Оглавление

[Введение 1](#_Toc421524017)

[Часть 1. Теория 3](#_Toc421524018)

[История развития идеи 3](#_Toc421524019)

[Структура семантических сетей. 4](#_Toc421524020)

[Графическое представление 4](#_Toc421524021)

[Математическая запись 5](#_Toc421524022)

[Лингвистическая запись 5](#_Toc421524023)

[Семантические отношения 6](#_Toc421524024)

[Иерархические 6](#_Toc421524025)

[Вспомогательные 7](#_Toc421524026)

[Семантическая паутина 8](#_Toc421524027)

[Использование семантических сетей и онтологий. 11](#_Toc421524028)

[Архитектура семантической паутины 11](#_Toc421524029)

[Семантические модели и их применения 13](#_Toc421524030)

[Использование технологий Semantic Web для интеграции информационных систем 15](#_Toc421524031)

[Сравнительный анализ семантических моделей для корпоративных приложений 21](#_Toc421524032)

[Постановка проблемы 21](#_Toc421524033)

[Пример: создание и анализ «обычной» семантической модели 22](#_Toc421524034)

[Пример: создание и анализ модели по стандарту ISO 15926 24](#_Toc421524035)

[Сходства и различия «обычной» семантики и ISO 15926 27](#_Toc421524036)

[Семантический рабочий стол 30](#_Toc421524037)

[Часть 2. Ключевые технологии 33](#_Toc421524038)

[Основная идея 33](#_Toc421524039)

[Semantic Desktop 33](#_Toc421524040)

[Принципы 34](#_Toc421524041)

[Один компьютер – один пользователь. 34](#_Toc421524042)

[У всего есть URI. 34](#_Toc421524043)

[Вся информация в виде RDF. 36](#_Toc421524044)

[Целевые приложения. 38](#_Toc421524045)

[Ассоциативные связи между ресурсами. 39](#_Toc421524046)

[Онтологии. 40](#_Toc421524047)

[Совместное использование 41](#_Toc421524048)

[Снова о URL 41](#_Toc421524049)

[URL и URI – Расположение и Идентификация. 42](#_Toc421524050)

[Кризис URI. 42](#_Toc421524051)

[Различные URI 43](#_Toc421524052)

[Различные контексты. 43](#_Toc421524053)

[Проблемы 44](#_Toc421524054)

[Часть 3. Приложение Semantic Desktop 44](#_Toc421524055)

[Заключение 45](#_Toc421524056)

# 

# Введение

В основе исследований в области искусственного интеллекта лежит подход, связанный со знаниями. Опора на знания — базовая парадигма искусственного интеллекта и экспертных систем. Как и многие фундаментальные научные категории (например, алгоритм, интеллект, деятельность и т. д.), понятие «знание» относится к интуитивно определяемым. В Большой Советской Энциклопедии дается следующее его толкование: «Знание — проверенный практикой результат познания действительности, верное ее отражение в сознании человека. Знания бывают житейскими, донаучными, художественными, научными (теоретическими и эмпирическими)». Знания о некоторой предметной области представляют собой совокупность сведений об объектах этой предметной области, их существенных свойствах и связывающих их отношениях, процессах, протекающих в данной предметной области, а также методах анализа, возникающих в ней ситуаций и способах разрешения, ассоциируемых с ними проблем.

В «Словаре русского языка» Ожегова знание определяется как «постижение действительности сознанием» и «совокупность сведений, познаний в какой-нибудь области». Интерпретация знаний как «совокупности сведений, образующих целостное описание, соответствующее некоторому уровню осведомленности об описываемом вопросе, предмете, проблеме и т. д.» дана в толковом словаре искусственного интеллекта. Семантические сети же являются моделью представления знаний.

Семантическая сеть – это [информационная модель](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C) [предметной области](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%BC%D0%B5%D1%82%D0%BD%D0%B0%D1%8F_%D0%BE%D0%B1%D0%BB%D0%B0%D1%81%D1%82%D1%8C) (domain model), имеющая вид ориентированного [графа](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_%28%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0%29), вершины которого соответствуют сущностям (объектам) предметной области, а дуги (рёбра) задают отношения между ними и называются предикатами. Объектами могут быть понятия, события, свойства, процессы, ресурсы, документы. Таким образом, семантическая сеть является одним из способов [представления знаний](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D1%81%D1%82%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B7%D0%BD%D0%B0%D0%BD%D0%B8%D0%B9). В названии соединены термины из двух наук: [семантика](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D0%BA%D0%B0) в языкознании изучает смысл единиц языка, а сеть в математике представляет собой разновидность [графа](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_%28%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0%29) — набора вершин, соединённых ребрами которым присвоено некоторое число. В семантической сети роль вершин выполняют [понятия](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BD%D1%8F%D1%82%D0%B8%D0%B5) базы знаний, а дуги (причем направленные) задают [отношения](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BD%D0%BE%D1%88%D0%B5%D0%BD%D0%B8%D0%B5) между ними. Таким образом, семантическая сеть отражает семантику предметной области в виде понятий и отношений.

Неправильно приравнивать друг другу понятия «Семантическая сеть» (Semantic Network) и «[Семантическая паутина](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%BF%D0%B0%D1%83%D1%82%D0%B8%D0%BD%D0%B0)» (Semantic Web). Семантическая паутина является частным случаем семантической сети, а именно семантической сетью, вершины графа которой представляют собой веб ресурсы в сети интернет и описанные на языке RDF или OWL.

# Часть 1. Теория

## История развития идеи

Идея систематизации знаний на основе каких-либо семантических отношений предлагалась учеными еще до начала научно технической революции. Примером этого может служить [биологическая классификация](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0) [Карла Линнея](https://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D0%BD%D0%BD%D0%B5%D0%B9,_%D0%9A%D0%B0%D1%80%D0%BB) [1735 года.](https://ru.wikipedia.org/wiki/1735) Если рассматривать её как семантическую сеть, то в данной классификации используется отношение подмножества, современное [AKO](https://ru.wikipedia.org/w/index.php?title=AKO&action=edit&redlink=1) (от англ. «A Kind Of», «разновидность, экземпляр»). Такое отношение еще называют «IsA»

Прародителями современных семантических сетей можно считать [экзистенциальные графы](https://ru.wikipedia.org/w/index.php?title=%D0%AD%D0%BA%D0%B7%D0%B8%D1%81%D1%82%D0%B5%D0%BD%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%B3%D1%80%D0%B0%D1%84&action=edit&redlink=1), предложенные [Чарльзом Пирсом](https://ru.wikipedia.org/wiki/%D0%9F%D0%B8%D1%80%D1%81,_%D0%A7%D0%B0%D1%80%D0%BB%D1%8C%D0%B7_%D0%A1%D0%B0%D0%BD%D0%B4%D0%B5%D1%80%D1%81) в [1909 г](https://ru.wikipedia.org/wiki/1909). Они использовались для представления [логических](https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D0%B8%D0%BA%D0%B0) высказываний в виде особых диаграмм. Пирс назвал этот способ «логикой будущего».

Важным направлением в исследовании сетей стали работы немецкого психолога Отто Зельца 1913 и 1922 гг. В них для организации структур понятий и ассоциаций, а также изучения методов наследования свойств он использовал графы и семантические отношения. Исследователи [Дж. Андерсон](https://ru.wikipedia.org/w/index.php?title=%D0%90%D0%BD%D0%B4%D0%B5%D1%80%D1%81%D0%BE%D0%BD,_%D0%94%D0%B6%D0%BE%D0%BD_%D0%A0%D0%BE%D0%B1%D0%B5%D1%80%D1%82&action=edit&redlink=1) (1973), [Д. Норман](https://ru.wikipedia.org/wiki/%D0%9D%D0%BE%D1%80%D0%BC%D0%B0%D0%BD,_%D0%94%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%B4) (1975) и другие использовали эти работы для моделирования человеческой памяти и интеллектуальных свойств.

Компьютерные семантические сети были детально разработаны Ричардом Риченсом в [1956 году](https://ru.wikipedia.org/wiki/1956_%D0%B3%D0%BE%D0%B4) в рамках проекта Кембриджского центра изучения языка по [машинному переводу](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BF%D0%B5%D1%80%D0%B5%D0%B2%D0%BE%D0%B4). Процесс машинного перевода подразделяется на две части: перевод исходного текста в промежуточную форму представления, а затем эта промежуточная форма транслируется на нужный язык. Такой промежуточной формой как раз и были семантические сети. В 1961 г. появилась работа [Мастерман](https://en.wikipedia.org/wiki/Margaret_Masterman), в которой она, в частности, определяла базовый словарь для 15000 понятий. Эти исследования были продолжены Робертом Симмонсом (1966), [Й. Уилксом](https://ru.wikipedia.org/wiki/%D0%A3%D0%B8%D0%BB%D0%BA%D1%81,_%D0%99%D0%BE%D1%80%D0%B8%D0%BA) (1972) и другими учёными.

Труды по семантическим сетям часто ссылаются на работу американского психолога Росса Квиллиана о «семантической памяти». В этой работе описываются так называемые ассоциативные карты памяти (Mind Map), которые имею структуру подобную семантической сети.

## Структура семантических сетей.

Математика позволяет описать большинство явлений в окружающем мире в виде логических высказываний. Семантические сети возникли как попытка визуализации математических формул. Основным представлением для семантической сети является [граф](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_%28%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0%29). Однако не стоит забывать, что за графическим изображением непременно стоит строгая математическая запись, и что обе эти формы отображают одно и то же.

## Графическое представление

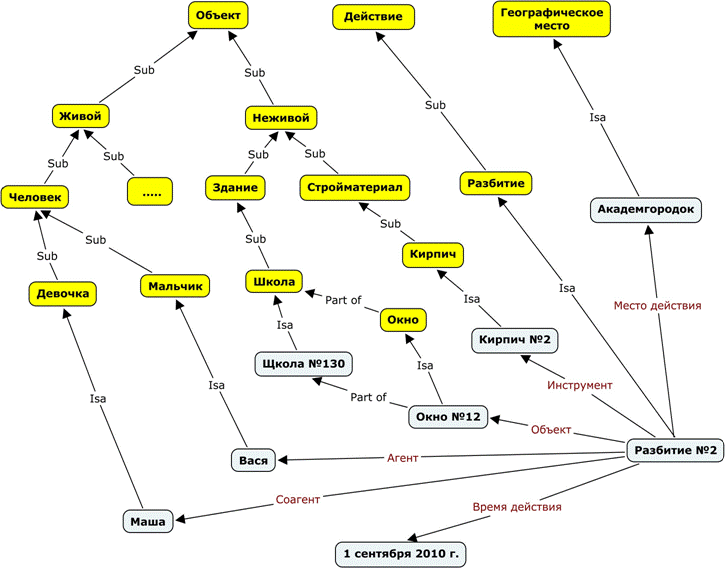
Основной формой представления семантической сети является граф. Понятия семантической сети записываются в овалах или прямоугольниках и соединяются стрелками с подписями — дугами. Это наиболее удобно воспринимаемая человеком форма. Её недостатки проявляются, когда мы начинаем строить более сложные сети или пытаемся учесть особенности естественного языка. Даже для небольших сетей размер графического изображения получается слишком большим и практически нечитаемым. Схемы семантических сетей, на которых указаны направления навигационных отношений, называют картами знаний, а их совокупность, позволяющая охватить большие участки семантической сети, атласом знания. 

Рисунок 1. Пример графического представления семантического графа

## Математическая запись

В математике [граф](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_%28%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0%29) представляется множеством вершин V и множеством отношений между ними E. Используя аппарат математической логики, приходим к выводу, что каждая вершина соответствует элементу предметного множества, а дуга — [предикату](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%B8%D0%BA%D0%B0%D1%82).

## Лингвистическая запись

В лингвистике отношения фиксируются в словарях и в [тезаурусах](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%B7%D0%B0%D1%83%D1%80%D1%83%D1%81). В словарях в определениях через род и видовое отличие родовое понятие занимает определённое место. В тезаурусах в статье каждого термина могут быть указаны все возможные его связи с другими родственными по теме терминами. От таких тезаурусов необходимо отличать тезаурусы информационно- поисковые с перечнями ключевых слов в статьях, которые предназначены для работы дескрипторных поисковых систем.

