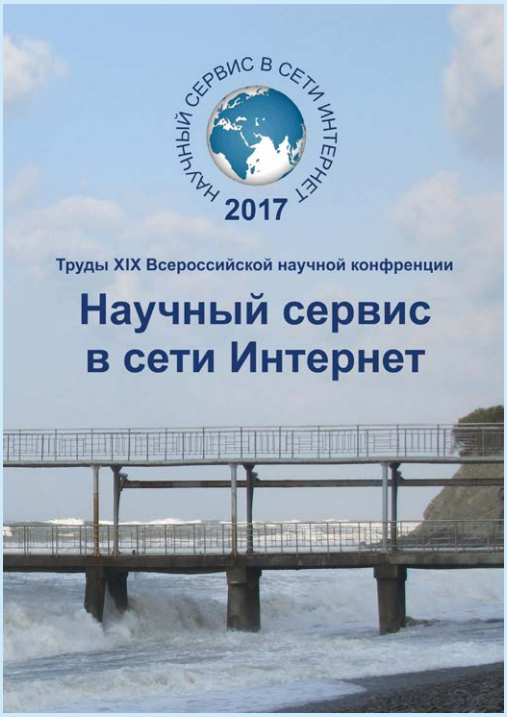
ИПМ им.М.В.Келдыша РАН

Абрау-2017 • Труды конференции



З.В. Апанович

Преподавание методов Semantic Web разработчикам программного обеспечения

Рекомендуемая форма библиографической ссылки

Апанович З.В. Преподавание методов Semantic Web разработчикам программного обеспечения // Научный сервис в сети Интернет: труды XIX Всероссийской научной конференции (18-23 сентября 2017 г., г. Новороссийск). — М.: ИПМ им. М.В.Келдыша, 2017. — С. 9-20. — URL: http://keldysh.ru/abrau/2017/37.pdf doi:10.20948/abrau-2017-37

Размещена также презентация к докладу

Преподавание методов Semantic Web разработчикам программного обеспечения

З.В. Апанович[1,2]

[1] *Институт систем информатики им. А.П. Ершова СО РАН,*
[2] *Новосибирский государственный университет Новосибирск*
apanovich@iis.nsk.su

Аннотация. С момента возникновения идеи Semantic Web в 2001 году это направление стремительно развивалось. В последние годы появилось осознание того, что ценность данных пропорциональна их взаимосвязанности не только Интернете, но и во многих приложениях, основанных на комбинации открытых и корпоративных данных. В данной работе обсуждаются как важные аспекты эволюции направления Semantic Web, так и преподавание методов Semantic Web

Ключевые слова: Semantic Web, RDF, RDFS, OWL, SPARQL

Научное направление Semantic Web, возникшее в 2001 году, развивается большими темпами. За последние годы размер Облака связанных данных значительно возрос и в настоящее время насчитывает порядка 10 000 наборов открытых данных в формате RDF. Google, обслуживающий около 9 млрд. запросов в день, создал в 2012 году Google Knowledge Graph[1], который насчитывает более 18 млрд. фактов, и составляет основу семантического поиска этой компании. Аналогичные проприетарные графы знаний используют для семантического поиска такие компании как Microsoft[2], Yandex и др. Снипеты, расширенные структурированными данными, и панели устранения неоднозначностей стали повсеместной практикой ответов на поисковые запросы. Помимо этого, были созданы глобальные словари, такие как schema.org, на основе которых разработчики коммерческих Интернет-сайтов, встраивают в свои html-страницы структурированные описания данных, улучшая их видимость. В настоящее время более 540 миллионов HTML страниц имеют встроенные структурированные описания данных. Значительные работы по созданию и предоставлению доступа к Связанным данным ведутся и в России [1, 2]. На базе уже существующих структурированных данных разрабатываются многочисленные приложения, что делает более важным и нужным знакомство с этим направлением современных

