University Of New Brunswick

Semantic Web Techniques (CS6795)

Project Report

On

Schema.org Combined with Rules for The Semantic Annotation of HTML5 Pages

Submitted By

Manisha Gupta

Sharath Chandra Maddur

Swapnil Sule

Instructor

Dr. Harold Boley

Fall 2012

December 17, 2012

Table of Contents

List of Figures

Chapter 1

Introduction

When the search engines were first introduced, they highly relied on the content of the webpages to filter the search results. Search engines had no way of expressing the content of the pages, which resulted in very little success rate, relatively, pertaining to the search keywords. In other words, there wasn't much semantics used in developing the webpages. This created a need to introduce semantics to the webpages in a way that search engines understand the content of the pages and thus refine the results based on it. This consequently led to the invention of RDF (Resource Description Framework). RDF changed the way web pages are understood by search engines. They introduced new semantic angle to web designing. However, RDF is complicated. HTML5 Microdata was introduced to overcome this issue. Microdata is quite straightforward and feels much more native to HTML than RDF. It uses vocabulary and itemtypes defined by central agencies. The central agencies that provide vocabularies for Microdata are: Data-vocabulary.Org and Schema.Org. This collection of Vocabularies contains a wide variety of classes and properties that can be used while designing the web pages so as to give the content a semantic underpinning. Microdata make the search results more precise, better fitting the requirements of the users.

HTML5 is a markup language for structuring and presenting content for the World Wide Web (WWW) and a core technology of the Internet. It is the fifth revision of the HTML standard (created in 1990 and standardized as HTML4 as of 1997) and, as of December 2012, is still under development. Its core aims have been to improve the language with support for the latest multimedia while keeping it easily readable by humans and consistently understood by computers and devices (web browsers, parsers, etc.). HTML5 is intended to subsume not only HTML4, but XHTML 1.0 and DOM Level 2 HTML as well.

Features and APIs:

- **Microdata**
- **Web Apps**
- **WebSocket Protocol**
- **WebRTC**
- **W3C Web Media Text Tracks – WebVTT**

Of all the features of HTML5, Microdata stands out for its capability to add semantics to the items on a webpage.

Chapter 2

Objectives

1. Employing the subClassOf definitions of the RDFS-formatted Schema.org Type Hierarchy in OO jDREW.
2. Defining rules in RuleML using schema.org classes for typing rule variables.
3. Defining further RuleML 1.0 Rules and Facts.
4. Testing the rulebase using the OO jDREW 1.0 Querying system.
5. Designing a website using HTML5 Microdata using classes from schema.org.
6. Creating classes that are not existing in schema.org and using them for the website.
7. Testing the Microdata/Rich Snippets of the website using the Google Rich Snippet Tool.

Chapter 3

Approach

The authors used OO jDREW querying system to query the RuleML queries using the RuleML Rulebase.

3.1 OO jDREW Taxonomy Querying System

OO jDREW is written in the Java programming language.
We query in two different ways in OO jDREW

- Bottom-Up execution is used to infer all derivable knowledge from a set of clauses (forward reasoning)
- Top-Down execution is used to solve a query on the knowledge base (backward reasoning)

It supports two languages:

POSL: POsitional Slotted Language that implements RuleML shorthand (positional part similar to Prolog).

RuleML: OO jDREW supports the XML syntax of RuleML.

There are few steps to use Taxonomy Querying System using OO jDREW

Step1 - Input RDFS in the Type Definition of OO jDREW 1.0
Select RDFS as the input format and select Load type information.

Step2 - Enter the rules in the RuleML format
Select RuleML as input format and select parse Knowledge Base.

Step3 – Once you have parsed the Knowledge Base, query that Knowledge Base with desired RuleML queries.

Select the input format as RuleML and click on Issue Query.

3.2 RULEBASE

Schema.RDFS.Org is an initiative launched on June 2, 2011 by Bing, Google, and yahoo. It was launched so that a common set of schemas can be created. Schema.RDFS.org provides the Schema.org Type Hierarchy in various formats including RDF/XML, RDF/Turtle, RDF/Triples and JSON.