## Семантические отношения

Количество типов отношений в семантической сети определяется её создателем, исходя из конкретных целей. В реальном мире их число стремится к бесконечности. Каждое отношение является, по сути, предикатом, простым или составным. Скорость работы с базой знаний зависит от того, насколько эффективно реализованы программы обработки нужных отношений.

### Иерархические

Наиболее часто возникает потребность в описании отношений между элементами, множествами и частями объектов. Отношение между объектом и множеством, обозначающим, что объект принадлежит этому множеству, называется отношением классификации (ISA). Говорят, что множество (класс) классифицирует свои экземпляры. (пример: «Шарик является собакой» = Шарик является объектом типа собака). Иногда это отношение именуют также MemberOf, InstanceOf или подобным образом. Связь ISA предполагает, что свойства объекта наследуются от множества. Обратное к ISA отношение используется для обозначения примеров, поэтому так и называется — «Example», или по-русски «Пример». Иерархические отношения образуют [древовидную структуру](https://ru.wikipedia.org/wiki/%D0%94%D1%80%D0%B5%D0%B2%D0%BE%D0%B2%D0%B8%D0%B4%D0%BD%D0%B0%D1%8F_%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0).

Отношение между надмножеством и подмножеством (называется AKO — «A Kind Of», «разновидность»). (Пример: «собака является животным» = тип с именем собака является подтипом типа животные). Элемент подмножества называется [гипонимом](https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%BF%D0%BE%D0%BD%D0%B8%D0%BC) (собака), а надмножества — [гиперонимом](https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%BF%D0%B5%D1%80%D0%BE%D0%BD%D0%B8%D0%BC) (животное), а само отношение называется отношением гипонимии. Альтернативные названия — «SubsetOf» и «Подмножество». Это отношение определяет, что каждый элемент первого множества входит и во второе (выполняется ISA для каждого элемента), а также логическую связь между самими подмножествами: что первое не больше второго и свойства первого множества наследуются вторым. Отношение АКО (Род-Вид) часто используется для навигации в информационном пространстве.

Объект, как правило, состоит из нескольких частей, или элементов. Например, компьютер состоит из системного блока, монитора, клавиатуры, мыши и т. д. Важным отношением является HasPart, описывающее связь частей и целого — отношение меронимии. В этом случае свойства первого множества не наследуются вторым. [Мероним и холоним](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%80%D0%BE%D0%BD%D0%B8%D0%BC_%D0%B8_%D1%85%D0%BE%D0%BB%D0%BE%D0%BD%D0%B8%D0%BC) — противоположные понятия:

Мероним — объект, являющийся частью для другого. (Двигатель — мероним автомобиля.)

Холоним — объект, который включает в себя другое. (Например, у дома есть крыша. Дом — холоним крыши. Компьютер — холоним монитора.)

Часто в семантических сетях требуется определить отношения [синонимии](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BD%D0%BE%D0%BD%D0%B8%D0%BC) и [антонимии](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D1%82%D0%BE%D0%BD%D0%B8%D0%BC). Эти связи либо дублируются явно в самой сети, либо определяются алгоритмической составляющей.

### Вспомогательные

В семантических сетях часто используются также следующие отношения:

* функциональные связи (определяемые обычно глаголами «производит», «влияет»…);
* количественные (больше, меньше, равно…);
* пространственные (далеко от, близко от, за, под, над…);
* временные (раньше, позже, в течение…);
* атрибутивные (иметь свойство, иметь значение);
* логические (И, ИЛИ, НЕ);
* лингвистические.

Этот список может сколь угодно продолжаться: в реальном мире количество отношений огромно. Например, между понятиями может использоваться отношение «совершенно разные вещи» или подобное: Не\_имеют\_отношения\_друг\_к\_другу (Солнце, Кухонный\_чайник).

## Семантическая паутина

Концепция организации [гипертекста](https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%BF%D0%B5%D1%80%D1%82%D0%B5%D0%BA%D1%81%D1%82) напоминает однородную бинарную семантическую сеть, однако здесь есть существенное отличие:

Связь, осуществляемая гиперссылкой, не имеет семантики, то есть не описывает смысла этой связи. Назначение семантической сети состоит в том, чтобы описать взаимосвязи объектов, а не дополнительную информацию по предметной области. Человек может разобраться, зачем нужна та или иная гиперссылка, но компьютеру эта связь не понятна.

Страницы, связываемые гиперссылками, являются документами, описывающими, как правило, проблемную ситуацию в целом. В семантической сети вершины (то, что связывают отношения) представляют собой понятия или объекты реального мира.

Попытка создания семантической сети на основе [Всемирной паутины](https://ru.wikipedia.org/wiki/%D0%92%D1%81%D0%B5%D0%BC%D0%B8%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D1%83%D1%82%D0%B8%D0%BD%D0%B0) получила название семантической паутины. Эта концепция подразумевает использование языка [RDF](https://ru.wikipedia.org/wiki/RDF) (языка описания структуры на основе [XML](https://ru.wikipedia.org/wiki/XML)) и призвана придать ссылкам и UNI адресам некий смысл, понятный [компьютерным](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) системам. Это позволит превратить [Интернет](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82) в распределённую [базу знаний](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B7%D0%BD%D0%B0%D0%BD%D0%B8%D0%B9) глобального масштаба.

Семанти́ческая паути́на (Semantic Web) — это общедоступная глобальная [семантическая сеть](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C), формируемая на базе [Всемирной паутины](https://ru.wikipedia.org/wiki/%D0%92%D1%81%D0%B5%D0%BC%D0%B8%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D1%83%D1%82%D0%B8%D0%BD%D0%B0) путём стандартизации представления [информации](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F) существующий в сети в виде, пригодном для машинной обработки.

В обычной Всемирной паутине, основанной на [HTML](https://ru.wikipedia.org/wiki/HTML)-страницах, информация заложена в тексте страниц и предназначена для чтения и понимания человеком. Семантическая паутина состоит из машинно-читаемых элементов — узлов [семантической сети](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C), с опорой на [онтологии](https://ru.wikipedia.org/wiki/%D0%9E%D0%BD%D1%82%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%8F_%28%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0%29). Благодаря этому, программы-[клиенты](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82_%28%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0%29) получают возможность непосредственно получать из интернета утверждения вида «предмет — вид взаимосвязи — другой предмет» и вычислять по ним [логические заключения](https://ru.wikipedia.org/wiki/%D0%92%D1%8B%D0%B2%D0%BE%D0%B4_%28%D1%80%D0%B0%D1%81%D1%81%D1%83%D0%B6%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5%29). Семантическая паутина работает параллельно с обычной Всемирной паутиной и на её основе, используя протокол [HTTP](https://ru.wikipedia.org/wiki/HTTP) и идентификаторы ресурсов [URI](https://ru.wikipedia.org/wiki/URI).

Название «Семантическая паутина» было впервые введено [сэром Тимом Бернерсом-Ли](https://ru.wikipedia.org/wiki/%D0%91%D0%B5%D1%80%D0%BD%D0%B5%D1%80%D1%81-%D0%9B%D0%B8,_%D0%A2%D0%B8%D0%BC) (изобретателем Всемирной паутины) еще в сентябре 1998 года, и называется им «следующим шагом в развитии Всемирной паутины». Позже в своём блоге он предложил в качестве синонима термин «гигантский глобальный граф» ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) giant global graph, GGG, по аналогии с WWW). Концепция семантической паутины была принята и в данный момент активно продвигается [консорциумом Всемирной паутины](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%81%D0%BE%D1%80%D1%86%D0%B8%D1%83%D0%BC_%D0%92%D1%81%D0%B5%D0%BC%D0%B8%D1%80%D0%BD%D0%BE%D0%B9_%D0%BF%D0%B0%D1%83%D1%82%D0%B8%D0%BD%D1%8B).

Семантическая паутина — это надстройка над существующей Всемирной паутиной, придуманная для того, чтобы сделать размещаемую в Интернете информацию пригодной для машинной обработки. Доступная в сети информация удобна для прочтения человеком. Семантическая паутина создана для того, чтобы сделать информацию пригодной для автоматического анализа, синтеза выводов и преобразования, как самих данных, так и сделанных на их основе заключений в различные представления, полезные на практике.

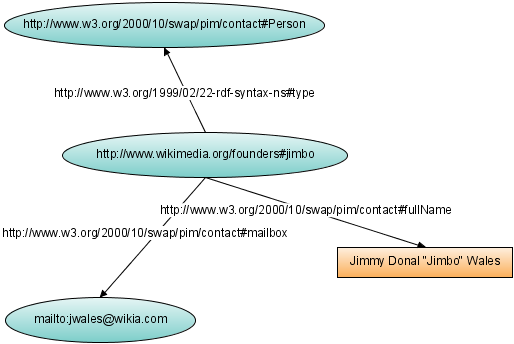


Рисунок 2. RDF граф

[Граф](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_%28%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0%29) визитной карточки основателя [Википедии](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%BA%D0%B8%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F) в формате [RDF](https://ru.wikipedia.org/wiki/RDF). Элементы этого графа — как узлы, так и дуги (кроме литерала, изображённого в оранжевом прямоугольнике) — являются [URI](https://ru.wikipedia.org/wiki/URI).

Машинная обработка возможна благодаря двум характеристикам семантической паутины:

* наличию [URI](https://ru.wikipedia.org/wiki/URI);
* использованию [семантических сетей](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C) и [онтологий](https://ru.wikipedia.org/wiki/%D0%9E%D0%BD%D1%82%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%8F_%28%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0%29).

[URI](https://ru.wikipedia.org/wiki/URI) — унифицированный идентификатор ресурса или адрес, используемый для указания [ссылок](https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%BF%D0%B5%D1%80%D1%81%D1%81%D1%8B%D0%BB%D0%BA%D0%B0) на какой-либо объект (например, [веб-документ](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0), [файл](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB) или ящик [электронной почты](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BE%D1%87%D1%82%D0%B0)). URI используются для именования объектов. Каждый объект глобальной семантической сети имеет уникальный URI. URI однозначно называет некоторый объект. Отдельные URI создают не только для страниц, но и для объектов реального мира (людей, городов, художественных произведений и так далее), и даже для абстрактных понятий (например, «имя», «должность», «цвет»). Благодаря уникальности URI одни и те же предметы можно называть одинаково в разных местах семантической паутины. Используя URI, можно собирать информацию о одном предмете из разных мест. Рекомендуется включать в адрес URI название одного из протоколов Всемирной паутины ([HTTP](https://ru.wikipedia.org/wiki/HTTP) или [HTTPS](https://ru.wikipedia.org/wiki/HTTPS)). То есть адрес URI рекомендуют начинать с «http://» или «https://»). Такой адрес можно одновременно использовать как адрес [URI](https://ru.wikipedia.org/wiki/URI) и как адрес веб-страницы ([URL](https://ru.wikipedia.org/wiki/URL)). На веб-страницах, адреса URL которых совпадают с URI, W3C рекомендует размещать описание предмета. Описание желательно предоставлять в двух форматах:

* в формате, удобном для чтения человеком;
* в формате, удобном для чтения машиной.

### Использование [семантических сетей](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C) и [онтологий](https://ru.wikipedia.org/wiki/%D0%9E%D0%BD%D1%82%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%8F_%28%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0%29).

Данные во Всемирной паутине, как правило, представлены в виде текста, записанного на [естественных языках](https://ru.wikipedia.org/wiki/%D0%95%D1%81%D1%82%D0%B5%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA). Такие тексты предназначены для восприятия человеком, но машина может понять их смысл, используя один из методов [обработки естественного языка](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%B5%D1%81%D1%82%D0%B5%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B3%D0%BE_%D1%8F%D0%B7%D1%8B%D0%BA%D0%B0). Методы выполняют [частотный](https://ru.wikipedia.org/wiki/%D0%A7%D0%B0%D1%81%D1%82%D0%BE%D1%82%D0%BD%D1%8B%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7) анализ и/или [лексический](https://ru.wikipedia.org/wiki/%D0%9B%D0%B5%D0%BA%D1%81%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7) анализ текста.

