

УДК 304.444

А.А. Сытник, Н.С. Вагарина, Н.И. Мельникова, Т.Э. Шульга, М.Ю. Апсаликов**О СОЗДАНИИ ИНСТРУМЕНТОВ ИЗУЧЕНИЯ СЕМАНТИЧЕСКОГО ВЕБА**

Потребность в методическом обеспечении преподавания технологий семантического веба обусловила создание соответствующих учебно-методических материалов и программных средств. Таковыми являются представленные в статье RDF-редактор MakeSense и SPARQL-тренажёр. На оба приложения получены свидетельства о государственной регистрации программ для ЭВМ и проведена апробация в учебном процессе при проведении занятий у бакалавров направления «Информатика и вычислительная техника». Проведённая апробация RDF-редактора MakeSense и SPARQL-тренажёра показала пути наращивания их функционала для работы с большими данными и федеративными запросами, а также необходимость расширения спектра учебных программных средств в области технологий семантического веба.

Технологии семантического веба, RDF-редактор Make Sense, SPARQL-тренажер

A.A. Sytnik, N.S. Melnikova, N.I. Vagarina, T.E. Shulga, M.Yu. Apsalikov**SEMANTIC WEB TOOLS**

The need for methodological support in training Semantic Web technologies led to creation of appropriate teaching materials and software. Two of the applications are presented in the paper: the RDF editor and SPARQL learning tool. These applications

received the certificates of state registration of computer programs. They are used in the learning process to train bachelors in Computer Science and Engineering. The functions of these applications can be expanded to work with large data and requests by the Federal agencies.

Semantic Web Technologies, RDF-editor MakeSense, SPARQL learning tool

Появление интернета стало большим шагом на пути оперативного доступа к информации. С каждым годом объём информации в интернете растёт быстрыми темпами. И встаёт вопрос о быстром и точном поиске интересующей пользователя информации. Одним из возможных решений этой проблемы может стать концепция семантического веба [1]. Перспективной семантической моделью представления данных на текущий момент является модель RDF (Resource Description Framework) [2]. Для доступа к данным, представленным с помощью RDF, применяются различные языки запросов к данным. Наиболее распространенным из подобных языков запросов является SPARQL [3].

Для обработки и понимания сути идеи семантического веба необходима учебная точка доступа к RDF данным, позволяющая осуществлять поиск на основе запросов языка SPARQL. Программные средства для работы по языком SPARQL несомненно, существуют [4]. Их можно разделить на два класса. К первому классу относятся точки доступа, по-английски называемые SPARQL-endpoint. Стоит отметить, что существующие точки доступа не позволяют в полной мере исследовать возможности SPARQL. Это объясняется тем, что некоторые запросы невозможно выполнить на готовых точках доступа. Во-вторых, не известна структура данных, к которым подаётся запрос. В-третьих, как правило, речь идёт о профессиональных наборах данных, которые очень велики по объёму, что затрудняет их использование в учебном процессе.

Ко второму классу можно отнести программное обеспечение, которое в большей степени ориентировано на практическое изучение SPARQL. Как правило, его возможно развернуть и запустить на локальном компьютере для выполнения запросов на собственных локальных данных. Но количество таких приложений невелико. К тому же готовых полноценных библиотек, предоставляющих возможность разрабатывать готовые приложения, очень мало. Для сравнения функциональных возможностей существующих средств для изучения SPARQL были выбраны два программных продукта: k4SparqlTrainer и Fuseki. Приложение k4SparqlTrainer представляет собой программное обеспечение (desktop) для изучения языка SPARQL. Используется только в качестве учебного программного продукта. В качестве ядра используется библиотека DotNetRdf. Приложение Fuseki представляет собой SPARQL сервер. Это приложение реализует, помимо возможностей, описанных в спецификации SPARQL, также возможности дополнительного SPARQL расширения, которое позволяет манипулировать триплетом в хранилище (обновлять, удалять, создавать). ПО Fuseki реализовано с помощью фреймворка Jena. При разработке приложений необходимо учесть достоинства и недостатки существующих аналогов, чтобы в результате получился максимально приспособленный для использования в учебных целях программный продукт. Результаты сравнительного анализа некоторых наиболее известных программных приложений указанных классов представлены в таблице.

