

УДК 004.04

**ВИЗУАЛИЗАЦИЯ
СЕМАНТИЧЕСКИХ МОДЕЛЕЙ ДАННЫХ**

Г.О. Легостаев

*ГОУ ВПО «Санкт-Петербургский государственный
университет», г. Санкт-Петербург*

Ключевые слова и фразы: граф; кластер; матрица; Traveling Salesmen Problem; Semantic Web; Web Ontology Language.

Аннотация: Разработана технология матричного представления семантических данных, выраженных в синтаксисе OWL, приведен алгоритм кластеризации полученного матричного представления, предложены пути усовершенствования визуальных представлений OWL-документов.

В активно развивающемся информационном обществе задача повышения эффективности использования бесконечно растущих объемов данных претворила в жизнь направление семантического моделирования информационных ресурсов – представление данных в виде информационной (или семантической) сети с последующим смысловым анализом компьютером.

Попытка создания семантической сети на основе World Wide Web (WWW), как универсального источника информации и знаний, получила название Semantic Web (семантическая паутина). Эта идея была предложена в 1998 году Тимом Бернерсом-Ли (Tim Berners-Lee), который, являясь разработчиком WWW, URI, HTTP и HTML, в настоящее время возглавляет консорциум W3C – лидер в развитии технологий для Web.

Архитектура Semantic Web предполагает наличие у любого ресурса (объекта), находящегося в такой сети, точного смысла, который однозначно определяет этот ресурс. Фактически это означает, что для каждого объекта Semantic Web, предназначенного для восприятия человеком, существует некоторый связанный с ним контекст, используемый машинами, для проведения однозначных логических заключений о свойствах этих ресурсов.

Одним из возможных способов компьютерной интерпретации семантических данных является использование средств Extensible Markup Language (XML) – языка описания ресурсов сети, и Resource Description

Легостаев Г.О. – аспирант кафедры «Технология программирования», e-mail: glegostaev@gmail.com, СПбГУ, г. Санкт-Петербург.

Framework (**RDF**) – языка описания метаданных, то есть информации о ресурсах. Другим, рекомендованным консорциумом W3C, способом формализации семантических данных является использование языка онтологий Web Ontology Language (**OWL**), который обладает расширенными возможностями для описания метаданных, по сравнению с RDF и одним из синтаксисов которого является XML.

Базой для OWL являются свойства, классы и индивиды, как инструменты описания семантических данных, представленных в виде множества объектов (классов, индивидов) с некоторым набором свойств. При этом OWL-документы формируются четко по принципам математической логики и, тем самым, позволяют выводить некоторые логические заключения об описываемых ресурсах.

Документы, выраженные в синтаксисе OWL, являются неудобными для чтения человеком, и задача визуализации семантических данных в удобное, интуитивно понятное для человека представление является в настоящее время актуальной для разработчиков и исследователей в области Semantic Web.

Целью настоящей статьи является разработка и анализ возможных способов визуализации семантических данных, описанных в синтаксисе OWL.

Визуализация семантических данных с помощью графа

Семантические данные, как элементы Semantic Web, представленные в RDF/OWL-синтаксисе образуют набор триплетов, каждый из которых состоит из субъекта, предиката и объекта [7]. В качестве субъектов, объектов и предикатов выступают имена данных, или, другими словами, имена ресурсов, имена сущностей реального мира (как конкретных, так и абстрактных). В целях однозначной интерпретации имен, имена для субъектов, предикатов, и объектов определяются унифицированным (единообразным) идентификатором ресурсов Uniform Resource Identifier (**URI**). Субъект и объект указывают на две сущности в реальном мире, а предикат определяет отношение между ними.

Тем самым, OWL-технология позволяет представить данные в виде графа, при этом сущности ($URI_1, URI_2, \dots, URI_k$) представляются как узлы графа, а отношения между сущностями ($URI^1, URI^2, \dots, URI^n$) – как дуги, или ребра.