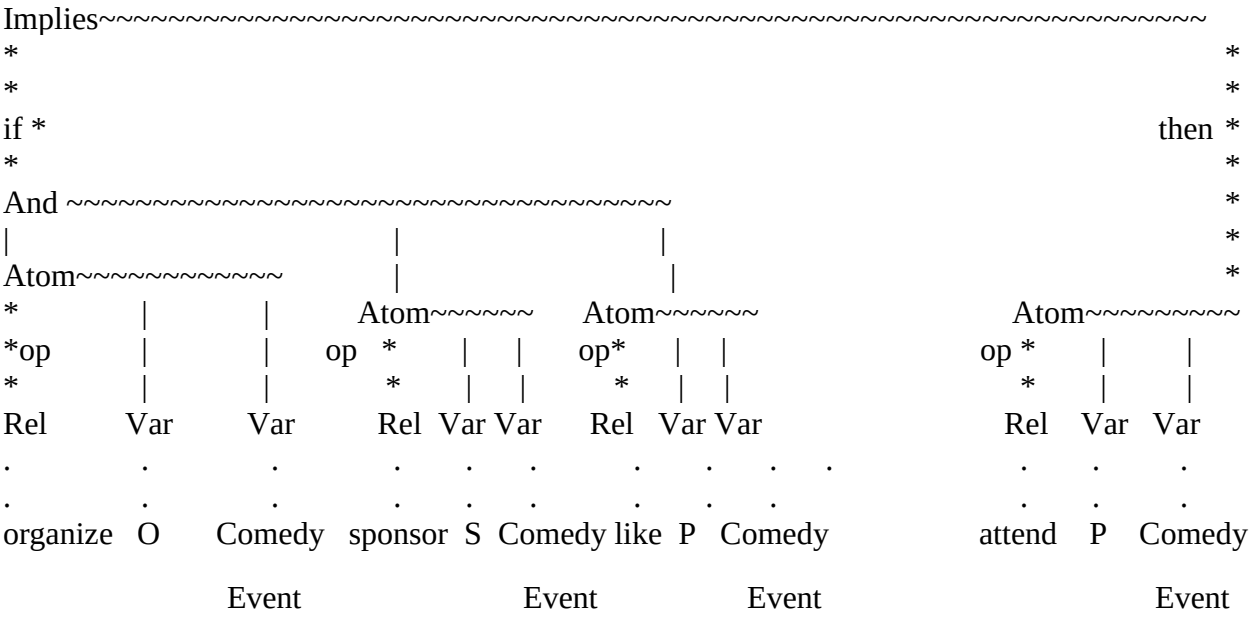
Many sites are generated from structured data, which is often stored in databases. When this data is formatted into HTML, it becomes very difficult to recover the original structured data. Many applications, especially search engines, can benefit greatly from direct access to this structured data.

Over the years, Rule MarkUp Language (RuleML) has become a Rule Modeling Language and rule Meta-Logic. It is a unifying family of XML-serialized rule languages spanning across all industrially relevant kinds of web rules. It has also become an over-arching specification of Web rules. It has already accommodated and extended other rule languages, building inter-operation bridges between them.

