировании функций и обеспечении их механизмами обмена сообщениями.

Сервисы могут быть простыми, или "примитивными", в том смысле, что они активизируют только какую-то одну доступную через Интернет программу, сенсор или устройство, не используя другие Web-сервисы, и поэтому взаимодействие между пользователем и сервисом, кроме простого вызова, отсутствует. Например, сервис, который возвращает почтовый код или широту и долготу, по указанному адресу. Они могут быть и составными, т.е. состоящими из нескольких примитивных. Такие сервисы обычно требуют взаимодействия или диалога между пользователем и сервисами, т.е. пользователь может осуществлять выбор или накладывать определенные условия. Пример – покупка книги через [www.amazon.com](http://www.amazon.com): пользователь ищет книги по различным критериям, а затем решает, покупать ли их и затем предоставляет кредитную карту или почтовый адрес.

Web-сервисы базируются на трех основных Web-стандартах: SOAP (Simple Object Access Protocol) — протоколе для отправки сообщений по протоколу HTTP и другим Интернет-протоколам; WSDL (Web Services Description Language) – языке для описания программных интерфейсов Web-сервисов; UDDI (Universal Description, Discovery and Integration) – стандарте индексации Web-сервисов

---

### **Постановка задачи**

---

Цель работы состоит в том, чтобы, проанализировав принципы и технологию сервис-ориентированных вычислений, области применения Web-сервисов, предложить методы анализа семантики Web-сервисов, направленные на автоматизацию их компоновки и основанные на онтологическом анализе. Кроме описания семантики самого сервиса, важно соотнести его с определенной ПрО, в терминах которой и определяются его возможности и ограничения. Для этого необходимо формализовать знания и представления разработчиков сервиса и представить их в форме онтологии, пригодной для автоматической обработки и опубликовать их, сделав доступными для потенциальных пользователей сервиса. Это вызывает потребность в методах создания, усовершенствования и анализа онтологий ПрО.

---

### **Компоновка Web-сервисов**

---

Возможность компоновки (composability) часто рассматривают как одно из основных преимуществ Web-сервисов. Компоновка Web-сервиса состоит из нахождения набора атомарных сервисов, необходимых для реализации запроса пользователя, и определение порядка их выполнения.

Компоновка Web-сервисов подобна проблеме планирования, которая исследовалась в искусственном интеллекте. Но классические планировщики непригодны для компоновки Web-сервисов, потому что предназначены для работы в статичном окружении (а среда Интернет динамична) и не имеют полной информации о системе, с которой работают (например, о возможностях различных сервисов).

Для автоматической компоновки программы должны уметь отбирать нужные им Web-сервисы и комбинировать их для достижения своих целей. Таким образом можно строить совершенно новые сервисы, комбинируя сервисы, уже имеющиеся в сети. Информация, содержащаяся в реестре UDDI, недостаточна для того, чтобы автоматически выполнить компоновку Web-сервисов, так как необходимы усилия человека для интерпретации семантики этих сервисов. Поэтому необходимо разрабатывать механизмы отображения семантики сервисов, запросов пользователей этих сервисов и их автоматизированного сопоставления с учетом специфики предметной области (ПрО), интересующей

пользователя. Для композиции Web-сервисов в динамичной среды необходимо иметь больше синтаксической и статической метаинформации о них: Web-сервисы в гетерогенной среде часто оказываются недействительными, меняются их версии или они заменяются другими сервисами.

---

### **Интеллектуальные сети**

---

Понятие интеллектуальных сетей ( Intelligent Networks, или IN ) возникло в связи с потребностью человечества в «умных» в сетях, которые могли бы сами подстраиваться под нужды пользователей, то есть разумно выполнять различные операции по хранению, обработке, передаче, систематизации информации, а также реализовать различные сервисы [3].

Интеллектуальные сети (IN) базируются на платформах традиционных телекоммуникационных сетей и включают интеллектуальную надстройку, обеспечивающую предоставление пользователям интеллектуальных сервисов различного назначения. В настоящее время телекоммуникационные и компьютерные приложения и услуги развиваются в направлении телематических услуг (мультилингвистические системы, интеллектуальные системы поиска информации, автоматизация бизнес-процессов, телемедицина, дистанционное обучение, системы обработки мультимедийной информации и др.). Кроме того, сегодня наблюдается процесс интеграции различных сетей (мобильных и наземных, специализированных сетей Ad-hoc, VPN с сетями общего (публичного использования (Internet, Frame Relay и т.д.).