[1] http://googleblog.blogspot.co.uk/2012/05/introducing-knowledge-graph-things-not.html

[2] http://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing/

ИТ-специалистов. В данной работе обсуждаются как важные аспекты эволюции направления Semantic Web, так и методы преподавания дисциплины «Принципы, методы и средства связывания данных в приложениях Semantic Web»

1. Принципы связанных данных и существующие наборы данных

Знакомство с топологией Облака связанных данных является первым шагом при знакомстве с Semantic Web[3]. В облако входят наборы данных, разбитые по следующим тематикам: кросс-доменные, такие как DBPedia и Wikidata, географические (Geonames), библиотечные (Europeana, Worldcat, VIAF), наборы, посвященные наукам о жизни (Bio2RDF, Uniprot), лингвистические (BabelNet), средства массовой информации (BBC, New York Times), социальные сети. С момента своего создания в 2007 году, Облако связанных данных значительно расширилось, и в настоящее время, его элементы рассредоточены по нескольким каталогам, таким как Datahub.io, publicdata.eu, data.gov, open.canada.ca. Имеется специальный набор данных LODStats [3], который поддерживает статистику о текущем состоянии облака данных.

Все наборы облака LOD созданы в соответствии с базовыми принципами Linked Data:

— использовать URIs для определения сущностей;
— использовать HTTP URIs таким образом, чтобы на эти сущности можно было ссылаться и чтобы они могли быть найденными человеком и программным клиентом;
— при разыменовании URI, предоставлять полезную информацию о соответствующей сущности, используя такие стандарты, как RDF и SPARQL;
— при публиковании данных в веб включать в описание ссылки на другие наборы данных.

В контексте изучения принципов Связанных Данных очень важным является осознание того, что URI (IRI) являются, прежде всего, инструментом *именования* (создания уникальных глобальных идентификаторов) как для информационных объектов, таких как HTML-страницы, так и для объектов реального мира. Эти объекты реального мира могут иметь описания как в человеко-читаемом формате, таком как страницы HTML, так и в формате, понятном компьютеру (различные синтаксические формы RDF), и что выбор подходящего описания объекта реального мира осуществляется при помощи такой процедуры, как *обсуждение контента*. Все эти понятия легко продемонстрировать на примере произвольного набора данных, хранящегося в облаке связанных данных. Например, объект реального мира Москва имеет URI <dbpedia.org/resource/Moscow> в наборе данных Dbpedia.org, html-страница

[3] http://lod-cloud.net/

этого ресурса имеет URI <http://dbpedia.org/page/Russia>, а URI <http://dbpedia.org/data/Moscow> соответствует файлу в формате RDF/XML. Одно из первых практических заданий, предлагаемых студентам, состоит в самостоятельном ознакомлении с одним из выбранных наборов данных и составление краткого отчета, отвечающего на такие вопросы как: По каким правилам формируются URI данного набора данных? Какая онтология используется для структурирования этого набора данных? С какими наборами данных связан данный набор? Какие способы доступа имеются к данному набору данных? В соответствии с какими лицензиями он используется? И т.д.

2. Знакомство с моделью данных RDF

Модель данных RDF является стандартом, разработанным W3C, и предназначена для интегрированного представления информации, которая происходит из нескольких источников и гетерогенно структурирована (представлена с использованием различных схем). В этой модели граф RDF является множеством триплетов вида <subject, predicate, object>, где субъект и предикат всегда являются URI, а объект может быть как URI, так и литералом (строкой). Предикат указывает на отношение, существующее между субъектом и объектом. Благодаря тому, что URI является глобальным уникальным идентификатором, отдельные триплеты с одинаковыми субъектами могут «склеиваться», образуя ориентированный граф с помеченными ребрами.