В качестве формата, удобного для чтения машиной, [W3C](https://ru.wikipedia.org/wiki/W3C) предлагает использовать [язык](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) [RDF](https://ru.wikipedia.org/wiki/RDF). Язык RDF позволяет описывать структуру семантической сети в виде [графа](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_%28%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0%29). Каждому узлу и каждой дуге графа можно назначить отдельный URI. Утверждения, записанные на языке RDF, можно интерпретировать с помощью онтологий. Для создания и описания онтологий рекомендуют использовать [языки](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) [RDF Schema](https://en.wikipedia.org/wiki/RDF_Schema) и [OWL](https://ru.wikipedia.org/wiki/Web_Ontology_Language). Онтологии создаются для получения из данных [логических заключений](https://ru.wikipedia.org/wiki/%D0%92%D1%8B%D0%B2%D0%BE%D0%B4_%28%D1%80%D0%B0%D1%81%D1%81%D1%83%D0%B6%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5%29). В основе онтологий лежат математические формализмы, определяемые дискрипционной логикой.

### Архитектура семантической паутины

Техническую часть Семантической паутины составляет семейство стандартов на языки описания, включающее [XML](https://ru.wikipedia.org/wiki/XML), [XML Schema](https://ru.wikipedia.org/wiki/XML_Schema), [RDF](https://ru.wikipedia.org/wiki/RDF), [RDF Schema](https://ru.wikipedia.org/wiki/RDF_Schema), [OWL](https://ru.wikipedia.org/wiki/Web_Ontology_Language), а также некоторые другие. Располагая их в порядке повышения уровня абстракции, реализуемого тем или иным языком, получаем:

* [**XML**](https://ru.wikipedia.org/wiki/XML) предоставляет синтаксис для определения структуры документа, подлежащего машинной обработке. Синтаксис XML не несёт семантической нагрузки.
* [**XML Schema**](https://ru.wikipedia.org/wiki/XML_Schema) определяет ограничения на структуру XML-документа. Стандартный синтаксический анализатор языка XML в состоянии проверить произвольный XML-документ на соответствие его структуры так называемой схеме документа, описанной в XML Schema.
* [**RDF**](https://ru.wikipedia.org/wiki/RDF) представляет собой простой способ описания экземплярных данных в формате субъект-отношение-объект, в котором в качестве любого элемента этой тройки используются только идентификаторы ресурсов. Существует стандартизованное отображение этих троек на XML-документы предопределённой структуры (то есть консорциумом W3 определена схема XML-документов, содержащих RDF-описания), а также на другие форматы представления (например, в нотацию [N3](https://ru.wikipedia.org/wiki/%D0%9D%D0%BE%D1%82%D0%B0%D1%86%D0%B8%D1%8F_3)).
* [**RDF Schema**](https://ru.wikipedia.org/wiki/RDF_Schema) описывает набор атрибутов (здесь их точнее назвать отношениями), таких, как rdfs:Class, для определения новых типов RDF-данных. Языком поддерживается также отношение наследования типов rdfs:subClassOf.
* [**OWL**](https://ru.wikipedia.org/wiki/Web_Ontology_Language) расширяет возможности по описанию новых типов (в частности, добавлением [перечислений](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D0%BC%D1%8B%D0%B9_%D1%82%D0%B8%D0%BF)), а также позволяет описывать новые типы данных RDF Schema в терминах уже существующих (например, определять тип, являющийся [пересечением](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D1%81%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2) или [объединением](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%B4%D0%B8%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2) двух существующих).
* [**Микроданные (HTML microdata)**](https://ru.wikipedia.org/wiki/%D0%9C%D0%B8%D0%BA%D1%80%D0%BE%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5)— это международный стандарт семантической разметки HTML-страниц, с помощью атрибутов, описывающих смысл информации, содержащейся в тех или иных HTML-элементах. Такие атрибуты делают контент страниц машиночитаемым, то есть позволяют в автоматическом режиме находить и извлекать нужные данные.

## Семантические модели и их применения

Основной движущей идеей при создании семантических технологий явилась необходимость обеспечения «понимания» алгоритмами вычислительных машин смысла (семантики) данных. Таким образом, исходной задачей этих технологий была аналитика: обеспечение возможностей извлечения знаний из связанных наборов информации.  
По мере развития этих технологий, экспериментов по их применению в различных сферах, выяснилось, что они исключительно удобны для объединения (связывания) данных из различных по структуре источников. Отсюда возникло второе направление развития инструментов, основанных на идеях семантических сетей – интеграция информационных систем.  
Никакого противоречия между аналитическим и интеграционным применением семантики нет; напротив, они находятся в неразрывном единстве. Ведь целью интеграции, как правило, является извлечение каких-то новых знаний из объединенного набора – таких знаний, которые не могли быть получены из каждого источника в отдельности. Задача упрощения переноса информации из одной информационной системы в другую тоже может быть решена при помощи семантических технологий, но является, скорее, дополнительным бонусом от их развития.

В ряде сфер применения удалось добиться существенных прорывов с использованием семантических технологий анализа информации. Особенно убедительными являются эти успехи в сфере медицины и биотехнологий. Например, на семантических технологиях строятся базы, объединяющие сведения о медицинских препаратах и их действии, клинические истории, генетическую информацию. Анализ таких баз помогает исследователям создавать новые лекарства. Это отличный пример ситуации, когда реляционные базы данных не в состоянии адекватно отразить многообразие связей между информационными объектами, и предоставить инструменты для анализа этих связей – а семантические технологии могут. Также семантические базы данных используются в здравоохранении (для анализа распространения заболеваний), и во множестве других применений.  
Средства анализа информации при помощи семантических технологий входят и в повседневную жизнь. Например, разработчики Facebook Graph Search придумали отличный пример, который позволяет на обыденном уровне продемонстрировать принципиальную новизну семантического поиска (анализа): очевидно, что ни одна из существующих поисковых машин, строящихся по принципу поиска в тексте, не сможет ответить на вопрос «Какие рестораны нравились моим друзьям?», или «В каких городах живут мои родственники?». Поиск по графу, используя формализованный набор информационных объектов (люди, рестораны, города) и отношений между ними (нравится, живет в), способен дать нужный ответ быстро и совершенно точно. При этом условия запроса можно варьировать в тех пределах, которые допускает онтология (набор тех самых типов информационных объектов и связей между ними): аналогичные вопросы можно задать не о ресторанах, а о фильмах, не о родственниках, а об одноклассниках. Понятно, что все содержимое социальной сети Facebook представляет собой огромный единый информационный граф, с миллиардами узлов и связей. Возможность анализа и использования этих связей и составляет всю его ценность, что прекрасно понимают владельцы ресурса.

Опираясь на сказанное выше, мы можем определить критерии, которые должны предъявляться к информационным моделям, строящимся по принципам семантических технологий. Часть этих критериев вытекает из общих требований к моделям, часть – из специфики технологий, связанных с условиями их практической полезности. Перечислим их.

1. Результат выполнения какого-либо действия в реальной системе и в модели должен совпадать (отношения подобия между моделью и системой в исходном и конечном состоянии описываются одними и теми же правилами). Это требование обеспечивает прогностический потенциал модели: если оно выполнено, мы можем моделировать развитие системы, и воплощать результаты моделирования.
2. Модель должна отражать свойства объектов и связи между ними таким образом, который делает возможным извлечение знаний из модели при помощи существующих технологий (таких, как SPARQL). Это сугубо практическое требование, обеспечивающее пригодность модели для анализа. Фактически, оно декларирует возможность выполнения расчетов на модели.
3. Модель должна обеспечивать возможности расширения и масштабирования (укрупнения и детализации), без пересмотра ее онтологического ядра. Это требование налагает ограничения на отбор способов классификации объектов, разграничение объектов и их свойств; это требование можно серьезно детализировать.

## Использование технологий Semantic Web для интеграции информационных систем

Технологии семантической паутины сейчас привлекают к себе внимание, потому что на их основе уже создаются новые интересные инструменты. В качестве примера можно привести социальный поиск Graph Search от Facebook.

Однако сфера применения семантических технологий не ограничивается социальными сетями и поисковыми сервисами. Идея применить эти технологии для организации обмена данными между информационными системами достаточно очевидна. Если одна система передает другой не только сами данные, но и информацию об их сущности в предметной области (смысле, семантике), это позволяет лучше абстрагировать обменивающиеся системы друг от друга, чем при использовании выгрузок в XML, JSON или веб-сервисов SOA.



Рисунок 3. Преобразование данных в RDF триплеты для обмена данными между информационными системами

Сегодня существует несколько реализаций такого подхода. Большинство из них, конечно, сделано зарубежными компаниями, но есть и российские разработки. В данном подразделе я расскажу об архитектуре одной таких систем, которую российские разработчики реализовали на практике.

При создании системы было задано, что нескольким информационным системам необходимо очень быстро (с интервалом всего в несколько секунд) сообщать друг другу об изменениях, происходящих в их хранилищах данных. Поэтому архитектура сильно напоминает шину обмена сообщениями (Message Queue), основной особенностью которой является то, что содержание сообщений выражено в синтаксисе RDF, то есть представляет собой RDF-триплеты, сохраненные в документе типа RDF/XML. Со стороны каждой из интегрируемых систем работает клиентский модуль обмена, который интерпретирует получаемые сообщения, и, если необходимо, вносит соответствующие изменения в хранилище данных своей системы. Клиентский модуль является автономным, то есть, если в хранилище произошли изменения, о которых нужно сообщить другим информационным системам – он кодирует сведения об этих изменениях в RDF, и отправляет в виде сообщения по корпоративной шине.

Роль маршрутизатора в шине выполняет центральный сервер-посредник, который обладает информацией о правах доступа информационных систем к разным видам данных, гарантирует доставку, контролирует целостность данных (и, при необходимости, старается ее восстановить), а также выполняет ряд других полезных функций. На рисунке ниже показана схема программных компонентов, участвующих в обмене. В качестве примера двух обменивающихся систем взяты, с одной стороны, некое веб-приложение на PHP/MySQL, и конфигурация 1С — с другой.

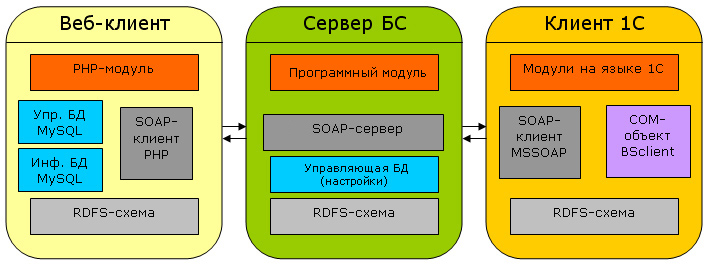


Рисунок 4. Возможная архитектура семантической корпоративной системой интеграции

Разумеется, на практике обменивающихся систем может больше, чем две.  
Преимущества этого подхода по сравнению с выгрузкой в XML или веб-сервисом SOAP достаточно очевидны, особенно если исходить из того, что необходимо связать между собой три, четыре или более систем. Они могут оперировать одними и теми же типами объектов (например, во всех информационных системах любой компании почти наверняка присутствует понятие «клиент»), но обладать разными наборами данных о них, и использовать их в разных контекстах. Также необходимо уточнить, что здесь хорошо проявляется один из принципов семантической паутины, а именно – глобальные типы. Существует множество информационных систем различного уровня, которые оперируют одинаковыми, по сути, сущностями (клиент, заказ, прибыль, работник), но в каждой системе эти сущности реализованы по-разному. Из-за эти ИС для обмена информацией друг с другом придется каждый раз преобразовывать данные из «неродного» формата. Однако если хранить информацию изначально в семантическом графе RDF, то сущность «Клиент» и в ERP системе маленького магазинчика и в CRM системе транснациональной корпорации будет представлена в одинаковом виде и обладать одним и тем же набором свойств. При традиционном подходе выгрузки в XML почти всегда будут избыточными, жестко связанными со структурой данных в базе данных системе-источнике, и потребуют написания программного кода для экспорта и импорта. Если, например, в CRM-системе хранятся таблицы с клиентами и сделками с ними, XML-выгрузка будет выглядеть примерно так:

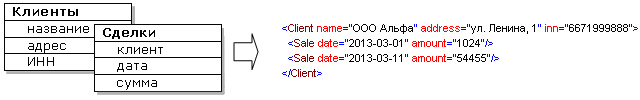


Рисунок 5. Представление информации в виде XML

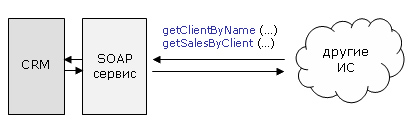
Понятно, что одной системе-потребителю этой информации сведения о сделках потребуются с детализацией по товарам, другой — без нее, зато со сведениями о грузополучателе и сроке доставки, и т.д. Использование выгрузок становится крайне проблематичным, если какие-нибудь данные могут измениться «задним числом». А уж если какая-нибудь из систем-получателей имеет право вносить изменения в эти данные (например, из системы отдела логистики могут поступать сведения о фактической отгрузке товара) — продолжать использовать выгрузки можно только из чистого упорства.  
Более прогрессивный вариант — использовать SOAP веб-сервис. Тогда схема взаимодействия системы-источника с потребителями информации будет выглядеть так:  


Рисунок 6. Использование веб-сервисов для интеграции информационных систем

Это будет удобно до тех пор, пока число сервисов (растущее пропорционально числу типов информационных объектов, которыми нужно обмениваться) не перевалит через несколько десятков. Тогда проблема мониторинга работоспособности сервисов, их документирования и поддержки начнет становиться по-настоящему критичной. Кроме того, это не решает проблему обратной связи систем: в упомянутом выше примере, когда система отдела логистики не только забирает информацию из CRM, но и может сама помещать в нее какие-либо данные, веб-сервисы придется реализовывать со стороны обеих систем, что совсем неудобно. Также программистам подобных систем придется самим заботиться о безопасности сервисов, поддержании целостности данных, обработке ошибок и т.п.