Интеллектуальная сеть - это вид глобальной сети для которой характерно:

- широкое использование различных методов обработки информации;
- эффективное использование ресурсов сети;
- модульность функций сети;
- интегрированные возможности разработки и внедрения услуг средствами модульных и многоцелевых сетевых функций;
- гибкое распределение сетевых функций по физическим элементам сети;
- возможность перемещения сетевых функций из одного физического элемента сети в другой;
- стандартизованное взаимодействие сетевых функций посредством независимых от услуг сетевых интерфейсов;
- возможность управления некоторыми атрибутами сервисов пользователями;
- стандартизованное управление логикой сервисов.

Набор услуг продолжает постоянно пополняться, однако он еще недостаточно широк и находится сегодня на стадии динамического развития. Основная проблема заключается в том, чтобы найти и скомпоновать уже существующие услуги таким образом, чтобы решить задачу, стоящую перед пользователем.

В современных IN интеллектуальная надстройка использует онтологии, программные агенты и технологию Web-сервисов, позволяющие представлять и применять знания (рис.1). Так, целью проектов Semantic Web и Semantic Grid и является переход от традиционного Интернет к распределенной базе знаний, предлагающей пользователям интеллектуальные услуги и обрабатывающей контент (как статичный, так и динамичный) на семантическом уровне.

Интеллектуальные Web-сервисы призваны кардинально улучшить взаимодействие людей и информационных систем, представить пользователям и корпорациям интеллектуальные услуги, обеспечить взаимное проникновение разных систем и процессов и обеспечивают новый этап в развитии современных интеллектуальных сетей.