Сравнительный анализ существующего ПО

Возможности	Приложения		
	k4SparqlTrainer	Fuseki	Разрабатываемое программное обеспечение
Уровень восприятия результата (удобный формат вывода)	Низкий	Средний	Высокий
Возможность анализировать данные, запрос и результат одновременно (на одной форме)	Да	Нет	Да
Возможность использовать как точку доступа	Нет	Да	Да
Возможность назначать имена графам	Нет	Да	Да
Тип приложения: desktop или web	desktop	web	web
Возможность оперировать с данными вручную (добавлять, удалять, редактировать) без необходимости загрузки новых данных из файла	Нет	Нет	Да
Удобство использования интерфейса пользователя	Высокое	Низкое	Высокое

Распространение технологий семантического веба требует создания новых образовательных траекторий в бакалавриате и магистратуре по направлению «Информатика и вычислительная техника» [5]. В связи с этим в СГТУ имени Гагарина Ю.А. актуализированы учебные планы в связи с развитием семантических технологий. Созданы рабочие программы новых дисциплин по выбору «Основы технологий семантического веба», «Структуры и стандарты связанных данных». Потребность в методическом обеспечении преподавания этих дисциплин обусловила создание учебно-методических материалов и программных средств для изучения модели представления данных RDF и языка запросов SPARQL (SPARQL Protocol and RDF Query Language) к семантически описанным данным. Такими программными средствами являются RDF-редактор MakeSense и SPARQL-тренажёр.

Редактор MakeSense позволяет в простой и наглядной форме представлять информацию в формате RDF, изучать и сравнивать различные сериализации RDF, строить соответствующие RDF-графы. Наличие такого инструмента необходимо как для студента, так и для преподавателя, в частности для создания учебных примеров, иллюстрирующих лекционный материал, а также контрольных и тестовых заданий, предлагаемых студентам на лабораторных занятиях.

В качестве интегрированной среды разработки данного приложения была выбрана Microsoft Visual Studio 2013 Ultimate. Для низкоуровневой работы с RDF-данными были использованы исходные коды библиотеки dotNetRdf. В библиотеке были изменены модификаторы доступа ряда методов и она была интегрирована в проект как скомпилированная библиотека классов DLL. Компонент FastColoredTextBox был использован вместо стандартного компонента RichTextBox в связи с тем, что подсветка синтаксиса и вывод всплывающих сообщений об ошибке в компоненте RichTextBox реализуется со значительными затратами ресурсов.

Для разработки MakeSense было рассмотрено использование двух фреймворков: WinForms и WPF (Windows Presentation Foundation). WinForms – это API (Application Programming Interface) программирования интерфейсов для Windows, который является оберткой для неуправляемого кода Win32 API. WPF – это фреймворк для разработки интерфейсов для Windows с графической подсистемой, использующей язык XAML, использует для работы DirectX. По ряду нижеуказанных причин была выбрана технология WinForms: поддержка MDI-контейнеров; более удобная обработка горячих клавиш; производительность элемента, отвечающего за содержимое; наличие компонента FastColoredTextBox; большая распространенность по сравнению с WPF; меньшие затраты времени на разработку интерфейса приложений, не требовательных к ресурсам на отрисовку.

Интерфейс приложения представляет собой MDI-контейнер с созданием внутри него дочерних форм (рис. 1), в которых располагаются конкретные документы. В верхней части приложения располагается меню, откуда вызываются все основные команды. Часть команд дублируется в контекстном меню дочерних форм и горячими клавишами. Дочерние формы содержат две вкладки. На первой расположен код, который можно редактировать. Текст обладает подсветкой синтаксиса, реализованной с помощью регулярных выражений. Вторая вкладка отображает структуру документа в виде списка всех триплетов.