В более крупных инфраструктурах применяют MDM-системы (что требует совсем другого порядка вложений и усилий), а также шины обмена сообщениями. Шины хороши тем, что позволяют объединять большое количество информационных систем без возрастания сложности обмена в зависимости от числа систем. Фактически здесь реализуется шина, только в отличие от «классической» шины по ней будут передаваться сообщения, выглядящие примерно так:  
Информация, кодированная в виде триплетов RDF

Как уже говорилось, каждая система-отправитель перед отправкой преобразует в такой формат сведения, которыми хочет поделиться, а каждая система-получатель интерпретирует их в соответствии со своими правилами.  
Преимущества такого подхода в сравнении с классической шиной обмена сообщениями состоят в том, что информационным системам нет необходимости создавать «протокол обмена», определяющий смысловую нагрузку сообщений. Его роль выполняет онтология, известная всем обменивающимся системам и серверу (сервер обладает наиболее полной версией онтологии, клиентские системы могут использовать ее подмножества). В настройках каждого клиентского модуля производится сопоставление (mapping) элементов онтологии структурным элементам локального хранилища данных (проще говоря, таблицам и полям базы). Разумеется, можно создавать пользовательские программные обработчики для сущностей и связей, которые не имеют однозначного отображения на структуру базы данных, но в практических внедрениях объем требуемого программного кода измеряется десятками строк. Также, в отличие от классических шин обмена сообщениями, в нашем случае сервер выполняет ряд высокоуровневых полезных функций, вроде уже упомянутого восстановления целостности данных, выявления дублей объектов, разрешения конфликтов. Кстати, заодно сервер может формировать SPARQL базу данных, в которой будут аккумулированы сведения из всех обменивающихся систем, представленные в виде графа. Это обещает обеспечить такие возможности аналитики, какие не сможет предоставить ни одна из интегрируемых систем в отдельности.

Конечно же, использование описанного здесь подхода позволяет решить все перечисленные выше проблемы интеграции: обеспечить независимость процедур обмена со стороны различных систем от структуры баз друг друга, отказоустойчивость, безопасность, гибкость настройки, возможность легко использовать одну и ту же информацию в разных контекстах. Разумеется, описанный подход является далеко не единственным вариантом применения семантических технологий для интеграции данных.

## Сравнительный анализ семантических моделей для корпоративных приложений

Идея применения семантических моделей в корпоративных информационных системах существует давно, но устойчивая практика такого их использования еще не сформировалась. Семантические модели можно применять для интеграции данных, аналитики, управления знаниями; однако, пока нет общепринятого мнения о том, как подходить к оценке их полезности, по каким методикам должны строиться такие модели.

В этом разделе мы на практическом примере сравним аналитический потенциал моделей, построенных по правилам интеграционного стандарта ISO 15926, который предписывает использование OWL и SPARQL для выражения моделей и работы с ними, и «обычных» семантических моделей, построенных без использования этого стандарта. Решение этого вопроса позволит выбрать диапазон задач, для решения которого целесообразно применять такие высокоуровневые парадигмы семантического моделирования, как ISO 15926.

### Постановка проблемы

Необходимо кратко осветить историю вопроса, и суть взаимоотношений между ISO 15926 и «обычной» семантикой. ISO 15926 – стандарт обмена информацией, предназначенный для использования в промышленности (прежде всего, нефтегазовой). Исторически, акцент при разработке стандарта делался на обмен данными между различными организациями, т.е. между различными информационными инфраструктурами. Основные его особенности – специфический подход к классификации объектов и их отношений, учет временнóй составляющей объектов (4D моделирование), возможность моделирования жизненного цикла систем (а не просто текущего состояния той или иной системы). Стандарт содержит онтологическое ядро, и подразумевает использование общих библиотек справочных данных для создания прикладных информационных моделей. Все это обеспечивает как его преимущества (возможность создания высококачественных и релевантных моделей жизненного цикла систем, отличный потенциал использования для передачи информации между различными организациями при помощи общего «онтологического словаря»), так и недостатки (возрастающую сложность получающихся моделей, высокий «порог входа» по уровню знаний, необходимых для овладения стандартом и его использования).  
Разработка стандарта была начата еще в 1990-е годы. С появлением в середине 2000-х технологий Semantic Web, они были утверждены в качестве технологической основы для выражения данных в соответствии с ISO 15926. Таким образом, основные концепции стандарта были заложены до возникновения Semantic Web, но только появление этих технологий предоставило необходимый технологический базис для создания способа выражения данных в соответствии со стандартом, обладающего потенциалом действительно широкого распространения. Некоторая идеологическая близость, но не идентичность этих технологий заложила основу того «противоречия», которое хотелось бы разрешить. Поскольку принципы, по которым выполняется моделирование в соответствии с ISO 15926, не вполне соответствуют принципам представления объектов и отношений между ними, например, в языке OWL, объединение этих двух технологий получилось несколько синтетическим. Данные, построенные в соответствии с ISO 15926, могут быть разложены на элементарные элементы — триплеты RDF, но анализ отношений между информационными объектами, представленной в таком виде модели средствами SPARQL, будет затруднен.

### Пример: создание и анализ «обычной» семантической модели

Рассмотрим теперь два «конфликтующих» способа построения моделей, и оценим их практическую полезность, исходя из перечисленных критериев. Пример мы возьмем из области промышленности, «родной» для обсуждаемого стандарта.  
  
Опишем в виде семантической модели информацию о событии – установке насоса в трубопровод. Наша модель должна содержать следующие сведения:

* Место установки насоса (обозначенное на схеме завода определенным идентификатором; далее в тексте будем называть его «функциональным местом», в соответствии с терминологией ISO 15926);
* Сам насос, как физический объект определенного типа с определенным серийным номером, присвоенным ему в определенный момент времени;
* Сведения о том, что данный насос подходит данному месту (может быть в него установлен);
* Дата и время установки.

Сначала смоделируем эту информационную структуру без опоры на ISO 15926 (разумеется, создать ее можно множеством различных способов, мы произвольно выберем один).

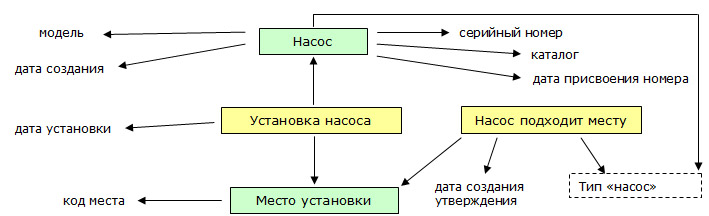


Рисунок 7. "Обычная" семантическая модель

Желтым показаны объекты, соответствующие событиям, зеленым – материальные объекты, без рамки – литералы. В пунктирную рамку обведено определение класса, которое, в принципе, относится к справочным данным, а не к конкретной модели. Стрелками показаны связи объектов друг с другом, и объектов с литералами (свойствами) – ребра графа.  
В результате импорта этой онтологии в SPARQL, получится набор из 16 триплетов (ребер графа). Они соответствуют показанным на схеме линиям, плюс по одному триплету для типа каждого объекта. Конечно, схема упрощена – например, «модель» должна быть не литералом, а ссылкой на соответствующий объект.

Рассмотрим возможности анализа этой модели. Например, пусть мы хотим узнать, в какое именно место был установлен насос с известным нам серийным номером. Для этого нам потребуется следующая последовательность простых запросов:

SELECT \* WHERE { ?pump <http://example.org/DC#Model> "Centrifugal Pump Model AB-123C"^^<http://www.w3.org/2001/XMLSchema#string> }

Этот запрос возвращает идентификатор объекта, содержащего информацию о насосе. Теперь найдем объект «установка насоса»:

SELECT \* WHERE { ?installation <http://example.org/DC#InstalledItem> <http://example.org/DC#DE-1234F> }

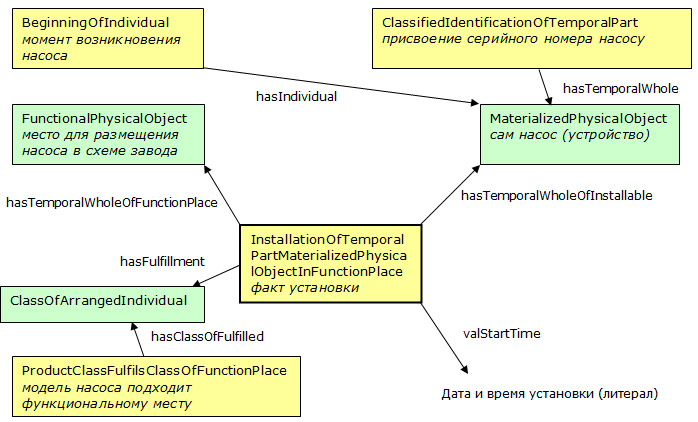
Осталось узнать место установки:

SELECT \* WHERE { <http://example.org/OurInstallation> <http://example.org/DC#InstalledPlace> ?place }

Мы просмотрели три ребра графа; конечно, эти запросы можно объединить в один.

### Пример: создание и анализ модели по стандарту ISO 15926

В соответствии со стандартом ISO 15926, наше событие – установка насоса – должно быть описано шаблоном InstallationOfTemporalPartMaterializedPhysicalObjectInFunctionPlace. В упрощенном виде структуру ролей этого шаблона, позволяющую выразить информацию, примерно эквивалентную той, что показана в примере выше, можно представить таким образом:



На этой диаграмме экземпляры шаблонов окрашены желтым цветом, экземпляры (instances) объектов – зеленым.  
Такая структура, заполненная минимально необходимыми данными (без аннотаций), при импорте в SPARQL точку доступа превращается в 36 триплетов.

Отметим, что структура этих данных в триплетах получается вполне разумной, и не так уж сильно отличается от структуры модели без использования стандарта. Увеличение числа триплетов по сравнению с «обычной» семантической моделью в два с лишним раза связано с добавлением новых информационных объектов, а также ссылок на определения базовых типов, содержащихся во многих из них. Однако преобразование в триплеты справочных данных, особенно – определений шаблонов, даст гораздо худшие результаты с точки зрения оптимальности структуры графа. Так, определение только одного шаблона InstallationOfTemporalPartMaterializedPhysicalObjectInFunctionPlace составляет 148 триплетов, многие из которых включают blank nodes (узлы графа, не имеющие собственных идентификаторов). В том числе, многие триплеты связывают между собой два blank node. Работа с такими структурами средствами SPARQL сильно затруднена. На практике это выльется в серьезное возрастание сложности программного обеспечения, реализующего возможности создания или просмотра шаблонов. Для сравнения, «обычная» семантическая модель тех же самых данных укладывается всего в 38 триплетов, то есть она на порядок компактнее, чем модель ISO 15926 (не забудем, что упомянутые выше 148 триплетов описывают только один шаблон, а их в нашем примере четыре, плюс определения необходимых стандартных типов). Еще одно важное отличие состоит в том, что модель ISO содержит связи со внешними элементами, находящимися за пределами той точки доступа, в которой размещена текущая онтология – в частности, в RDL.