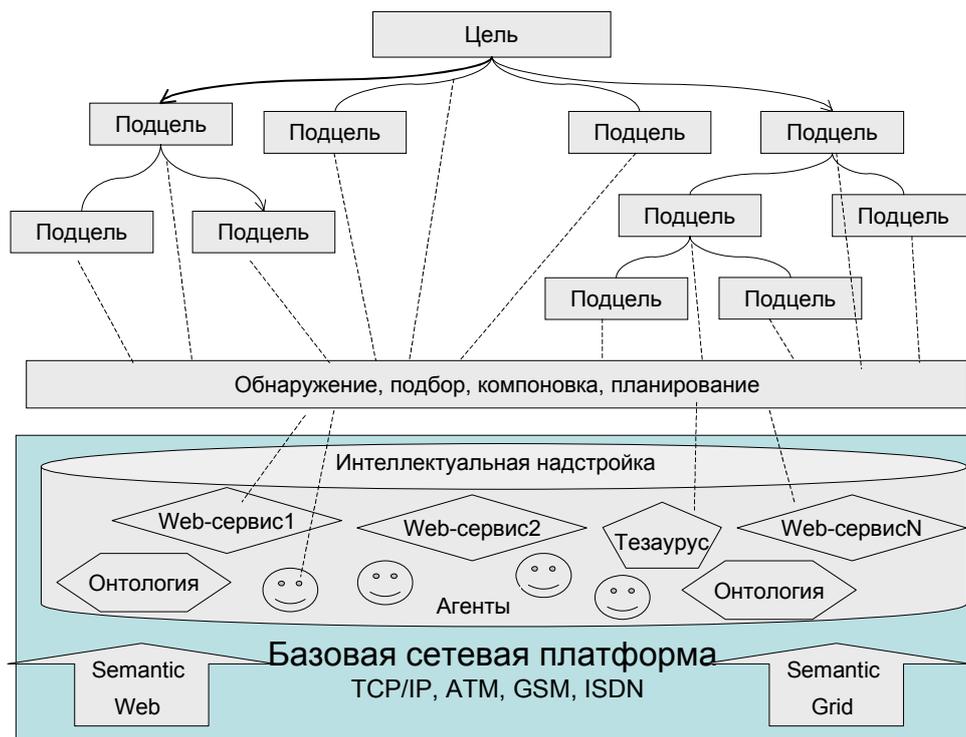


Рис. 1 Модель реализации интеллектуальных услуг на основе IN

### Онтология OWL-S: семантическая разметка Web-сервисов

Чтобы использовать Web-сервис, программные агенты нуждаются в интерпретируемом компьютером определении сервиса и средств доступа к нему. Важной целью языков разметки проекта Semantic Web является установление структуры, в которой такие описания можно создавать и распространять. Web-сайты должны использовать набор базовых классов и свойств для декларирования и описания сервисов, и именно онтологические механизмы структурирования OWL обеспечивают подобную структуру.

OWL-S предназначен для того, чтобы помочь пользователям и поисковым агентам, обнаруживать, вызывать, компоновать и контролировать Web-сервисы [4]. Цель OWL-S – сделать возможным использование логического вывода для Web-сервисов, планирование компоновки Web-сервисов, автоматическое использование сервисов программными агентами.

OWL-S – это онтология сервисов, реализующая эту функциональность. Толчком к его разработке стал проект Semantic Web, который стремится обеспечить доступ к IP Web не по ключевым словам, а по контенту. Важным результатом в этом направлении стала разработка языков разметки Web, таких как OWL и его предшественник DAML+OIL. Эти языки позволяют создавать онтологии любой Про и устанавливать связи с этими онтологиями Web-сайтов для их описания.

OWL-S обеспечивает декларативные описания свойств Web-сервиса и возможности, которые могут использоваться для автоматического обнаружения сервиса, декларативный API для автоматизированного выполнения Web-сервисов, которые являются необходимыми для Web-сервисов. Структурирование онтологии сервисов связано с тремя типами знаний о сервисах:

- Профиль сервиса – какую информацию сервис требует от пользователя или агента и что сервис ему предлагает: класс SERVICE представляет SERVICEPROFILE;

- Модель сервиса – как он работает: класс SERVICE описывается в SERVICEMODEL;
- Основание – как это использовать: класс SERVICE поддерживается в SERVICEGROUNDING.

Для каждого опубликованного сервиса должен существовать один экземпляр класса SERVICE.

Профиль сервиса строится на основе контента UDDI, описывающем свойства сервиса, необходимые для его автоматического обнаружения, такие, например, как предложение сервиса, его входы и выходы, предварительные условия и дополнительные действия. На основе профиля, который представляет информацию о провайдере, функциональных возможностях, и функциональных атрибутах сервиса, могут быть созданы описания и запросы сервиса. Профиль Web-сервиса дает описание его свойств: категорию сервиса (например, по классификации UNSPSC) и его качественную оценку (скорость, надежность и т.п.). OWL-S решает задачи четырех типов:

**Автоматическое обнаружение Web-сервисов.** Автоматическое обнаружение (discovery) Web-сервисов включает в себя автоматическое определение местонахождения Web-сервисов, которые предоставляют определенную услугу и удовлетворяют наложенным ограничениям. Например, пользователь хочет найти сервис по продаже авиабилетов между двумя определенными городами, принимающий его кредитную карту. Если сервис размечен при помощи OWL-S, то семантическая информация, необходимая для нахождения сервиса, представлена на сайте Web-сервиса в форме, интерпретируемой компьютером, и тогда поисковая система, способная обрабатывать онтологическую информацию, может обнаружить такой сервис автоматически. Альтернативный подход заключается в том, что сервер может проактивно помещать в некоторые реестры сервисов OWL-S-описание предоставляемого им сервиса, а такие реестры могут быть найдены по соответствующему запросу. Таким образом, OWL-S должен обеспечивать декларативное представление свойств и возможностей сервиса, которые могут использоваться для автоматического обнаружения этого сервиса.