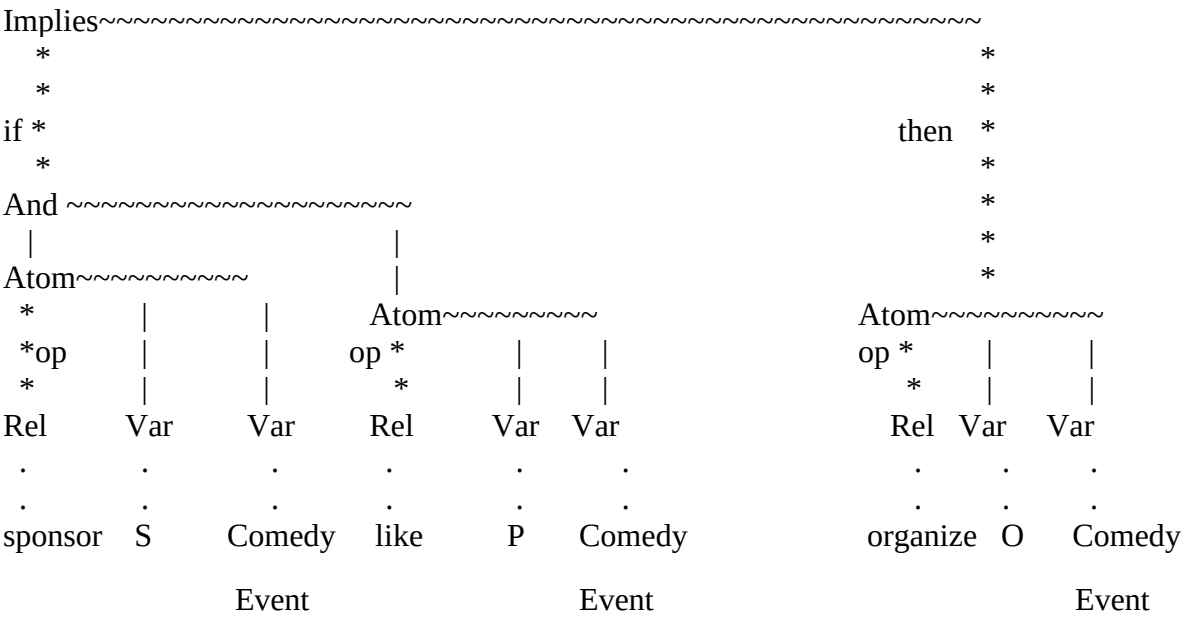
In this project, the authors made rules using some facts in RuleML. These rules correspond to the following events which indeed are some of the subclasses of 'Event' in the taxonomy of schema.org.

1. ComedyEvent
2. MusicEvent
3. DanceEvent
4. PoojaEvent
5. CakeCuttingEvent

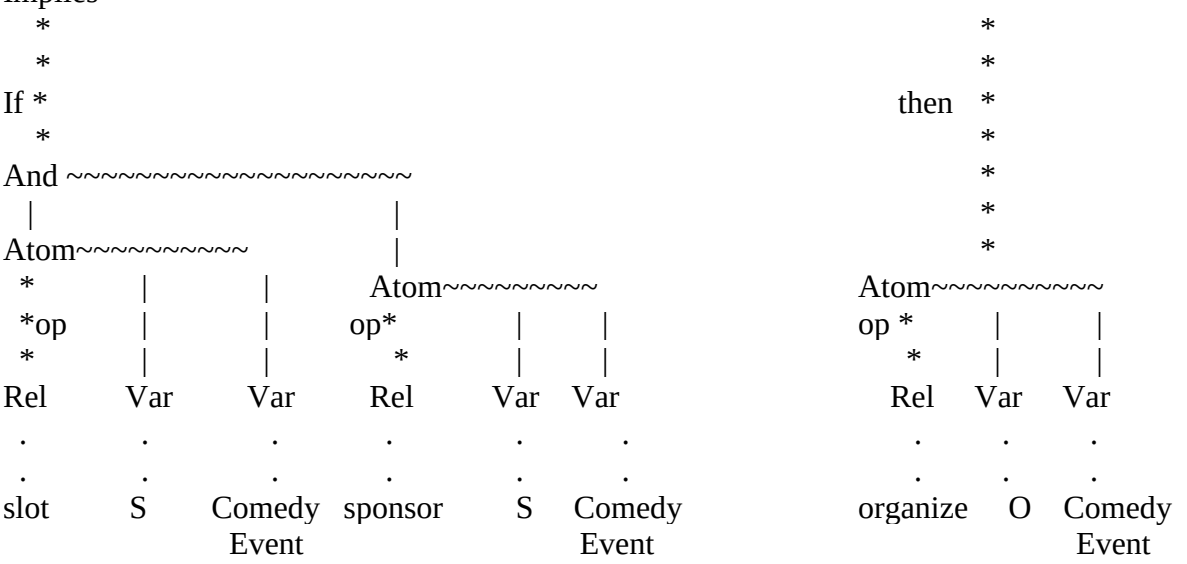
Example of OrdLab Tree:



Rule 1: Comedy Event



Rule 2: Comedy Event



Rule 3: Comedy Event

“Slot” is a username which means “availability” or “vacancy” of ComedyEvent.