Рассмотрим возможности анализа модели, построенной по правилам ISO 15926. Выполним те же задачи, которые описаны выше для «обычной» семантической модели. Пусть мы хотим узнать, в какое именно место был установлен насос с известным нам серийным номером. В модели ISO нам потребуется следующая последовательность простых запросов:

SELECT \* WHERE { ?temporalpart <http://standards.iso.org/iso/15926/tpl#valIdentifier> "S/N DE-1234F"^^<http://www.w3.org/2001/XMLSchema#string> }

Получаем в результате идентификатор экземпляра шаблона ClassifiedIdentificationOfTemporalPart. Теперь спрашиваем, с каким физическим объектом «насос» связан этот шаблон:

SELECT \* WHERE { <http://example.com/tpl#CITP456> <http://standards.iso.org/iso/15926/tpl#hasTemporalWhole> ?pump }

Получаем идентификатор насоса (объект типа MaterializedPhysicalObject). Теперь можем получить список экземпляров шаблона, описывающего установку насоса:

SELECT \* WHERE { ?installation <http://standards.iso.org/iso/15926/tpl#hasTemporalWholeOfInstallable> <http://example.com/tpl#MPO456> }

Получили идентификатор экземпляра шаблона InstallationOfTemporalPartMaterializedPhysicalObjectInFunctionPlace. Узнаем теперь, на какое функциональное место производилась установка:

SELECT \* WHERE { <http://example.com/tpl#T123> <http://standards.iso.org/iso/15926/tpl#hasTemporalWholeOfFunctionPlace> ?place }

Итак, нам потребовалось пройти четыре ребра графа. Важно отметить, что для того, чтобы составить эти запросы, разработчик должен быть детально знаком с принципами ISO 15926, и обладать аннотированной библиотекой шаблонов (которой, на самом деле, нет в открытом доступе).

### Сходства и различия «обычной» семантики и ISO 15926

Выводы из рассмотренных примеров использования семантических моделей вполне очевидны:  
1. С технологической точки зрения, «обычные» семантические модели вполне симметричны моделям ISO 15926, если говорить о проектных данных (выражающих информацию о конкретных системах и процессах). Модели ISO имеют бóльшую сложность, и этот разрыв, в сравнении с «обычными» моделями, растет в зависимости от объема модели по линейному закону. Это объясняется наличием отдельных сущностей для выражения темпоральных частей объектов, а также необходимостью классифицировать объекты в соответствии с классификатором типов верхнего уровня.  
2. С точки зрения вычислительного потенциала этих моделей – вычисления на них также несколько сложнее, чем в «обычной» семантике, но различие не является радикальным. Более существенно то, что для построения запросов необходимо не только знакомство с моделью, но и владение концепциями ISO 15926, а также наличие навигатора по шаблонам (который, насколько нам известно, отсутствует в открытом доступе; набор шаблонов и порядок их утверждения, насколько нам известно, тоже далеки от желаемых).  
3. Система справочных данных и высокоуровневых сущностей ISO 15926 сильно усложнена по сравнению с «обычной» семантикой (если брать за показатель количество триплетов, требуемых для выражения модели – в 10 раз и более). В особенности это касается библиотек высокоуровневых сущностей, таких, как шаблоны. Работа с определениями (не экземплярами!) этих сущностей средствами семантических технологий значительно затруднена. Тем не менее, любое приложение, предоставляющее пользователю возможности работы с шаблонами, и «прячущее» от него их низкоуровневое представление, должно обладать широкими возможностями такой работы (поиск и просмотр, создание и редактирование определений шаблонов, их заполнение). Частичным решением проблемы может быть работа с шаблонами, выраженными не в виде триплетов в RDF-хранилище, а в виде файлов OWL.  
4. Концепции ISO 15926, считающиеся его «ноу-хау», и обеспечивающие особую ценность этого стандарта – использование федеративного доступа и библиотек RDL, учет темпоральных частей – доступны и в «обычной» семантике. Все зависит от того, каким образом построена модель данных, и как реализовано разделение данных на проектные и справочные. Кстати, заметим, что нет никаких практических препятствий для использования библиотек RDL, построенных в соответствии с ISO 15926, в приложениях, использующих не соответствующие ему модели данных.   
5. Действительную ценность стандарта составляет, прежде всего, его статус стандарта; общепринятые способы классификации и сами классификаторы (а также способы их администрирования) обеспечивают потенциал использования стандарта для интеграции между различными предприятиями, но несколько усложняют выполнение вычислительных задач на моделях. Это естественная ситуация: за любую универсальность приходится платить быстродействием.  
  
Таким образом, стандарт ISO 15926 представляет собой один из способов построения семантических моделей, обладающий определенными достоинствами и недостатками по сравнению с другими способами, содержащими меньше высокоуровневого формализма. С точки зрения практической реализации и потенциала использования, между стандартом и другими способами нет принципиальной разницы, которая позволила бы противопоставить их как различные технологии. Декларирование наличия такой разницы могло бы считаться маркетинговым приемом пропаганды стандарта, если бы оно не играло вместе с тем отпугивающей роли для специалистов, уже знакомых с «обычными» семантическими технологиями (как это происходит сейчас на практике). К тому же, объяснить разницу между ISO 15926 и «обычной» семантикой на технологическом уровне людям, не являющимся ИТ-специалистами, но принимающим решения о создании той или иной программной инфраструктуры, крайне сложно.  
  
Создание качественных моделей систем возможно как с использованием данного стандарта, так и без него; решение о его использовании должно приниматься исходя из контекста применения разрабатываемой информационной системы, прежде всего – с точки зрения возможного включения модели в интеграционные процессы, и с точки зрения требований к проведению вычислений на модели. Следование стандарту можно высказать в качестве общей рекомендации, однако при определенных обстоятельствах необходимость быстрых вычислений на модели может потребовать реализации более рациональной онтологии. Решать проблему оптимизации вычислений только аппаратными средствами, как правило, нерационально.  
Основными препятствиями к распространению стандарта следует считать:

* высокий «порог входа» – объем знаний, необходимый для успешного использования этой технологии;
* отсутствие полноценной методической базы, документации, развитых сообществ поддержки, сборников примеров использования;
* отсутствие доступного широкому кругу пользователей программного обеспечения, предоставляющего возможности работы с моделями данных, построенными в соответствии со стандартом;
* отсутствие убедительных и открытых примеров успешного использования стандарта (выходящих за рамки простой декларации о том, что его применяют такие-то компании для таких-то целей).

Именно с этими проблемами и нужно бороться, распространяя стандарт как передовую практику. Его искусственное противопоставление «обычным» семантическим технологиям может играть в этом процессе только негативную роль.

Таким образом, несмотря на свои положительные особенности, этот стандарт проигрывает в гибкости обычным «не стандартизированным» семантическим моделям и его сложно назвать универсальным. С другой стороны создание стандарта ISO 15926 было, по сути, первой серьезной попыткой сделать универсальный способ обмена промышленной информацией. Другими словами он пытался решить те же проблемы в информационных технологиях корпоративного уровня, что сейчас адресуются семантическим сетям.

## Семантический рабочий стол

Семантический рабочий стол — это устройство, на котором каждый хранит всю свою цифровую информацию — документы, мультимедиа и сообщения. Они интерпретируются как ресурсы семантической паутины, каждый из которых имеет уникальный идентификатор ([URI](https://ru.wikipedia.org/wiki/URI)), все данные доступны, и их можно запросить как [RDF](https://ru.wikipedia.org/wiki/Resource_Description_Framework)-граф. Ресурсы из сети могут сохраняться, а авторским контентом можно делиться с другими. Онтологии позволяют пользователю выражать персональные интеллектуальные модели и формировать семантическую связку, соединяющую информацию и системы. Приложения действуют указанным образом, и хранение, считывание, взаимодействие происходят через онтологии и протоколы семантической паутины. Семантический рабочий стол — существенное дополнение к пользовательской памяти.

Представление семантического рабочего стола может рассматриваться как ответ на осознаваемые проблемы существующих пользовательских интерфейсов. Не обладая качественными [метаданными](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%B0%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5), компьютеры не способны без труда узнавать множество зачастую необходимых атрибутов файлов. Например, предположим, что кто-то скачивает документ, созданный конкретным автором, по конкретной теме; хотя наверняка из документа будет ясна его тема, автор, источник и, возможно, информация об [авторском праве](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D1%80%D1%81%D0%BA%D0%BE%D0%B5_%D0%BF%D1%80%D0%B0%D0%B2%D0%BE), компьютеру может быть не так просто получить эту информацию и обрабатывать её в различных приложениях — таких, как файловые менеджеры, локальные поисковики файлов и т. д. Это означает, что компьютер не сможет осуществлять поиск, фильтрацию или иные действия над информацией с максимальной эффективностью. Это основная проблема, которой занимается [семантическая паутина](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%BF%D0%B0%D1%83%D1%82%D0%B8%D0%BD%D0%B0).

Также существует проблема зависимости различных файлов друг от друга. Например, в UNIX-подобных операционных системах [электронные письма](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BE%D1%87%D1%82%D0%B0) хранятся отдельно от файлов. Ни одна из проблем не имеет отношения к задачам, заметкам или запланированным действиям, которые могут храниться в программе-календаре. Контактные данные могут храниться в другой программе. Так или иначе, все эти формы представления информации могут одновременно быть актуальными и необходимыми для какой-либо конкретной задачи.

В связи с этим, пользователь часто будет получать — с помощью [браузера](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) или иной программы — доступ к большому количеству информации из интернета, которая отделена от информации, локально хранимой на компьютере.

Семантический рабочий стол — это попытка решить часть или все эти проблемы разом путём расширения возможностей операционной системы, делая возможным управление всеми данными с помощью технологий семантической паутины. Улучшенные пользовательские интерфейсы (или плагины к существующим приложениям), основанные на такой интеграции данных, могут давать пользователю целостностное представление о хранимых знаниях. Некоторые операционные системы, такие как [BeOS](https://ru.wikipedia.org/wiki/BeOS), имеют файловые системы, схожие по функциональности с реляционными базами данных, которые хранят метаданные о документе прямо в файловой системе, что является существенным шагом на пути к семантическому рабочему столу.

Существуют различные интерпретации идеи семантического рабочего стола. В наиболее узком смысле она может интерпретироваться как добавление механизмов привязывания считываемых компьютером [метаданных](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%B0%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5) к файлам. В крайних случаях она может трактоваться как полная замена существующим пользовательским интерфейсам, унифицирующая все формы данных и обеспечивающая единый последовательный интерфейс. В зависимости от решаемой проблемы, есть много степеней градации между двумя перечисленными вариантами.

Рисунок 8. Структура Semantic Desktop

# Часть 2. Ключевые технологии

## Основная идея

Эта часть является описанием основных идей, на которых строится система ПИМ. Ключевая мысль заключается в том, чтобы перенести существующие технологии СВ на персональный компьютер. Ресурсы на компьютеры получают UNI адреса а их метадата конвертируется в RDF. Существующие десктопные приложения включены в эту структуру, а не заменены. Далее в деталях рассмотрено значене UNI в контексте его использования для идентифицирования идей и концепций.

## Semantic Desktop

СВ позволяет улучшить WWW путем включения дополнительных метаданных. Этот же подход может быть использован и на персональных компьютерах. Подобно Вебу сервер приложений публикует доступную информацию. Локальный Semantic Web server может быть использован для того, чтобы получить доступ к информации расположенной на компьютере. Файлы и другие ресурсы могут быть использованы как ресурсы веб. Они идентифицируются с помощью UNI и позволяют создавать связи между различными документами. Существующая информация представлена в виде RDF документов и может быть использована любым другим приложением. Таким образом проявляется одно из основных свойств семантической паутины – различные программы одинаково обрабатывают данные с семантическими метаданными. Единый стандарт представления информации позволяет упростить взаимодействие между приложениями. В данном случает под приложениями понимаются десктопные программы, однако подобная структура может быть масштабирована до корпоративного уровня или даже глобального.