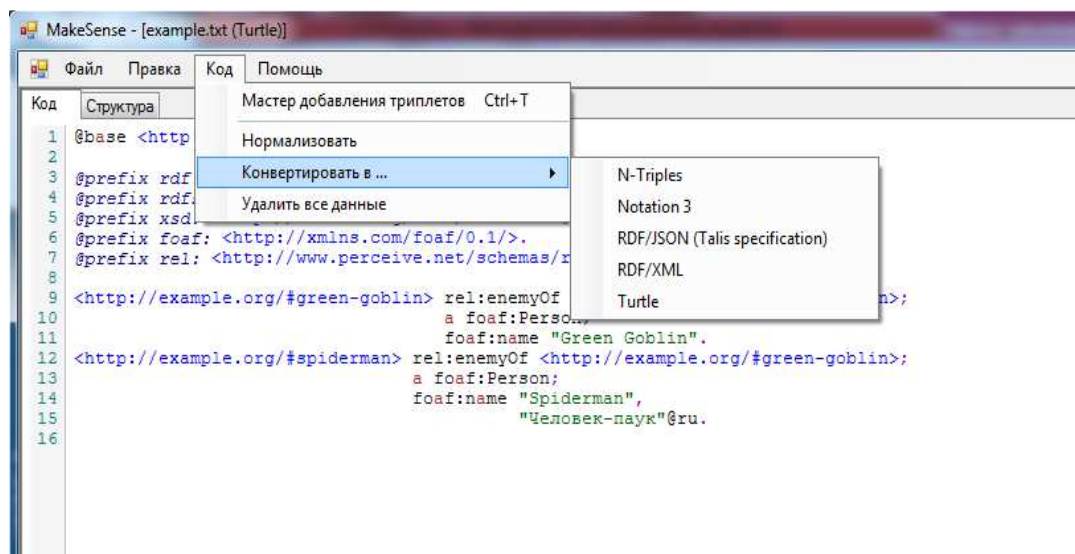


Рис. 1. Интерфейс редактора MakeSense

Отдельно стоит обратить внимание на мастер добавления триплетов (рис. 2) с пояснениями на русском языке. Этот мастер отличает разрабатываемое приложение от существующих аналогов.

Добавление триплета

Триплет «субъект – предикат – объект» является минимальной логической сущностью в RDF. Субъект – это сущность, которая может быть обыкновенной Web-страницей, ее частью или их совокупностью (в том числе и Web-сайтом целиком), либо чем-либо не доступным через Интернет (в том числе реальной вещью из окружающего мира). Описывается с помощью URI, может быть пустым. Предикатом является какой-либо аспект, характеристика или отношение, которое принадлежит субъекту, имеет специфический смысл, область допустимых значений, тип субъектов, которые может обслуживать и сценарий взаимодействия с другими предикатами. Объект – это конкретное значение предиката субъекта. Может быть представлен в виде литералов или URI.