Чтобы опубликовать граф RDF в Интернете, он сначала должен быть сериализован при помощи синтаксиса RDF. Количество синтаксических форм RDF значительно возросло за последние годы. Помимо таких форм как RDF/XML, Turtle и N-Triples, появились две синтаксические формы, ориентированные на описание нескольких именованных графов Trig и N-Quads, а также форматы RDFa и JSON-LD, позволяющие встраивать структурированные данные в HTML- страницы. В курсе демонстрируются все указанные варианты синтаксических форм, а также осуществляется знакомство с инструментами, позволяющими конвертацию данных между этими представлениями. Помимо этого, рассматриваются такие понятия модели RDF, как пустые узлы, типизированные литералы, абсолютный и относительный URI, идентификатор фрагмента, понятие базы и т.д. Все основные примеры рассматриваются параллельно в форматах RDF/XML и Turtle.

Задание, позволяющее быстро познакомиться с простейшими этапами создания Связанных данных, выглядит следующим образом. Прочитать описание основных классов и свойств словаря FOAF (Friend Of A Friend)[4], затем создать описание собственной персоны в формате RDF/XML при помощи приложения FOAF-a-matic[5], после чего вручную добавить к сгенерированному

[4] http://xmlns.com/foaf/spec/
[5] http://www.ldodds.com/foaf/foaf-a-matic.html

файлу в формате RDF/XML пять синтаксически правильных триплетов, использующих в качестве предикатов различные предикаты, описанные в разных наборах данных облака LOD. Проверить синтаксическую правильность полученного файла при помощи RDF-валидатора[6].

После знакомства с основными понятиями RDF проводится вводная лекция по SPARQL, знакомящая с его основными конструкциями. Эта вводная часть воспринимается студентами достаточно легко, поскольку SPARQL имеет много схожих черт с SQL. На последующих занятиях теоретические вопросы Связанных данных рассматриваются параллельно с углубленным изучением SPARQL 1.1. и использованием запросов SPARQL для ознакомления с понятиями Связанных данных. В частности, знакомство с форматами RDFS и OWL сопровождается знакомством с классами и свойствами конкретных словарей, описанными в Облаке связанных данных. Их исследование базируется на запросах SPARQL 1.1.

3. Языки описания словарей RDFS и OWL и словари, используемые в облаке связанных данных

RDF Schema (RDFS) и язык Web Ontology Language (OWL) являются наиболее известными языками описания словарей (онтологий) в области Semantic Web. RDFS – это минимальный язык описания онтологии, построенный поверх RDF, который позволяет определить:

- классы индивидуальных ресурсов;
- свойства, связывающие два ресурса;
- иерархии классов;
- иерархии свойств;
- ограничения на область определения и область значений свойств (*rdfs:domain, rdfs:range*).

OWL обеспечивает разработчиков онтологий гораздо более сложными и полезными конструкциями, чем RDFS, благодаря чему популярность OWL постоянно растет. Как и в RDFS, основными элементами OWL являются классы, свойства и индивиды, которые являются членами классов. Свойства OWL являются бинарными отношениями, среди них принято выделять ObjectProperty и DataTypeProperty. Свойства типа ObjectProperty связывают двух индивидов, тогда как DataTypeProperty связывают индивида с литералом.

В курсе по Semantic Web делается акцент на том, что стандарты RDFS и OWL, связаны между собой, что основной синтаксической формой OWL2 является RDF/XML, и, стало быть, к схемам, описанным в форматах *rdfs* и *owl*, можно писать запросы SPARQL. Например, в том, что *owl:Class* является подклассом *rdfs:Class*, а *owl:DatatypeProperty* и *owl:ObjectProperty* являются подклассами *rdf:Property* легко убедиться, при помощи простого запроса SPARQL:

[6] http://www.w3.org/RDF/Validator/

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:< http://www.w3.org/2000/01/rdf-schema#>
SELECT ?s ?o
FROM <http://www.w3.org/2002/07/owl>
WHERE  {?s rdfs:subClassOf ?o }
```

Одним из важнейших свойств Semantic Web является возможность интегрировать данные из различных источников, описанных при помощи разных *словарей*. Словарь (онтология) состоит из классов, свойств и типов данных, которые определяют смысл данных. RDF-словари сами выражаются и публикуются в соответствии с принципами связанных данных. Web of Data применяет двойственный подход к разнородности данных. Во-первых, рекомендуется уменьшать уровень разнородности за счет использования терминов из широко используемых словарей. Во-вторых, в случае использования проприетарных терминов, каждый словарный термин должен быть связан со своим описанием. Кроме этого, рекомендуется публиковать файлы соответствий между терминами из разных словарей в виде RDF.

Для того чтобы иметь возможность повторно использовать имеющиеся словари, надо знать, где их найти. Большое количество словарей может быть обнаружено при помощи поискового движка Watson[7]. Около 600 различных словарей представлены в специализированном наборе данных Linked Open Vocabularies (LOV) [4]. Набор данных LOV строит экосистему словарей, поддерживающую их повторное использование. Он подсчитывает популярность каждого термина словаря, а также устанавливает отношения между словарями, используя словарь VOAF.

Помимо того, что LOV является весьма полезным вспомогательным средством при поиске онтологий, он является также прекрасным инструментом для использования в образовательных целях. В частности, благодаря имеющейся конечной точке SPARQL, можно знакомить учащихся со многими вопросами устройства онтологий, как это будет показано в разделе 4.

Все словари, знакомство с которыми осуществляется в данном курсе (VOID, Dublin Core, FOAF, goodRelations, schema.org, dbpedia.org/ontology, owl, rdfs и др), присутствуют в наборе данных LOV.

4. Запросы SPARQL – важный инструмент работы с данными

Запросы SPARQL [5] позволяют не только получить исчерпывающую информацию о любом незнакомом наборе данных, но также копировать данные, создавать новые данные, конвертировать данные, осуществлять контроль качества данных, не только проверяя выполнение таких ограничений как корректность типов данных, но и соответствие бизнес правилам. Так, например, при знакомстве с новым набором данных первый вопрос, на который надо ответить «что это за данные?»:

[7] http://watson.kmi.open.ac.uk/WatsonWUI/

SELECT * WHERE { ?s ?p ?o . }#LIMIT 50

Поскольку это запрос к графу по умолчанию, а многие хранилища триплетов хранят данные в именованных графах, вторым уместным запросом будет запрос, который покажет URI всех именованных графов, входящих в данный набор:

SELECT DISTINCT ?g WHERE { GRAPH ?g {?s ?p ?o } }

В частности, при запуске этого запроса на конечной точке SPARQL словаря LOV выдается список всех словарей (онтологий) имеющихся в настоящий момент в этом наборе данных, а также URI всех словарей, по которым можно строить запросы SPARQL к любому из описанных словарей. Также, в курсе рассматриваются такие типы запросов, как: «Какие классы декларированы в данном наборе данных? Какие свойства декларированы? Какие значения имеет данное свойство? Какие классы используются? Сколько экземпляров имеется у каждого класса заданного набора данных? Какие классы используют определенное свойство?». Ответы на все выше перечисленные базовые вопросы студенты учатся находить для произвольного набора данных, который они видят первый раз в жизни.

Помимо вопросов, направленных на исследование незнакомого набора данных, на практических занятиях рассматриваются такие вопросы, как объединение данных из нескольких разрозненных наборов, конструирование новых данных на основе имеющихся, преобразование данных, описанных при помощи одной онтологии, в данные, основанные на другой онтологии, автоматизированное решение задачи идентификации сущностей (генерация отношений *owl:sameAs*), установление соответствия между классами и свойствами разных онтологий при помощи запросов SPARQL. Наконец, изучается вопрос контроля качества наборов связанных данных при помощи запросов SPARQL.