При использовании технологий семантической паутины на персональном компьютере в системной архитектуре создается семантический уровень. С этого момента ресурсы доступ к ресурсам может быть предоставлен посредством Semantic Web Server параллельно доступа к ним через уже установленные программы. Ресурсы идентифицируются независимо от формата своих файлов. На основе этого семантического уровня появляется возможность построить систему классификации, охватывающую все существующие ресурсы на компьютере. Такие концепции как «личное» или «работа» будучи один раз представлены в системе могут быть использованы всеми приложениями. Все ресурсы могут быть идентифицированы через UNI и просмотрены в браузеро-подобных программах. Поисковик, вроде Google, может быть создан поверх системы для того чтобы создать полнотекстовый индекс всех существующих текстовых данных. В отличие однонаправленных гипертекстовых ссылок в вебе связь между локальными ресурсами является двусторонней.

Отношение между двумя ресурсами может быть установлено в любой точке системы и быть доступно из любых других мест.

Как будет показано в примере привнесение технологий СВ в архитектуру персонального компьютера может быть с пользой использовано различными способами. При получения доступа к ресурсам через Semantic Web server открывает новые возможности по использованию существующих данных.

## Принципы

### Один компьютер – один пользователь.

Представленная система семантической паутины спроектирована с учетом того, что ею будут пользоваться один пользователь, чьи ресурсы входят в эту семантическую сеть. Другие существующие PIM тоже поддерживают только одного пользователя, но не все. В данном случае ограничение на одного пользователя упрощают систему, так как отпадает необходимость поддерживать нескольких пользователей. Программа работает на одном персональном компьютере. Это не распределенная многопользовательская система и поэтому в ней отсутствуют механизмы пользовательской аутентификации и авторизации. В системе нет встроенной возможности обмена данными, если необходимо поделиться информацией с другими пользователями, она должна быть экспортирована и импортирована вручную. Вся RDF информация хранится на одном компьютере. Таким образом, весь RDF граф принадлежат одному пользователю. Так как каждый пользователь – уникальная личность и может быть идентифицирован, программа SemanticDesktop идентифицируется с помощью *hostname* (имя хоста). Так как имя хоста (доменное имя) глобально уникально, для каждого ресурса на компьютере возможно задать уникальный идентификатор. Если у компьютера нет уникального, доменного имени можно использовать другой способ. Этот способ заключается в том, чтобы использовать альтернативный идентификатор, который является доменным именем, принадлежащим владельцу компьютера плюс какое-нибудь имя для компьютера. Например, у моего компьютера нет уникального доменного имени, но я владею доменом

**http://www.arthur.lazy-magister.org**  
В этом случае уникальное имя для моего компьютера будет

**http://www.arthur.lazy-magister.org/learnmachine**

В данном случае уникальность заключается в том, что пока я являюсь владельцем доменного имени, никто другой не может им воспользоваться. Следует отметить, что, хотя идентификатор и похож на URL, он всего лишь используется как уникальное имя и не указывает на какой-либо ресурс в сети.

### У всего есть URI.

Для работоспособности системы необходимо наличие у ресурсов отличительных идентификаторов. В настоящее время на подавляющем большинстве персональных компьютеров нет встроенной системы идентификации ресурсов. Некоторые операционные системы на базе ядра Linux частично реализуют механизм идентификации за счет использования семантической файловой системы, однако такие операционные системы очень редки. При использовании PIM у нас должна быть возможность ссылаться на объекты независимо от их принадлежности различным установленным приложениям. Другими словами у нас должна быть возможность получить данные и из контакта в email клиенте, и из музыкального файла, и из документа, расположенного в файловой системе. На данный момент всех эти ресурсы идентифицируются по-разному. Файл определяем его полное имя (адрес), у email’а есть message-id в заголовке, книга идентифицируется по Международному стандартному книжному номеру (ISBN) или по Международному стандартному серийному номеру (ISSN). Таким образом, видно, что существует множество способов идентификации ресурсов и даже несколько для одного типа. Если бы мы решились создать книжный интернет-магазин, нам бы пришлось выбирать, по какому идентификатору сохранять книги в базе данных (ISBN или ISSN). Что бы мы ни выбрали, нам бы пришлось ограничить все данные о книгах одним идентификатором, и у нас не было бы возможности его поменять. Использование же UNI для идентификации ресурсов имеет несколько преимуществ по сравнению с другими ссылочными классификаторами наподобие ISBN или ISSN. UNI адреса глобально уникальны. Их использование не ограничено локальной базой данных и могут быть использованы на мировом уровне. Другими словами, если закрепить какую-либо UNI строку за определенным ресурсом, то во всем мире этот идентификатор бы ссылался именно на этот ресурс. UNI адрес содержит идентифицирующую информацию. Вместо того чтобы поддерживать различные системы идентификации, такие как имя в файловой системе или message-id, мы решили трансформировать все идентификаторы в UNI схему. И теперь для того чтобы сохранить ссылку на любой ресурс на нашем компьютере нам необходим только UNI. Информации по каждому ресурсы может быть запрошена по его уникальному UNI. Нет необходимости указывать тип ресурсы и где его можно найти, вся информация уже содержится в самом идентификаторе.

UNI адрес строится по следующей схеме:

<protocol>://<hostname><path>?<query>#<anchor>

<протокол>://<доменное имя><путь>?<параметры>#<якорь>

Ресурсы, находящиеся на компьютере пользователя получают UNI, который содержит в себе доменное имя, принадлежащее пользователю. Таким образом, использование доменных имен в UNI делает идентификатор ресурса глобально уникальным.

Некоторые ресурсы на персональном компьютере уже идентифицируются подобным образом. Если бы на нашем хосте работал FTP сервер, ftp URI мог бы быть использован для идентификации файлов. Но так как никакого FTP сервера у нас нет мы будем использовать альтернативный способ. URI путь начинающийся на file:// завязан на файловую систему. Путь [file://artur.lazy-magister.org/data](smb://artur.lazy-magister.org/data) будет указывать на нашу домашнюю директорию. Это похоже на открытый доступ по локальной сети к файлам: домашний каталог как удаленный корневой каталог а “data” – его публичное имя. Таким образом, файлы в домашнем каталоге в нашей системе могут быть проиндексированы с помощью URI.

Протоколы в URI схеме такие как http:// и file:// могут быть использованы для доступа и идентификации ресурсов на веб-сервере.

### Вся информация в виде RDF.

SemanticDesktop берет на себя ту роль, которую в корпоративных системах исполняет Интеграции Приложений Предприятия (*Enterprise Application Integration*, *EAI*) а именно – концентрирует все метаданные в одном месте – в семантическом графе, записанном в виде RDF. Это происходит путем трансформирования разрозненных данных в единый формат. Все вышеперечисленные типы ресурсов должны быть доступны в центральном графе. Приведение всех данных к одному формату позволяет нам использовать единый интерфейс доступа к этой информации без необходимости каждый раз извлекать данные из различных приложений, установленных на нашем компьютере. Доступ к данных происходит через локальный сервер. Также, так как мы используем технологии семантической паутины в качестве формата данных для семантического графа был выбран RDF (Resource Description Framework, Среда Описания Ресурсов). Нотация записи RDF не так важна. Существует две наиболее распространенных нотации: RDF/XML для представления графа семантической паутины в виде структурированного XML документа и N3 «Тройная нотация», которая представляет собой простой текстовый формат. Так как ядром нашей системы служит свободное программное обеспечение Apache Jena, поддерживающая все форматы, мы свободны в своем выборе. Для данной системы я выбрал формат записи RDF/XML как наиболее лаконичный.

Есть два подхода к трансформации данных. Первый показан на рисунке «Трансформация и буферизация. В данном случае задача разделятся на два процесса: извлечение данных и получение данных. Вся необходимая информация извлекается из своих приложения, трансформируется в RDF и буферизуется в базе данных. При изменении ресурсов необходимо обновиться, поэтому за изменением ресурсов всегда должно вестись наблюдение. Извлечение данных и их конвертирование в RDF – это простой процесс. Для хранение информации в виде RDF могут быть использованы существующие RDF базы данных. Эти базы данных поддерживают нативную RDF структуру, и подходят для нашей задачи. Оценив несколько существующих решений, я решил остановиться на базе данных, встроенной во фреймворк Apache Jena. Запросы информации к базе данных проходят быстро, и все содержащиеся в ней данные могут быть проиндексированы. К сожалению, при подобном подходе приходится постоянно следить за изменением внешних ресурсов, что значительно усложняет программу. Кроме того, одним из следствий этого решения является ситуация, когда вся обрабатываемая информация сохранена и в исходных файлах и в базе данных. И хотя на физическую память не налагаются большие ограничения, подобных подход будет потреблять в два раза больше дискового пространства.



Рисунок . Трансформация и буферизация

Во втором подходе трансформация исходных данных в RDF происходит по требованию, когда в систему поступает запрос о метаданных конкретного ресурса. Смотри рисунок «Извлечение на лету». Сначала запрос анализируется для того чтобы определить метаданные какого ресурса требуются. Затем происходит проход регистра ресурсов особым алгоритмом, чтобы определить какой именно исходный файл представляет этот ресурс. После этого файл ресурса читается, информация преобразуется в RDF и инициатору запроса. Такой подход замедляет обработку запросов в системе, так как поиск исходных файлов, извлечение информации и ее преобразование в RDF требует времени и происходит при каждом запросе. Преимущество этого метода заключается в том, что преобразование данных из исходных файлов происходит на лету и поэтому для каждого запроса гарантируется возвращение не устаревшей информации.

При создании системы SemanticDesktop был выбран первый подход, так как он обеспечивает концентрацию всей метаинформации о ресурсах персонального компьютера в одном месте в виде семантического графа и позволяет быстрее извлекать данные и проходит по узлам графа.

В следующей главе мы подробно рассмотрим приложения, установленные на персональном компьютере и способ извлечения из них информации.

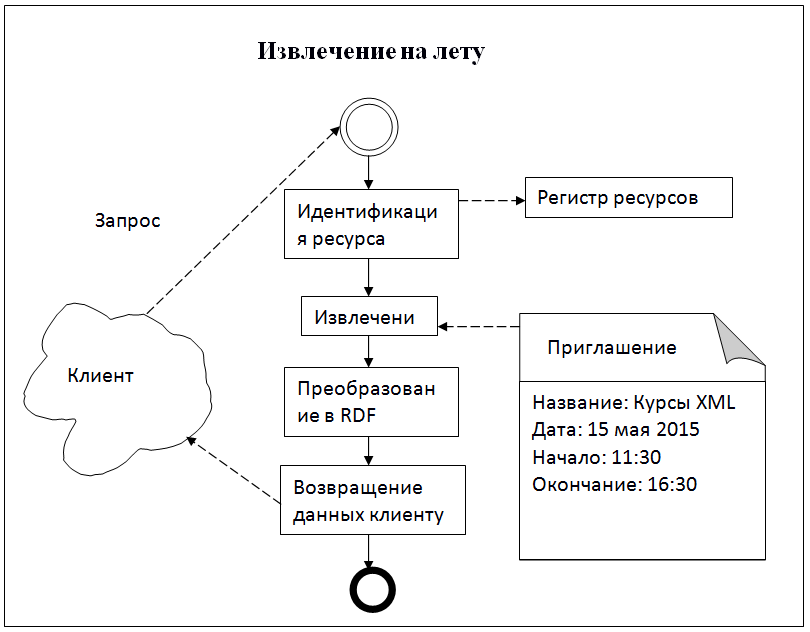


Рисунок . Извлечение на лету

### Целевые приложения.

На обычном персональном компьютере существуют предустановленные программы. Эти программы (приложения) могут манипулировать, к которым операционная система предоставила доступ. Например, приложения Microsoft Word может изменять файлы документами Microsoft Word. Подобные приложения являются лучшим способом создания и изменения информации. Они могут извлекать метаданные, создавать новые файлы, удалять их, печатать их, отображать их и т.п.

В данной работе я решил называть их целевые приложения. В дальнейшем я буду часто использовать этот термин. Эти приложения отвечают за свои файлы (ресурсы). Каждый ресурс должен иметь целевое приложение, иначе его невозможно модифицировать пользователю. Важно заметить, что один из ключевых принципов системы SemanticDesktop подразумевает незаменяемость этих приложением программой семантической паутины. Другими словами, SemanticDesktop работает параллельно с уже установленными программами. Таким образом, существующая информация остается в своем нативном формате и может быть дальше модифицироваться с помощью своего целевого приложения. Система семантической паутины лишь добавляет возможность идентификации любого ресурса и извлечении информации о нем.