Субъект
Пустой узел
Пустой узел
URI

Предикат
URI

Объект
Пустой узел

Добавить Отменить

Рис. 2. Мастер добавления триплетов

Мастер добавления триплетов отображает только доступные варианты узлов для конкретной сериализации и элементов триплета. Например, в различных сериализациях RDF по-разному используются пустые узлы. Эту особенность среди прочих мастер учитывает и предопределяет возможные варианты синтаксиса триплетов. В мастере предусмотрена автоподстановка ранее использованных узлов подходящего типа в RDF код. На практике в RDF-документе значения субъектов, объектов и предикатов часто повторяются. Следовательно, такая подстановка сделает добавление триплетов более удобным. Если необходимого триплета в выпадающем списке нет, то пользователь может ввести свой триплет, элементы которого затем будут добавлены в выпадающие списки мастера. Кроме того, функционал мастера предполагает обработку исключений. Пользователь может ввести в поля ввода для субъекта, предиката и объекта любые строковые значения. Однако не все из этих значений будут корректными. Например, пользователь вместо URI может ввести число, либо диалект RDF/XML требует, чтобы все URI были корректны с точки зрения стандартов W3C. Все эти варианты обрабатываются в мастере добавления триплетов. Если пользователь вводит что-либо с ошибкой, то триплет в код не добавляется, а пользователю предлагается исправить допущенные ошибки. Для разработки ПО была выбрана платформа Microsoft .NET Framework. Для реализации ПО был выбран язык программирования C#. Использовалась библиотека DotNetRdf, которая и отвечает за ядро системы, а именно за обработку данных и выполнение запросов. Приложение SPARQL-тренажер имеет web-интерфейс и включает пять web-форм: Главная, Данные, Запрос, Результат, Внешние запросы. Web форма Главная позволяет загружать, просматривать и редактировать данные, вводить SPARQL запрос и выполнять его, а также анализировать результат (рис. 3).

SPARQL Editor

Локальные запросы Внешние запросы

Данные Запрос

Результат

Выберите файл Файл не выбран Загрузить

Подключенные дополнительные графы и подграфы Очистить все

Рис. 3. Главная форма приложения SPARQL-тренажер

Web форма Данные необходима для более удобного и эффективного манипулирования данными (загружать, редактировать, удалять), а также для создания графов и входящих в них подграфов. Web форма Запрос используется для формирования запроса, а также для его выполнения. Web форма Результат необходима для более удобного анализа результата после выполнения запроса. Web форма Внешние запросы необходима для работы с данными внешних точек доступа.

Запросы возможно производить как к собственным данным, загружаемым из файла или вводя их вручную в специальном текстовом поле, так и к внешним существующим точкам доступа.

Данная программа позволяет исследовать весь функционал языка SPARQL, например, использование различных опций при формировании вывода результата. В частности, на рис. 4 приведён пример запроса с использованием опции OPTIONAL.

The screenshot shows a web interface for SPARQL queries. It is divided into three main sections: 'Данные' (Data), 'Запрос' (Query), and 'Результат' (Result).

Данные (Data): This section contains two lists of RDF triples. The first list has 5 triples for 'Sergei' and 'Olga'. The second list has 5 triples for 'Andrei'.

Запрос (Query): This section contains a SPARQL query. The query uses the 'PREFIX' statement to define namespaces for 'ab' and 'd'. It then uses a 'SELECT' statement to retrieve 'first', 'last', and 'workTel' properties. The query includes an 'OPTIONAL' clause to handle missing data for the 'workTel' property.

Результат (Result): This section displays the results of the query in a table. The table has three columns: 'first', 'last', and 'workTel'. It contains three rows of data, corresponding to the three individuals mentioned in the query.

first	last	workTel
Sergei	Ivanov	
Olga	Petrova	
Andrei	Sidorov	(333) 333-3333

Рис. 4. Запрос с блоком OPTIONAL

На оба приложения получены свидетельства о государственной регистрации программ для ЭВМ [6, 7]. Это позволяет использовать приложения в учебных целях согласно требованиям образовательных стандартов. Апробация редактора MakeSense и Sparql-тренажёра в реальном учебном процессе в 2014-15 учебном году при проведении занятий у студентов третьего курса направления «Информатика и вычислительная техника» показала их эффективность в организации учебного процесса и удобство пользовательского интерфейса для студентов. В результате изучения учебных продуктов студенты легко осваивают работу с профессиональными точками доступа к семантическим данным.

Проведённая апробация RDF-редактора MakeSense и SPARQL-тренажёра также показала пути наращивания их функционала. Направления наращивания функционала: Sparql-тренажёр для осуществления запросов для больших данных с использованием модели MapReduce, а также федеративные запросы с использованием Nadoop.