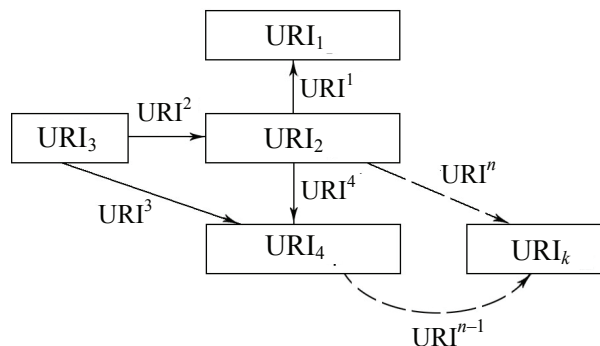


Рис. 1. Представление семантических данных в виде ориентированного графа

Однако такое представление нельзя назвать удовлетворительным, так как при большом количестве узлов результатом построения графа становится большая, загроможденная структура, которая с трудом поддается человеческому восприятию.

В качестве решения данной проблемы Э. Петрига (Emmanuel Petriga) из Парижского университета INRIA & Laboratoire de Recherche en Informatique (**LRI**) предложил трансформировать (преобразовать) полученный граф, используя таблицы стилей [3]. Однако данная технология обработки RDF/OWL-графов, все же остается малоэффективной при количестве узлов более 1000.

В данной статье приводится матричный способ визуализации OWL-графов для получения интуитивно понятных представлений содержимого OWL-графов. Затронут вопрос о способах улучшения полученных результатов посредством кластеризации матрицы.

MatrixExplorer как инструмент визуализации OWL-документов

В качестве инструмента для графического представления содержания OWL-документа в матричном виде был использован программный продукт MatrixExplorer, который подлежал доработке в целях адаптации к семантическим данным. В исходном виде продукт MatrixExplorer предназначен для исследователей в области социальных сетей и представляет собой систему визуализации социальных сетей одновременно в двух представлениях – в виде матрицы и графа [2].

Входными данными для программы MatrixExplorer является .dot файл – текстовый файл на специальном языке разметки графа, на выходе программы формируется граф в виде графического файла, который в дальнейшем может быть преобразован в матрицу.

Таким образом, в целях матричной визуализации OWL-документов, возникает задача преобразования OWL-документов в .dot файл. Решением данной задачи является разработанный программный продукт OwlToDotConverter.

OwlToDotConverter

Программный продукт OwlToDotConverter предназначен для конвертации OWL-документов в .dot файл.

Технологии: Java (пакет Jena для работы с OWL-документами), MySQL.

Для иллюстрации рассмотрим разработанные консорциумом W3C OWL-документы, описывающие онтологии вина и пищи [6]. Эти онтологии, описанные посредством языка OWL, представляют собой сложную совокупность элементов – индивидов, классов и их свойств со всевозможными связями, в зависимости от типов и свойств этих элементов. Так, два элемента онтологии могут одновременно рассматриваться как индивиды и как классы с различными типами связей, присущими этим индивидам и классам. Поэтому в целях последующего матричного представления данных, построение графа целесообразно осуществлять отдельно для каждого

элемента онтологии (например «класс») с определенным типом связей (например «подкласс класса»).

В настоящий момент OwlToDotConverter поддерживает конвертацию OWL-документов в .dot файл для следующих типов связей между элементами онтологий:

- 1) тип связей для «классы» – подкласс класса (subClassOf);
- 2) тип связей для «свойства» – подсвойство свойства (subPropertyOf);
- 3) тип связей для «индивиды» – связь индивида с классом (rdf:type).

Для описания принципа построения .dot файла в качестве узлов графа рассмотрим элементы онтологии типа «класс», в качестве ребер – тип связи подкласс класса – «subClassOf».

В результате обработки OWL-документов содержание .dot файла представляется следующим образом:

```
Fowl -- DarkMeatFowl [label = "SubClassOf"]
Fowl -- LightMeatFowl [label = "SubClassOf"]
Meat -- NonRedMeat [label = "SubClassOf"]
Meat -- RedMeat [label = "SubClassOf"]
RedMeat -- SpicyRedMeat [label = "SubClassOf"]
RedMeat -- NonSpicyRedMeat [label = "SubClassOf"]
vin_Wine [label = "Wine"]
food_Wine [label = "Wine"]
.....
```

Результат матричной визуализации .dot файла с помощью программного продукта MatrixExplorer приведен на рис. 2.