SemanticDesktop зависит от целевых приложения установленных на персональном компьютере и от их API (Application Programming Interface, Интерфейс Программирования Приложений), который используется для того, чтобы извлекать информации из ресурсов (файлов) этих приложений. Если пользователь захочет изменить или просмотреть информации ресурса, будет вызвано целевое приложение.

В случае если целевое приложение пригодно только для открытия своих файлов, но не может извлекать метаданные, необходима помощь стороннего программного обеспечения. Например, аудиоплееры обычно могут только проигрывать MP3 файлы, но извлекать и них метаданные, поэтому для этой цели придется использовать стороннюю библиотеку под названием «MP3File».

Основная идея нашей разрабатываемой системы заключается в том, чтобы интегрировать существующие приложения в семантическую сеть, а не заменить их.

### Ассоциативные связи между ресурсами.

Человеческий мозг хорошо справляется с запоминанием ассоциаций. Концепция ассоциативной базы данных хорошо подходит для программ типа Personal Information Management. Программы для создания ассоциативныех карт (mind map) тоже работают по этому принципу. Ресурсы в нашей SemanticDesktop системе тоже могут быть ассоциативно связаны посредством RDF триплетов. Ресурсы (файлы данных) различных программ могут быть соединены друг с другом. Таким образом, мы преодолевая границы между различными приложениями на персональном компьютере. Ресурс может быть связан сразу с несколькими другими ресурсами и классифицирован более чем в одну категорию. Самый простой способ образования связей между ресурсами заключается в том, чтобы создать RDF триплет с предикатом типа relation из вокабуляра Dublin Core, который будет соединять один объект с другим. В данном случае большей проблемой будет создание пользовательского интерфейса, позволяющего ему создавать смысловые связи между ресурсами. Необходимо будет использовать системы онтологий вроде RDFS для того чтобы определить полезные и возможные способы линковки ресурсов различного типа.

### Онтологии.

В семантической паутине, онтологии использую для того чтобы описать метаданные об определенных ресурсах. Общие системы онтологий являют собой тезаурусы и в них могут быть указаны классификации и возможные свойства различных ресурсов. Двумя распространенными языками описания онтологий являются RDFS и OWL. Мы в системе SemanticDesktop сосредоточимся на использовании RDFS. Онтологии будут необходимы в четырех случаях.

Во-первых, сам сервер SemanticDesktop зависит от онтологий. Внутренние операционные данные, например счетчики, и конфигурационные файлы описаны в виде онтологий. Они записаны на языке RDFS и сохранены как RDF/XML файлы на сервере.

Во-вторых, онтологии описывают информацию, извлекаемую из ресурсов целевых приложений. Например, программы - почтовые клиенты, такие как Microsoft Outlook могут хранить информацию о контактах электронной почты в особых файлах. В данном случае онтологии описывают структуру извлекаемой информации из ресурса. Для различных приложений и ресурсов необходимы разные онтологии.

В третьем случае использования онтологий – это публичные онтологии. Они созданы группой экспертов (таких как консорциум всемирной паутины, W3C) и описывают распространенные типы ресурсов вроде проектов, людей и документов. Эти онтологии опубликованы в интернете и их могут использовать любые заинтересованные стороны. Важная функция публичных онтологий – это глобальная стандартизация элементов семантической паутины. На данный момент подобных онтологий не очень много. В нашем проекте SemanticDesktop мы будем использовать две публичные онтологии.

* “Friend of a Friend”, созданный Dan Brickley и Libby Miller. Эта онтология содержит классы для людей и общих понятий, таких как организации и документы. Основная идея этой онтологии заключается в описании личностей и отношений между ними.
* “Dublic Core”, созданный Dublin Core Metadata Initiative. Эта онтология используется для детального описания метаданных документов. Она содержит исчерпывающий словарь предикатов, который описывает такие свойства как автор, тему, название копирайт документа.

И наконец, третий способ применения онтологий – это пользовательские онтологии созданные пользователями системы. Любой пользователь может решить создать новую онтологию, которая подходит для его специфических нужд. Специфические классы, такие как «нужно будет скоро сделать» и «относится к работе» могут быть созданные в таких онтологиях. Наша система SemanticWeb может использовать пользовательские онтологии, но в ней нет встроенного редактора, поэтому задача создание онтологии полностью ложится на плечи пользователя.

Онтологии могут быть включены просто как файлы RDF/XML или быть сохранены в центральном репозитории (базе данных). SemanticWeb умеет испортировать онтологии посредством веб-интерфейса.

### Совместное использование

Создание ассоциативной сети ресурсов подобно переносу структуры мыслей пользователя в электронный вид. Полученный в результате семантический граф является очень субъективным и наилучшим образом подходит для пользования только одним человеком. Так как информацией подобного вида тяжело делиться с другими пользователями по сети и для того чтобы упростить разработку нашего прототипа программного обеспечения, функционал совместного использования не был включен в текущую версию продукта. Система является изолированной и в ней отсутствует возможность делиться метаданным с другими системами SemanticDesktop. Также система не поддерживает автоматическую публикацию метаданных и пересылку их посредством Email.

Перед тем как станет возможным обмен семантическими метаданными между пользователями системы, необходимо будет установить некоторые правила совместного пользования подобной информацией. Прежде всего, необходимо чтобы все пользователи сети согласились на использование общей онтологии. Для этого придется либо конвертировать метаданные конкретного пользователя в общий вид, либо заставить всех пользователей изначально пользоваться только заранее обговоренными онтологиями. Последний способ как раз представляет собой одну из основных идей семантической паутины, когда существует всеобъемлющая глобальная онтология и любой граф в этой паутине строится только на ее основе.

Если в будущем будет существовать подобная онтология, система SemanticDesktop может быть расширена посредством добавления модуля для совместного пользования метаданными. А так как для идентификации ресурсов мы уже использует глобальные идентификаторы, добавление подобного функционала не должно вызвать больших проблем.

## Снова о URL

Всем известно, что в интернете URL используется для идентификации ресурсов сети. Анализируя его составные части – протокол, домен и путь – браузер или другие приложения могут определить местонахождение ресурса в сети и то, как именно получить к нему доступ. Спецификация URL – это один из трех столпов, на которых построена Мировая Паутина: URL, HTTP, HTML. Интернет адреса преодолевают границы персональных компьютеров. Их можно увидеть в рекламе по телевизору, на визитных карточках и даже на товарах массового потребления – адреса давно вошли в повседневную жизнь.

Кажется, вполне ясно, что собой представляет URL и для чего он нужен, но это не совсем так. С самого начала своего существования природа URL была предметом ожесточенных споров. В этом разделе мы подробно опишем роль URL в нашем приложении SemanticDesktop.

## URL и URI – Расположение и Идентификация.

Во времена зарождения интернета существовала только одна концепция – URI (тогда еще расшифровывающийся как Universal Resource Identifier “Универсальный Идентификатор Ресурса”, но потом переименованный Uniform Resource Identifier “Единый Идентификатор Ресурса”), который служил как для идентификации документов в сети так и для указания их расположения. Затем, после обсуждения в IETF (“Internet Engineering Task Force”, Инженерный совет интернета), Был придуман термин URL (Uniform Resource Locator). Различие между ними заключается в том, что при изменении месторасположения документа, его URL меняется, а вот URI остается прежним. В системе SemanticDesktop URI является одновременно и идентификатором и адресом ресурса. URI обрабатывается и анализируется с двумя целями. Во-первых, для того чтобы определить физическое расположение ресурса, и во-вторых, для того чтобы извлечь метаданные этого ресурса. По идентификатору ресурса возможно определить его целевое приложении и с его помощью (или используя стороннее программное обеспечение, если целевое приложение на это не способно) извлекается метаинформация о данном ресурсе в системе.

Это разыменовывание URI для того чтобы определить где находятся ресурсы (домен) и как получить к ним доступ (протокол) являлось одной из причин такого успеха Всемирной Паутины. Существует некоторые до сих пор нерешенные проблемы, касающиеся URI (например, отсутствие поддержки символов национальных алфавитов), но, не смотря на это, URI является лучшим способом глобальной идентификации ресурсов. Однако же вопрос, что же собой представляет URI, остается. В данный момент эта проблема неопределенности получила название «Кризис URI».

## Кризис URI.

Когда задается цель дать возможность пользователям классифицировать различные ресурсы, прежде всего, необходимо определить саму классификацию. Во введении мы показали пример создания классификации. Файлы, например, могут быть поделены на две группы: «личное» и «работа», а для того чтобы выразить это разграничение они должны быть сохранены в двух разных директориях, названных «личное» и «работа». В Семантической Паутине тоже возможно осуществить подобную классификацию. В RDF онтологии необходимо представление классов. Мы можем определить классы в файле RDF схемы и обозначить объекты как экземпляры этих классов. Необходимым условием для этого будет возможность идентифицировать классы с помощью URI. Таким образом, нам нужны идентификаторы для классов «личное» и «работа» и мы будем их использовать в нашем программном обеспечении. Но перед этим надо ответить на вопрос, что собой представляет URI и что он может идентифицировать. Неоднозначность концепции идентификации получила название «Кризис URI». Единственный URI может иметь различные значения. У него может быть четыре аспекта. Например, используя определение концепции лени (lazy) можно показать четыре аспекта. Что идентифицирует следующий URI:

**“http://x.org/lazy”**?

Само название **http://x.org/lazy**, это строка, которая соответствует структуре URI.

* Определенную концепцию лени
* Адрес в сети, по которому располагается документ
* Сам документ, который описывает концепцию лени и к которому можно получить доступ и в дальнейшем обработать.

Когда URI используется в приложениях семантической паутины, он может использоваться во всех вышеперечисленных значениях. W3C Technical Architecture Group (подразделение консорциума Всемирной Паутины), озаботившись разграничением *концепции* от *веб адреса*, разработал два подхода к решению проблемы двойственности. Опишем их коротко.

## Различные URI

Первый подход заключается в том, чтобы использовать разные URI для разных контекстов, в которых находится ресурс. Попросту говоря, необходимо использовать новую систему для определения концепций, а URI идентифицирующие веб-адрес уже существуют. URI-концепции могут содержать в себе части, которые бы отличали их от адресов: например, можно использовать другой протокол (rdfp://, info:// и т.д.). Однако это подразумевает создание еще одного протокола передачи данных, что не очень практично. Также можно использовать идентификатор фрагмента (символ #) в конце строки URI для того, чтобы обозначить концепцию (…lazy#concept). Идентификатор фрагмента часто используются для указания на часть веб-документа и, таким образом, может быть использован для идентификации ресурса. И все же идентификаторы фрагмента – неустойчивое решение. Специальное хост-имя в пределах сервера, которое будет содержать в себе только термины концепций, может быть вставлено перед именем домена. Например, сравните “**http://x.org/lazy**” и “**http://concept.x.org/lazy”**. Но для этого понадобится устанавливать отдельный сервер. Недостаток всего этого подхода заключается в том, что URI для идентификации концепции может уже существовать в сети, и невозможно будет различать концепцию и адрес документа.

## Различные контексты.

Второй подход – это определение различных контекстов для URI. Попросту говоря, Один и тот же URI используется как «идентификатор концепции» или «идентификатор ресурса». Приложения должны будут интерпретировать URI по-разному в зависимости от того, в каком контексте его рассматривает, ресурс или концепция. Проблема заключается в том, что текущая спецификация RDF не допускает подобного. Также неочевидно, какие правила использования различных концепций.

Решение этой проблемы не было принято сообществом по причине сильной философской направленности. Что «есть» веб-сайт и что «есть» концепция и как их различить? Многие компании сильно привязаны к своим веб воплощениям. Так в чем же отличие сайта “www.amazon.com” и компанией, расположенной в Сиэтле, особенно с точки зрения пользователя. Приложение «Я купил книгу на Амазоне» и «Я купил книгу на www.amazon.com» не особенно отличаются. Что же такое «есть» Амазон и что подразумевают когда о нем думают и говорят? Термин «www.amazon.com» - это символ, который мы используем в нашем языки и он что-то означает. Вопрос о том, что такое символ слишком философский, поэтому я не буду его рассматривать в этой работе. Вместо этого я обращу свое внимание на практическое использование URI для идентификации концепций и вещей.