5. Инструменты для работы со Связанными данными

Процесс предоставления связанных данных часто характеризуется как жизненный цикл, в котором данные создаются, повторно используя другие источники данных, преобразуются, а затем делаются доступными для использования. Вновь созданные данные могут стать одним из источников данных, используемых при подготовке дальнейших данных. В упрощенной формулировке жизненного цикла связанных данных можно выделить три основных стадии.

Создание связанных данных: Извлечение данных, создание HTTP URI в качестве имен, и выбор словарей для описания предикатов и для классификации сущностей.

Связывание данных: Поиск и выражение связей между сущностями-синонимами из разных наборов данных.

Публикация связанных данных: Создание метаданных о наборе данных и обеспечение доступа к набору данных.

Инструменты, применяемые на каждой стадии жизненного цикла, быстро эволюционируют. Каждый год появляются новые инструменты, а некоторые инструменты устаревают и перестают использоваться. Поэтому эта часть курса нуждается в ежегодном обновлении. Тем не менее, есть некоторые базовые инструменты, знакомство с которыми важно для понимания курса.

Наиболее распространенные форматы, на основе которых осуществляется генерация данных RDF, — это таблицы или табличные данные, реляционные базы данных и текстовые данные. Для разных видов данных существуют различные стратегии и инструменты, позволяющие преобразовать их в RDF. Реляционные БД обычно отображаются в RDF при помощи правил установления соответствия и инструментов, таких как D2R [6]. В рамках данного курса, студенты знакомятся с языком R2RML, который является рекомендацией W3C для определения отображения между реляционными базами данных и связанными данными. Также студенты знакомятся с простым инструментом OpenRefine[8], который позволяет решать задачу преобразования табличных данных в RDF в интерактивном режиме. В качестве инструментов извлечения данных из свободного текста рассматриваются Open Calais[9] и DBpedia Spotlight [7]. На этапе связывания данных решается задача установления связей идентичности между отдельными индивидами, а также установления словарных связей между свойствами и классами, описанных в разных онтологиях словарях. Для автоматизированного решения первой задачи часто используется инструмент SILK [8], с которым осуществляется знакомство в данном курсе. Дополнительно, в курсе изучается вопрос, как триплеты с предикатом *owl:sameAs* могут быть созданы в результате логического вывода и при помощи запросов SPARQL. Что касается задачи создания словарных связей, на практических занятиях демонстрируется, как можно решать эту задачу при помощи запросов SPARQL.

После того, как наборы данных RDF был созданы и взаимосвязаны, процесс публикации включает такие задачи как создание метаданных для описания набора данных, предоставление доступа к набору данных и валидация набора данных. В качестве основной формы предоставления связанных рассматриваются каталог DataHub, а также хранилища триплетов, такие как Jena[10] и Virtuoso[11]. В качестве инструмента валидации данных студенты знакомятся с RDF-валидатором, и RDF Triple-Checker[12], который помогает находить опечатки и распространенные ошибки в данных RDF.

[8] http://openrefine.org/
[9] http://www.opencalais.com/opencalais-demo/
[10] http://jena.apache.org
[11] https://virtuoso.openlinksw.com/
[12] http://graphite.ecs.soton.ac.uk/checker/

6. Приложения Связанных данных

Приложения связанных данных являются тем, что подтверждает ценность этого направления. Среди существующих приложений можно выделить следующие три группы.

1) *Навигаторы Связанных данных,* которые разыменовывают URI, чтобы получить описание ресурса. Типичным примером навигатора Связанных данных являются DBPedia.org.