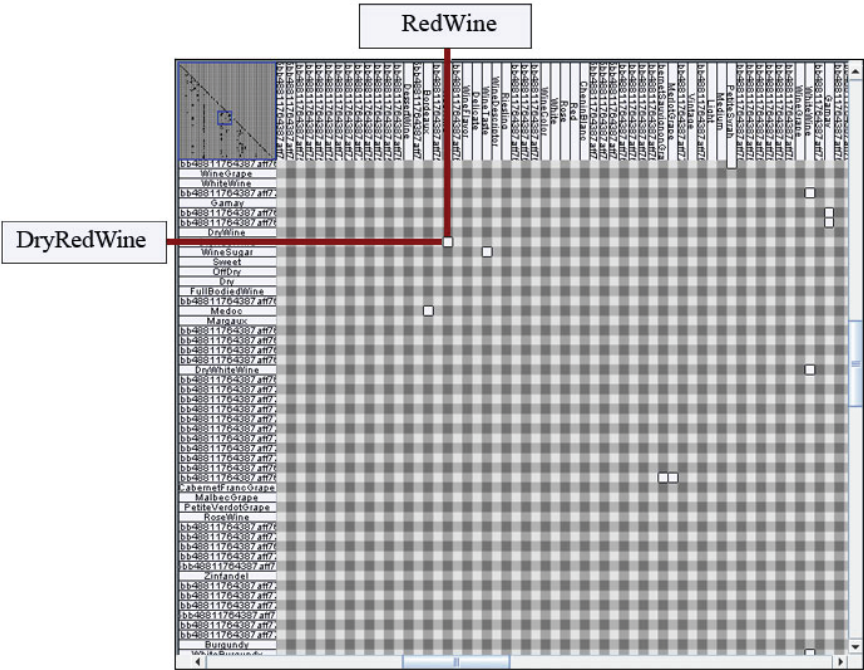


Рис. 2. Представление OWL-документа в виде матрицы смежности

OWL-граф представляется в виде разреженной матрицы смежности (см. рис. 2). При этом строки (столбцы) матрицы – узлы OWL-графа или, иными словами, элементы онтологии. В данном случае строки (столбцы) матрицы представляют собой перечень элементов «классы» связанных типом связи SubClassOf. Так, класс «DryRedWine» является подклассом класса «RedWine».

Видно, что полученное матричное представление OWL-графа, характеризующегося большим количеством узлов, не дает полной информации о структуре OWL-документа, что усложняет его анализ, так как ненулевые элементы матрицы хаотично распределены по всей матричной сетке (см. рис. 2). Решение данной задачи возможно при переупорядочивании ненулевых элементов, то есть выявлении кластерной структуры данных.

Кластеризация OWL-матрицы

Существует большое количество алгоритмов кластерного анализа, что обусловлено разнообразием постановок задач, критериев, определяющих качество производимой кластеризации для того или иного набора данных, а также базовых принципов, заложенных в самих алгоритмах.

Большинство алгоритмов кластеризации предполагают сравнение объектов между собой на основе меры близости (сходства). Мерой сходства может быть любой параметр, в том числе и длина – евклидово расстояние, расстояние городских кварталов (манхэттенское расстояние), расстояние Чебышева, степенное расстояние и т.п. [4].

Кластеризация OWL-графа, представленного в виде матрицы смежности, подразумевает перестановку строк матрицы, а если она не является симметричной, то и столбцов, с целью группировки более схожих элементов.

В данной статье задача кластеризации OWL-графа моделируется в виде задачи о коммивояжере Traveling Salesmen Problem (TSP). Другими словами, для заданного графа $G = (V, E)$, где V – множество вершин графа $|V| = n$; E – множество ребер графа, каждому из ребер $e_{ij} = (v_i, v_j) \in E$, $v_i, v_j \in V$, которого сопоставляется неотрицательная целочисленная стоимость C_{ij} , необходимо найти тур коммивояжера минимальной стоимости.

Для придания веса (стоимости) более отдаленным объектам, в качестве функции стоимости c_{ij} между двумя вершинами, рассмотрим метрику – квадрат евклидова расстояния, определяемую по формуле

$$C_{ij} = C(v_i, v_j) = \sum_{m=1}^k (v_{im} - v_{jm})^2, \quad (1)$$

где $v_i, v_j \in V$; v_{im}, v_{jm} – величины m -й координаты в k -мерном пространстве.