Кроме того, требуется расширение спектра учебных программных средств области технологий семантического веба, что будет способствовать актуализации подготовки студентов в области информатики. Логическим следующим шагом в этой области является, например, разработка учебного OWL-редактора для управления семантическими данными на более высоком уровне абстракции.

ЛИТЕРАТУРА

1. Berners-Lee T., Hendler J., Lassila O. The Semantic Web [Электронный ресурс]. URL: <http://www.cs.umd.edu/~golbeck/LBSC690/SemanticWeb.html> (дата обращения: 16.09.2015).
2. Klyne G., Carroll J., McBride B. RDF 1.1 Concepts and Abstract Syntax [Электронный ресурс]. URL: <http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/> (дата обращения: 16.09.2015).
3. Prud'hommeaux E., Seaborne A. SPARQL Query Language for RDF [Электронный ресурс]. URL: <http://www.w3.org/TR/rdf-sparql-query/>.
4. SparqlEndpoints - W3C Wiki [Электронный ресурс]. URL: <http://www.w3.org/wiki/SparqlEndpoints> (дата обращения: 16.09.2015).
5. Вагарина Н.С., Мельникова Н.И. Адаптация зарубежного опыта применения технологий семантического веб в образовании [Электронный ресурс]. URL: <http://sibac.info/2009-07-01-10-21-16/9834> (дата обращения: 16.09.2015).

6. С. 2015615980 от 28.05.2015 Российская Федерация. MakeSense/ Вагарина Н.С., Апсаликов М.Ю.; Правообладатель Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования Саратовский государственный технический университет имени Гагарина Ю.А. (СГТУ имени Гагарина Ю.А.) (RU).-№2015612914; заявл. 13.04.15

7. С. 2015615979 от 28.05.2015 Российская Федерация. Программный комплекс для изучения языка запросов SPARQL/ Вагарина Н.С., Попов С.М.; Правообладатель Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования Саратовский государственный технический университет имени Гагарина Ю.А. (СГТУ имени Гагарина Ю.А.) (RU).-№2015612916; заявл. 13.04.15

Сытник Александр Александрович –
доктор технических наук, профессор,
заведующий кафедрой «Информационные
системы и технологии» Саратовского
государственного технического университета
имени Гагарина Ю.А.

Alexander A. Sytnik –
Dr. Sc., Professor
Head Department: Information systems and technology
Yuri Gagarin State Technical University of Saratov

Вагарина Наталия Сергеевна –
доктор физико-математических наук, доцент
кафедры «Информационные системы
и технологии» Саратовского государственного
технического университета имени Гагарина Ю.А.

Nataliya S. Vagarina –
Ph. D., Associate Professor
Department of Information systems and technology
Yuri Gagarin State Technical University of Saratov

Мельникова Нина Ивановна –
доктор социологических наук, профессор
кафедры «Информационные системы
и технологии» Саратовского государственного
технического университета имени Гагарина Ю.А.

Nina I. Melnikova –
Dr. Sc., Professor
Department of Information systems and technology
Yuri Gagarin State Technical University of Saratov

Шульга Татьяна Эриковна –
доктор физико-математических наук, доцент,
заведующий кафедрой «Прикладная информатика
и программная инженерия» Саратовского
государственного технического университета
имени Гагарина Ю.А.

Tatyana E. Shulga –
Dr. Sc., Associate Professor
Head Department: Applied Informatics
and Software Engineering
Yuri Gagarin State Technical University of Saratov

Апсаликов Михаил Юрьевич –
студент
Саратовского государственного
технического университета имени Гагарина Ю.А.

Mikhail Yu. Apsalikov –
Student
Yuri Gagarin State Technical University of Saratov

Статья поступила в редакцию 12.05.15, принята к опубликованию 15.09.15