**Автоматический вызов Web-сервиса.** Автоматический вызов (invocation) Web-сервиса заключается в автоматическом выполнении идентифицированного ранее Web-сервиса компьютерной программой или агентом. Например, пользователь может запросить покупку с определенного сайта авиабилета на конкретный рейс. Если не использовать Web-сервис, то пользователь должен выполнить вручную набор действий: зайти на этот сайт, заполнить определенную форму и нажать на кнопку для выполнения сервиса или же отправить запрос с соответствующими параметрами непосредственно к сервису. Выполнение Web-сервиса можно рассматривать как последовательность вызовов функций. Разметка OWL-S Web-сервиса должна обеспечивать декларативный, интерпретируемый компьютером API для выполнения этих функциональных вызовов. Программный агент должен быть способен интерпретировать разметку, чтобы понять, какие нужны входные данные для вызова сервиса, какая информация будет возвращена и как выполнить сервис автоматически. Таким образом, OWL-S должен обеспечивать декларативные API для Web-сервисов, которые нужны для их автоматического выполнения.

1. **Автоматическая композиция и взаимодействие Web-сервисов.** Эта задача включает в себя автоматический выбор, композицию и взаимодействие Web-сервисов для выполнения определенной задачи, обусловленной высокоуровневым описанием задания. Например, пользователь хочет спланировать все мероприятия, связанные с поездкой на конференцию. Обычно пользователь должен выбрать все нужные Web-сервисы, вручную задать порядок их выполнения и удостовериться, что все программное обеспечение, необходимое для взаимодействия, соответствует его требованиям. Но если Web-сервис размечен при помощи OWL-S, то информация, необходимая для выбора и композиции сервисов, представлена на их сайтах. Соответствующее программное обеспечение может манипулировать этими данными, а также спецификациями целей задачи, для автоматического выполнения задачи. Таким образом, OWL-S должен обеспечивать декларативные спецификации

предварительных условий и результатов выполнения отдельных сервисов, необходимые для автоматической композиции и взаимодействия сервисов.

2. **Автоматический мониторинг выполнения Web-сервисов.** Отдельные сервисы, и, более того, композиции сервисов, обычно требуют некоторого времени для выполнения. Пользователь может захотеть в этот период узнать, каков статус его запроса, или же его планы могут измениться. Например, пользователь хочет удостовериться в том, что резервирование отеля успешно выполнено. Для этого было бы полезно иметь возможность выяснять, на каком этапе находится процесс выполнения запроса и не возникли ли какие-либо непредвиденные помехи. Таким образом, OWL-S должен обеспечивать декларативные описания для состояния выполняемых сервисов.

---

### **Онтология ПрО и Web-сервисы**

---

Наличие явного представления знаний о ПрО, к которой относится сервис, допускает переформулировку запросов контекстно-зависимым способом и переговоры о возможностях этого сервиса. Наиболее распространенным на сегодня механизмом представления знаний о ПрО являются онтологии [5]. Одна и та же ПрО может иметь несколько онтологий, поскольку информация о ПрО доступна даже экспертам лишь частично. Использование при поиске семантики ПрО обеспечивает более корректную компоновку Web-сервисов. Описывая определенный Web-сервис для публичного использования, его провайдеры должны связывать его с определенными онтологиями, которые позволяют формализовать терминологию соответствующей ПрО. Например, назначение сервиса описано как "оценка квартиры". При этом во входных параметрах используются такие термины, как адрес квартиры, метраж и количество комнат. Но пользователю необходимо пояснить, что "метраж" - это "общая площадь", а не "жилая площадь". Другой пример - при обращении к сервису покупки автомобиля или компьютера пользователю надо предложить структуру параметров (таксономию либо более сложную конструкцию), характеризующих это устройство (причем приводить несколько видов альтернативных названий). При этом можно соотносить друг с другом сервисы различных провайдеров для планирования последовательности их совместного использования.

Алгоритмы нахождения соответствия между запросом и сервисом, использующие онтологическое представление знаний, позволяют автоматизировать нахождение семантического подобия между запросом и описанием сервиса, несмотря на синтаксические различия между ними. Для этого запрос согласовывается на основе иерархии понятий ПрО, отображенной в онтологии [6]. Например, запрос об автомобилях соответствует объявлению о транспортных средствах, так как автомобили включены в категорию "транспортные средства". Соответствие между описанием Web-сервиса и запросом обнаруживается, когда все выходы запроса согласованы с выходами описания, и все входы описания – со всеми входами запроса, то есть когда сервис способен удовлетворить потребности запрашивающей стороны, и запрашивающая сторона обеспечивает все входы согласованных сервисных потребностей в его действиях.