Задача о коммивояжере TSP является NP-полной. Нахождение алгоритма, дающего точное решение TSP за полиномиальное время, является трудноразрешимой задачей. Вместо поиска быстрого алгоритма целесооб-

разно заняться разработкой приближенного алгоритма с полиномиальным временем выполнения. В [1] доказано, что для функции стоимости C_{ij} , удовлетворяющей неравенству треугольника, существует приближенный алгоритм с полиномиальным временем работы, позволяющим решить задачу о коммивояжере.

Таким образом, определение тура коммивояжера T сводится к задаче минимизации стоимости тура

$$C(T) = \min \sum_{v_i, v_j \in V} C_{ij}.$$

Метод решения TSP существенно зависит от полноты матрицы расстояний, то есть существования ребра между любыми двумя вершинами хотя бы в одном направлении

$$e_{ij} = (v_i, v_j) \notin E \Rightarrow e_{ji} = (v_j, v_i) \in E.$$

Полнота матрицы расстояний, в свою очередь, зависит от элементов онтологии, представляющих узлы и типов связей между ними, представляющих ребра. Введем следующее ограничение: какой бы ни был тип связи между элементами онтологии, связь между узлами является двусторонней. Визуализация онтологии, в таком случае, ведет к построению неориентированного OWL-графа, для которого существует полная матрица расстояний, где расстояние определяется количеством ребер между любыми двумя вершинами.

Описание метода

Дан OWL-граф, представленный в виде матрицы смежности $A = [a_{ij}]$, где

$$a_{ij} = \begin{cases} 1, & \text{если существует ребро, связывающее вершину } v_i \text{ и вершину } v_j; \\ 0, & \text{в противном случае.} \end{cases}$$

$v_i, v_j \in V$. Пусть вес каждого ребра, связывающего две вершины $c_{ij} = c_{ji} = 1$.

Метод кластеризации матрицы смежности, представляющей OWL-граф $G = (V, E)$, заключается в отыскании порядка размещения строк $p = \langle v_i, v_j, \dots, v_n \rangle$ и столбцов $g = \langle v^i, v^j, \dots, v^n \rangle$ по признаку схожести. Эта задача эквивалентна отысканию тура коммивояжера в равносильной TSP и выделению из него кратчайшего пути, проходящего через все вершины графа не более одного раза.

Предлагаемый алгоритм состоит из следующих этапов:

1 этап. На основе заданного OWL-графа $G = (V, E)$ вычисляем матрицу $B = [b_{ij}]$ кратчайших расстояний (Shortest Path) между вершинами (объектами) v_i, v_j . Здесь длина пути представляется как сумма стоимостей ребер, составляющих путь. Иными словами задача Shortest Path состоит в

поиске пути между парой вершин, в котором содержится наименьшее количество ребер. Вычисление матрицы кратчайших расстояний реализовано алгоритмом Дейкстры (Dijkstra).

2 этап. Вычисление матрицы расстояний $D=[d_{ij}]$ между строками матрицы $B=[b_{ij}]$. Иными словами, среди всех строк матрицы $B=[b_{ij}]$ найдем наиболее похожие. В качестве меры «похожести» строк используется мера расстояния между объектами (1). Полученная матрица расстояний $D=[d_{ij}]$ соответствует графу $G'=(V',E')$.

3 этап. Определим TSP как задачу нахождения тура коммивояжера T' для графа $G'=(V',E')$ и матрицы расстояний $D=[d_{ij}]$. Заметим, что T' представляет собой цикл: обход по туру совершается с посещением каждой вершины ровно по одному разу и завершается в той же вершине, из которого он начался.

Выделение из полученного тура T' кратчайшего пути $p'=\langle v'_i, v'_j, \dots, v'_n \rangle$ возможно путем внедрения дополнительной вершины в графе $v'_{n+1} : c_{in+1} = \text{const } C$, для $\forall i = \overline{1, n}$. Иными словами, расстояние от вершины v'_{n+1} до каждой вершины графа $G'=(V',E')$ равно некоторой постоянной величине.

Решая TSP для расширенного таким образом графа $G'=(V',E')$, кратчайший путь $p'=\langle v'_i, v'_j, \dots, v'_n \rangle$ определим следующим образом: поиск кратчайшего пути в туре T'' , содержащем «дополнительную» вершину начинается с первой вершины, следующей за v'_{n+1} , а заканчивается вершиной, расположенной перед v'_{n+1} . Заметим, что $C(p') = C(T'') - 2\text{const } C$.