URI можно использовать для идентификации различных сущностей.

Для того чтобы идентифицировать книгу «Два капитана» Вениамина Каверина мы можем воспользоваться страницей на веб сайте Амазон, на которой описывается эта книга. Если ввести URL адрес в адресную строку программы браузера отобразится веб страница с этой книгой. В высказывании «Я хочу почитать книгу “Два капитана”», URL страницы на Амазоне URL страницы можно использовать для идентификации книги. Существует несколько URI, которые ведут на страницу с этой книгой, поэтому здесь присутствует неточность и двусмысленность. Для людей, URI адреса вполне подходят – чтобы определить на какой ресурс он ссылается достаточно просто вставить его в адресную строку. Также очень просто идентифицировать самих людей, если они сохранены в приложениях вроде электронной книги контактов (адресная книга). После этого можно взять URI записи в адресной книге и использовать его в дальнейшем для ссылки на этого человека, что мы и будем делать в приложении SemanticDesktop.

## Проблемы

На текущий момент не существует единой онтологии и списка ресурсов и свойств этих ресурсов. Dublin core и FOAF не обеспечивают необходимой универсальности. В будущем ожидается существование множества онтологий, которые в дальшейшем должны будут объединиться. Консорциум W3C рекомендует пользоваться существующими онтологиями а не изобретать свои.

# Часть 3. Приложение Semantic Desktop

При старте сервера приложение Semantic Desktop автоматически транслирует информацию о контактах (в виде файла vCARD), расположенный в специальной директории и добавляет метаданные в семантический граф.

Извлечение информации из списка контактов (файл vCard)

Извлечение информации из списка сообщений (файл MBOX)

Добавление извлеченных метаданных в семантический граф

Добавление извлеченных метаданных в семантический граф

Добавление списка контактов в граф

Добавление списка сообщений электронной почты в граф

Извлеченную информацию о добавленных контактах можно просмотреть в веб-браузере в виде HTML документа.

# 

Также программа извлекает информацию о полученных сообщениях электронной почты сохраненных в файле типа MBOX.

Добавление ресурсов компьютера (файлов) в семантический граф происходит с помощью вспомогательной программы-клиента с графическим интерфейсом пользователя (GUI). Сама программа-клиент вызывается посредством контекстного меню файла в проводнике Windows

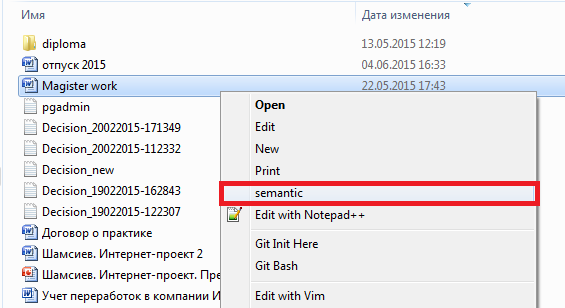


Рисунок . Вызов программы-клиента для файла Magister work.docx

# 

Рисунок . Добавление файла в семантическую сеть с помощью программы-клиента

После добавления файла в семантическую сеть в графе появляется новый ресурс, соединенный семантической двусторонней связью с другим ресурсом типа «контакт», загруженным ранее.

# 

Рисунок . Новый ресурс в графе

В формате RTF/XML новый ресурс в графе представлен следующим образом:

<rdf:Description rdf:about="file://artur.lazy-magister.org/data/Documents/Дипломная работа.docx">

<dc:creator rdf:resource="http://artur.lazy-magister.org/resources/contact/1"/>

<dc:title>Дипломная работа</dc:title>

<arlm:LABEL>Дипломная работа</arlm:LABEL>

<arlm:OWNER>http://artur.lazy-magister.org/resources/user</arlm:OWNER>

<rdf:type>file://artur.lazy-magister.org/types/file</rdf:type>

</rdf:Description>

Где file://artur.lazy-magister.org/data/Documents/Дипломная работа.docx – уникальный идентификатор ресурса (URI) типа «файл».

<dc:creator rdf:resource="http://artur.lazy-magister.org/resources/contact/1"/> **-** свойство, ссылающееся на создателя ресурса, в данном случае это http://artur.lazy-magister.org/resources/contact/1 - уникальный идентификатор ресурса типа «контакт».

<dc:title>Дипломная работа</dc:title> **-** свойство, означающее название ресурса.

<arlm:OWNER>http://artur.lazy-magister.org/resources/user</arlm:OWNER> - свойство указывающее на хозяина данного компьютера.

<rdf:type>file://artur.lazy-magister.org/types/file</rdf:type>

</rdf:Description> - свойство, обозначающее тип ресурса.

# Часть семантического графа в формате RTF/XML

<rdf:RDF  
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
 xmlns:dc="http://purl.org/dc/elements/1.1/"  
 xmlns:arlm="http://artur.lazy-magister.org/properties#"  
 xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#" >  
 <rdf:Description rdf:about="file://artur.lazy-magister.org/data/Documents/Догово о практике.docx">  
 <dc:creator rdf:resource="http://artur.lazy-magister.org/resources/contact/1"/>  
 <dc:creator rdf:resource="http://artur.lazy-magister.org/resources/contact/4"/>  
 <dc:creator rdf:resource="http://artur.lazy-magister.org/resources/contact/3"/>  
 <dc:title>Догово о практике</dc:title>  
 <arlm:LABEL>Догово о практике</arlm:LABEL>  
 <arlm:OWNER>http://artur.lazy-magister.org/resources/user</arlm:OWNER>  
 <rdf:type>file://artur.lazy-magister.org/types/file</rdf:type>  
 </rdf:Description>  
 <rdf:Description rdf:about="http://artur.lazy-magister.org/resources/mail/3">  
 <arlm:FROM>-NO-REPLY-@superjob.ru</arlm:FROM>  
 <arlm:BODY>PCFT0NUWVBFIGh0bWwgUFVCTElDICItLy9XM0MvL0RURCBYSFRNTCAxLjAgU3RyaWN0Ly9FTiIgImh0dHrLmdpZiI=</arlm:BODY>  
 <dc:date>15-04-2015 13:09</dc:date>  
 <dc:subject> Как описать опыт работы в резюме</dc:subject>  
 <arlm:OWNER>http://artur.lazy-magister.org/resources/user</arlm:OWNER>  
 <rdf:type>http://artur.lazy-magister.org/types/mail</rdf:type>  
 </rdf:Description>  
 <rdf:Description rdf:about="file://artur.lazy-magister.org/data/Documents/Отпуск 2015.docx">  
 <dc:creator rdf:resource="http://artur.lazy-magister.org/resources/contact/3"/>  
 <dc:title>Отпуск 2015</dc:title>  
 <arlm:LABEL>Отпуск 2015</arlm:LABEL>  
 <arlm:OWNER>http://artur.lazy-magister.org/resources/user</arlm:OWNER>  
 <rdf:type>file://artur.lazy-magister.org/types/file</rdf:type>  
 </rdf:Description>  
 <rdf:Description rdf:about="http://artur.lazy-magister.org/resources/contact/16">  
 <rdf:type>http://artur.lazy-magister.org/types/contact</rdf:type>  
 <arlm:OWNER>http://artur.lazy-magister.org/resources/user</arlm:OWNER>  
 <arlm:LABEL>Анастасия Краснокутская</arlm:LABEL>  
 <vcard:Given>Анастасия</vcard:Given>  
 <vcard:Family>Краснокутская</vcard:Family>  
 <vcard:EMAIL>akrasnokutskaya@star-staff.ru</vcard:EMAIL>  
 </rdf:Description>  
</rdf:RDF>

## Исходный код. Компонент, отвечающий за добавление файлов mp3 в семантическую сеть

package org.arthan.semantic.service.graph.impl;  
  
import com.google.common.base.Preconditions;  
import com.google.common.base.Strings;  
import com.hp.hpl.jena.rdf.model.Resource;  
import org.arthan.semantic.model.File;  
import org.arthan.semantic.model.MP3File;  
import org.arthan.semantic.service.graph.GraphMusicService;  
import org.arthan.semantic.service.graph.GraphRepository;  
import org.arthan.semantic.service.graph.Props;  
import org.arthan.semantic.service.graph.ResourceType;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Component;  
  
import java.util.List;  
import java.util.stream.Collectors;  
  
/\*\*  
 \* Created by artur.shamsiev on 15.05.2015  
 \*/  
  
@Component  
public class GraphMusicServiceImpl implements GraphMusicService {  
  
 @Autowired  
 GraphRepository graphRep;  
  
 @Override  
 public void writeMusicToGraph(MP3File mp3File) {  
 addMusicToGraph(mp3File);  
  
 graphRep.writeGraph();  
 }  
  
 @Override  
 public Resource addMusicToGraph(MP3File mp3File) {  
 Preconditions.*checkNotNull*(  
 Strings.*emptyToNull*(mp3File.getPath()),  
 "Не могу добавить музыкальный файл без идентификатора-пути в граф"  
 );  
 Resource mp3Res = graphRep.addResource(mp3File.getUriID(), ResourceType.*MP3FILE*.getUri());  
 mp3Res.addProperty(Props.*MUSIC\_ALBUM*, mp3File.getAlbum());  
 mp3Res.addProperty(Props.*PERFORMER*, mp3File.getPerformer());  
 mp3Res.addProperty(Props.*MUSIC\_GENRE*, mp3File.getGenre());  
 mp3Res.addProperty(Props.*MUSIC\_TITLE*, mp3File.getMusicTitle());  
  
 return mp3Res;  
 }  
  
  
 @Override  
 public List<MP3File> allUserMusic() {  
 List<Resource> musicResList = graphRep.findResourcesWithType(ResourceType.*MP3FILE*);  
  
 return musicResList.stream()  
 .map(this::resourceToMP3File).collect(Collectors.*toList*());  
 }  
  
 @Override  
 public MP3File findMusicByID(String musicPath) {  
 Resource musicRes = graphRep.getResource(File.*URI* + musicPath);  
 return resourceToMP3File(musicRes);  
 }  
  
 private MP3File resourceToMP3File(Resource resource) {  
 MP3File mp3File = MP3File.*fromUri*(resource.getURI());  
 mp3File.setMusicTitle(resource.getProperty(Props.*MUSIC\_TITLE*).getObject().toString());  
 mp3File.setGenre(resource.getProperty(Props.*MUSIC\_GENRE*).getObject().toString());  
 mp3File.setPerformer(resource.getProperty(Props.*PERFORMER*).getObject().toString());  
 mp3File.setAlbum(resource.getProperty(Props.*MUSIC\_ALBUM*).getObject().toString());  
  
 return mp3File;  
 }  
}

# Заключение

Хоть технологии Semantic Web разрабатывались для развертывания семантической сети в глобальной сети Интернет, они достаточно универсальный для построения семантических графов, распределенных по электронным сетям любого уровня. Так, с помощью языка описания семантических документов RDF и идентификаторов URI можно выстроить семантическую сеть на уровне предприятия или даже группы предприятий (например, для стандартизации финансовых показателей в конгломерате компаний). Более того, в семантическую сеть можно даже превратить ресурсы одного персонального компьютера с помощью определенных приложений, таких как персональный информационный менеджер или даже специализированных операционных систем, построенных на идее семантически соединенных файлов и их метаданных в компьютере (BeOS).

Практической реализацией этой идеи является так называемый Семантический рабочий стол – компьютерный пользовательский интерфейс, который дает возможность управления данными на персональном компьютере таким образом, что упрощается автоматическая обработка этих данных и взаимодействие между различными приложениями. Не обладая качественными [метаданными](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%B0%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5), компьютеры не способны без труда узнавать множество зачастую необходимых атрибутов файлов. Однако если позволить компьютеру управлять данными с помощью технологий семантической паутины, возможна смысловая обработка всей информации в системе, что превращает персональный компьютер в персональную базу знаний.

Семантические графы и их электронные представления хоть и находятся на заре своего развития, имеют поистине безграничный потенциал. Их не зря называют будущим Интернета и баз знаний.