---

### **Формирование онтологий ПрО**

---

Изложенные выше подходы оставляют открытыми вопросы о том, кто и каким образом формирует онтологии ПрО, с понятиями которых связаны имена параметров Web-сервиса, и как строится описание Web-сервиса в OWL-S (разработчики и провайдеры Web-сервисов не обязаны владеть онтологическим анализом и знать инструментальные средства создания онтологий). В связи с этим актуальной задачей представляется разработка средств и методов автоматизированного формирования онтологий по информационным ресурсам, соответствующим определенному Web-сервису. Важным вопросом является

также создание общего словаря (тезауруса) ПрО, обеспечивающего взаимопонимание пользователей и разработчиков Web-сервисов. Кроме того, очень актуальна разработка эффективных алгоритмов сравнения онтологий, которые, возможно, являются различными концептуализациями одной и той же ПрО – для нахождения соответствия между онтологиями пользователей и разработчиков Web-сервисов. В ряде случаев пользователи и разработчики могут воспользоваться готовыми онтологиями, но поиск таких онтологий также является нетривиальной задачей при нестандартном терминологическом базисе. Иногда ПрО приложения настолько специфична, что требует значительного уточнения и расширения для уровня приложения самим пользователем. Предлагается использование методов индуктивного обобщения для автоматизированного извлечения онтологических знаний о ПрО из набора информационных ресурсов, релевантных этой ПрО.

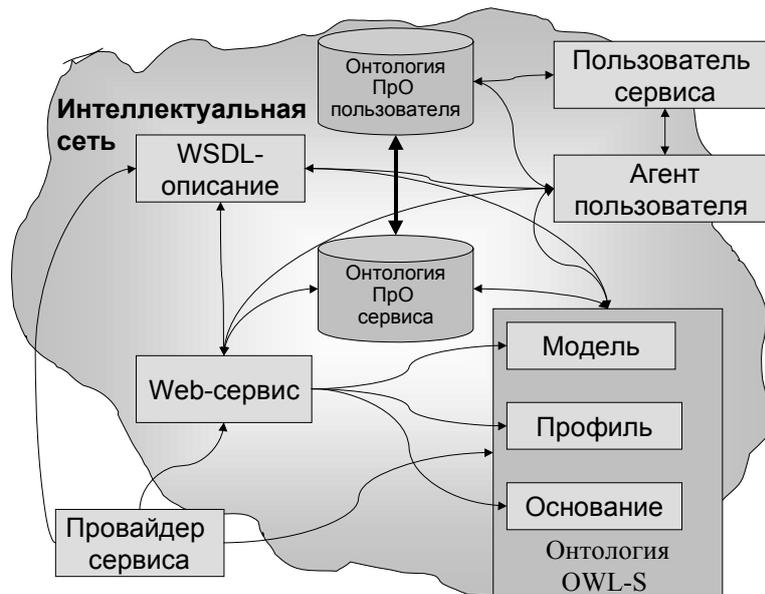


Рис.1. Средства задания семантики Web-сервиса

## Методы сопоставления онтологий

При наличии онтологических описаний как Web-сервисов, так и запросов пользователей возникает проблема сравнения этих описаний. Если Web-сервисы и запросы ссылаются на одну онтологию, то можно легко установить, связаны имена параметров атомарных сервисов с одним понятием онтологии или с разными. В противном случае необходимо установить, являются ли понятия различных онтологий эквивалентными (например, синонимами) или находятся в иерархических отношениях (например, являются подклассом). В общем случае эта довольно сложная задача, которая имеет высокую вычислительную сложность. Наш подход к ее решению базируется на предположении, которое в поисках информации используются относительно небольшие и простые за структурой фрагменты онтологий, которые характеризуют семантику конкретных ИП и запросов пользователей. Алгоритм сравнения таких онтологий состоит из следующих этапов:

1. Строится пересечения терминов онтологий Web-сервиса и запроса  $T(O) = T(O_s) \cap T(O_q)$ .
2. Если это пересечение не пусто, для каждого термина из  $T(O)$  строятся два множества  $T_s$  и  $T_q$ - термины, которые связанные с ним в каждой онтологии любыми отношениями.
3. Для каждого термина из  $T(O)$  строится пересечение множеств  $T_s$  и  $T_q$ .