Полученный кратчайший путь $p'=\langle v'_i, v'_j, \dots, v'_n \rangle$, касательно нашей задачи кластеризации, представляет собой искомый порядок $p=\langle v_i, v_j, \dots, v_n \rangle$ размещения строк в матрице смежности $A=[a_{ij}]$.

4 этап. Аналогично 2-3 этапу, 4 этап подразумевает вычисление матрицы расстояний $F=[f_{ij}]$ между столбцами матрицы $B=[b_{ij}]$ с целью нахождения кратчайшего пути $g'=\langle v'^i, v'^j, \dots, v'^n \rangle$, однозначно определяющий искомый порядок размещения столбцов $g=\langle v^i, v^j, \dots, v^n \rangle$. Заметим, что если $A=[a_{ij}]$ – симметричная, то $p = g$.

Ниже проиллюстрирован результат работы вышеописанного алгоритма кластеризации с помощью MatrixExplorer (рис. 3).

Результатом использования вышеописанного алгоритма является один кластер с упорядоченным набором данных. Следует заметить, что OWL-документ, как результат описания некоторой модели данных, может представляться в виде не одной, а нескольких несвязанных между собой совокупностей объектов (графов). Таким образом, в случае применения алгоритма кластеризации для каждого такого графа отдельно, можно улучшить визуальное представление OWL-данных и, тем самым, упростить их анализ, а также существенно сократить время работы алгоритма.

связей между данными. Предложены пути его усовершенствования – сокращение времени работы алгоритма за счет выделения несвязанных графов в структуре OWL-документа и улучшение визуального представления посредством выделения из матрицы k -кластеров.

Результаты приведенного алгоритма кластеризации OWL-матрицы напрямую зависят от выбранного эвристического метода решения задачи о коммивояжере, которые, в свою очередь, сильно зависят от начальных данных и требуют индивидуального подхода к каждой поставленной задаче. В связи с этим, остался открытым вопрос об анализе алгоритмов решения задачи о коммивояжере для каждого типа связи между элементами онтологии с целью получения более точных результатов предложенного алгоритма кластеризации.

В данной работе рассмотрен один из возможных способов визуализации OWL-данных – матричный. Полученные результаты могут послужить основой для создания более совершенных алгоритмов визуализации. Так, совмещение матричного представления OWL-данных и представления с помощью графа смогут значительно улучшить человеческое восприятие содержимого OWL-документов. Построение трехмерной (кубической) матрицы позволит рассмотреть более сложные связи между элементами онтологии.

Список литературы

1. Алгоритмы, построение и анализ : пер. с англ / Т. Кормен [и др.]. – 2-е изд. – М. : Вильямс, 2005. – 1168 с.
2. Henry, N. MatrixExplorer: a Dual-Representation System to Explore Social Networks / N. Henry, J.-D. Fekete // IEEE Transactions on visualization and computer graphics. – 2006. – Vol. 12, № 5. – P. 677–684.
3. Pietriga, E. Semantic web data visualization with graph style sheets / E. Pietriga // In Proceedings of the 2006 ACM Symposium on Software Visualization, Brighton, United Kingdom, 4–5 September, 2006 / New York. – P. 177–178.
4. Sokal, R.R. Distance as a measure of Taxonomic Similarity / R.R. Sokal // Syst. zool. – 1961. – Vol. 10, № 1. – P. 70–79.
5. Языки описания и средства визуализации графов [Электронный ресурс] / А. Качур. – Режим доступа : <http://shcherbak.net/yazyki-opisaniya-isredstva-vizualizacii-grafov>. – Загл. с экрана.
6. OWL Web Ontology Language Guide [Электронный ресурс]. W3C Recommendation. 10 February 2004. – Режим доступа : <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>. – Загл. с экрана.
7. Resource Description Framework (RDF) : Concepts and Abstract Syntax. [Электронный ресурс]. W3C Recommendation. 10 February 2004. – Режим доступа : <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210>. – Загл. с экрана.

Visualization of Data Semantic Models

G.O. Legostaev

St. Petersburg State University, St. Petersburg

Key words and phrases: cluster; graph; matrix; Traveling Salesmen Problem; Semantic Web; Web Ontology Language.

Abstract: The technology of matrix representation of semantic data expressed in OWL syntax is developed; the algorithm of clustering the produced matrix representation is given; the ways of improvement of visual representations of OWL documents are proposed.

© Г.О. Леростаев, 2010