2) *Поисковые системы связанных данных*, которые позволяют посылать запросы к связанным данным. В отличие от обычных поисковых систем, семантическая поисковая система используется для поиска онтологий, словарей и документов RDF. К таким системам относятся Watson и LOV. Системы Семантического поиска, встроенные в поисковые системы Google и Bing, опираются на внутренние Графы Знаний и позволяют помимо поиска по ключевым словам выдавать дополнительную информацию о сущностях, отображенных на эти Графы Знаний. Поисковые системы, основанные на Графах Знаний, все чаще используются в промышленности. Появляется все больше коммерческих приложений, использующих интеграцию данных о продуктах, услугах, предложениях работы и т.д. [9].

3) *Приложения Связанных данных, ориентированные на конкретную предметную область*. Эти приложения создаются для решения конкретного круга проблем в указанном домене. Подавляющее большинство приложений Связанных данных попадают в эту третью категорию. Примером приложения семантических технологий являются многочисленные разделы портала bbc.com. Для управления контентом этого портала реализована специальная архитектура, которая называется «Динамическая Семантическая Публикация» и направлена на автоматизацию агрегации или публикацию взаимосвязанного контента в рамках портала BBC[13]. Другое интересное приложение, которое рассматривается в курсе – Open Pharmacology Space[14] - семантическая исследовательская среда для фармакологии, ориентированная на разработку новых лекарств. Она позволяет интегрировать данные из множественных источников информации от баз данных белков и химических соединений до моделей биологических путей. Еще один интересный проект GRAVITATE[15] направлен на создание инструментов, позволяющих археологам реконструировать разрушенные культурные объекты, части которых хранятся в разных коллекциях.

Помимо примеров конкретных приложений, в курсе рассматривается архитектура приложений, работающих со Связанными данными.

[13] http://www.bbc.co.uk/blogs/bbcinternet/2012/04/sports_dynamic_semantic.html
[14] http://www.openphacts.org/open-phacts-discovery-platform
[15] http://gravitate-project.eu/

Заключение

В данной работе кратко представлены основные части курса, который все еще находится в процессе развития, поскольку в процессе развития находится научное направление Semantic Web. Наблюдается процесс интеграции знаний из Облака Открытых Связанных данных с данными, встроенными в html-страницы, методы машинного обучения используются для улучшения качества данных Semantic Web, и одновременно методы Semantic Web используются для поддержки открытия новых знаний [10]. Это значит, что потребность в специалистах, способных работать в этом направлении, будет и дальше возрастать.

Литература

1. Марчук А.Г., Марчук П.А. Особенности построения цифровых библиотек со связанным контентом // Труды RCDL'2010, Казань, 2010. — 19-23.
2. Серебряков В.А., Шорин О. Н. Проблемы семантической интеграции библиотечных данных // Библиотековедение. — 2014. — № 5 .— С. 45-51
3. Ermilov I., Lehmann J., Martin M., Auer S. LODStats: The Data Web Census Dataset //ISWC 2016. — pp. 38-46.
4. Vandenbusschea P-Y, Atemezingb G. A., Poveda-Villalónc M., Vatantd B., Linked Open Vocabularies (LOV): a gateway to reusable semantic vocabularies on the Web// Semantic Web 1. — 2014. — 1–5
5. DuCharme B. Learning SPARQL, Second Edition, O'Reilly Media, Inc. URL: http://it-ebooks.info/book/2574/
6. D.-E. Spanos, P. Stavrou, N. Mitrou, Bringing relational databases into the semantic web: A survey, Semant. Web 3 (2) . — 2012. — pp. 169–209.
7. Mendes P.N., Jakob M., García-Silva A., Bizer C., Dbpedia spotlight: Shedding light on the web of documents, in: Proceedings of the 7th International Conference on Semantic Systems, I-Semantics'11, ACM, New York, NY,USA, 2011. — pp. 1–8.
8. Isele R., Jentzsch A., Bizer Ch. Silk Server — Adding missing Links while consuming Linked Data// 1st International Workshop on Consuming Linked Data (COLD 2010) . — 2010.
9. Paulheim H., Automatic Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods// Semantic Web. — 2016
10. Ristoski P., Paulheim H. SemanticWeb in data mining and knowledge discovery: A comprehensive survey//Web Semantics: Science, Services and Agents on theWorldWideWeb 36. — 2016. — pp. 1–22.