4. Анализ типов отношений между терминами из  $T(O)$  и пересечения множеств  $T_s$  и  $T_q$  (все отношения онтологии делятся на три типа - иерархические, синонимические и прочие).

5. Строится коэффициент сходства онтологий, который является количественным отображением сходства семантики двух онтологий. При этом учитываются следующие факторы: вхождение одного и того же термина в обе онтологии; то, что два термина находятся в разных онтологиях в одном и том же отношении; то, что два термина находятся в разных онтологиях в отношениях одного типа или разных (например, в иерархическом отношении и отношении синонимии); существуют ли вообще любые отношения (прямые или опосредствованные) между одними и теми же терминами. Для этого используются статистические методы, нечеткую логику, интенциональные отношения и эмпирические правил.

6. Строится коэффициент подобия запроса и Web-сервиса – аналогично п.5, но учитываются только термины из  $T(O) = T(O_s) \cap T(O_q)$ , на которые ссылаются имена параметров Web-сервиса. Если полученный коэффициент выше определенного пользователем коэффициента доверия, то считается, что Web-сервис удовлетворяет потребностям пользователя и может использоваться при компоновке составного Web-сервиса.

---

## Выводы

---

Рассмотрев базовые составляющие сервис-ориентированных вычислений в распределенной среде Интернет и проанализировав перспективы их развития, можно сделать выводы о том, что автоматизация компоновки Web-сервисов, которая должна обеспечить их значительно более широкое применение, должна базироваться на семантическом описании их функциональных возможностей. Сегодня описание семантики Web-сервисов, как и многих других информационных ресурсов распределенной гетерогенной среды Интернет, связывают с онтологическим подходом к представлению знаний (OWL-S). Однако открытыми остаются вопросы как создания онтологий, адекватно отражающих специфику определенных ПрО, так и проблемы, связанные со сравнением и установлением соответствий между различными онтологиями. В данной работе предложены подходы к установлению подобия между онтологиями, характеризующими Web-сервисы и потребности пользователей.

---

## Литература

---

1. Cowles P. Web Services and the Semantic Web. – <http://www.sys-con.com/webservices/article.cfm?id=419>.
2. Christensen E., Curbera F., Meredith G. Web services description language (WSDL) 1.1, 2001 [www.w3.org/TR/wsdl](http://www.w3.org/TR/wsdl).
3. Гладун А.Я., Несен М.В., Штонда В.Н. Интеллектуальные агентно-ориентированные услуги, базирующиеся на платформах интеллектуальных сетей // Компьютерные средства, сети и системы, 2004, №6, с. 112-122.
4. OWL-S: Semantic Markup for Web Services. – <http://www.daml.org/services/owl-s/1.0/owl-s.html>.
5. Клещев А.С., Артемьева И.Л. Отношения между онтологиями предметных областей. Ч. 1. // Информационный анализ, Выпуск 1, С.2, 2002. – С.4-9.
6. Рогушина Ю.В., Гладун А.Я. Онтологический подход к мультилингвистическому анализу информационных ресурсов в сети Интернет // Сб.трудов VI междунар. конф. "Интеллектуальный анализ информации ИАИ-2006", К.: Просвіта, 2006. – С.237-246.

---

**Информация об авторах**

---

**Гладун Анатолий Ясонович** – Международный научно-учебный центр информационных технологий и систем НАНУ, Киев-187 ГСП, 03680, просп. акад. Глушкова, 40, email: [glanat@yahoo.com](mailto:glanat@yahoo.com)

**Рогущина Юлия Витальевна** – Институт программных систем НАНУ, Киев-187 ГСП, 03680, просп. акад. Глушкова, 40, Киевский Славистический Университет, email: [jjj\\_@ukr.net](mailto:jjj_@ukr.net)

**Штонда Виктор Николаевич** – Издательство компьютерной литературы «Диалектика», Киев-157, 03412, просп. акад. Глушкова, 4, Киевский Государственный Университет им. Т.Г.Шевченко, email: [shtonda@dialektika.com](mailto:shtonda@dialektika.com)