Similarly, one can make the OrdLab tree for other four events with specific rules.

For vertical association and horizontal association with schema.org classes, We defined the following rules using as types to constrain rule variables, which can be explained as follows in test cases.

3.3 Test Cases

1. ComedyEvent:

Rule 1: The first rule implies that a person of type ‘Person’ which is a class in schema.org attends a Comedy Event of type ‘ComedyEvent’ if an organizer which is a type of ‘Organization’ organizes a ComedyEvent, sponsor which is of type ‘Organization’ sponsors a ComedyEvent and the person likes Comedy Event. Similarly,

Rule 2: The second rule implies that an organizer organizes a Comedy Event if a sponsor sponsors a ComedyEvent and a person likes Comedy Event.

Rule 3: The third rule implies that an organizer organizes a ComedyEvent if the organizer has a slot for a ComedyEvent and sponsor sponsors ComedyEvent.

Rule 4: The fourth rule is the set of facts.

Test Result:

To test the rules, one can use the following query:

```
<Atom>
        <Rel>attend</Rel>
        <Var type="Person">P</Var>
        <Var type="ComedyEvent">CE</Var>
</Atom>
```

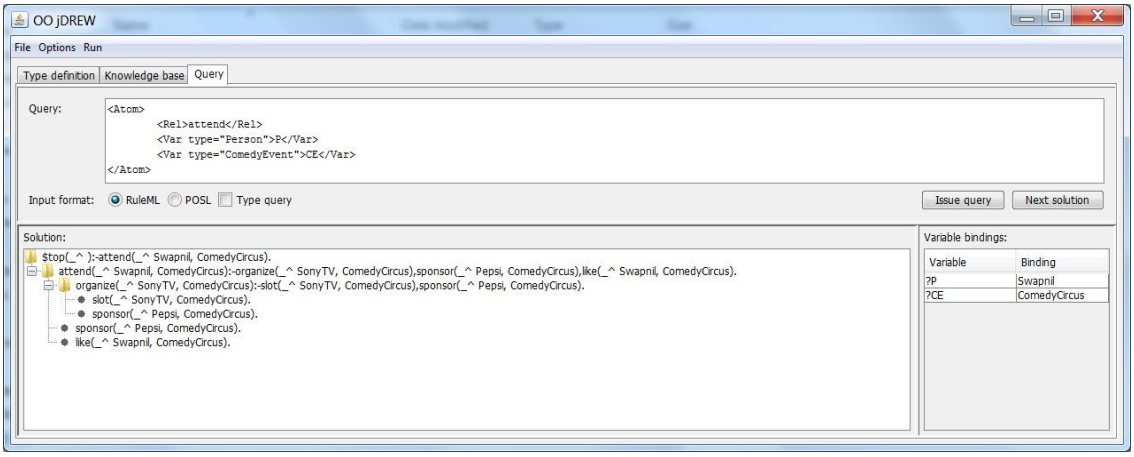


Fig 1: Snapshot of Querying OO jDREW for ComedyEvent

2. MusicEvent:

Rule 1: The first rule implies that a person performs in a Music Event if organizer organizes a Music Event, Sponsor sponsors Music Event and person is available for the Music Event.

Rule 2: The second rule implies that an organizer organizes a music Event if the organizer has a slot for a Music Event and sponsor sponsors Music Event.

Rule 3: Third rule is the set of facts.

Test Result:

To test the rules, one can use the following query:

```
<Atom>
        <Rel>perform</Rel>
        <Var type="Person">P</Var>
        <Var type="MusicEvent">ME</Var>
</Atom>
```

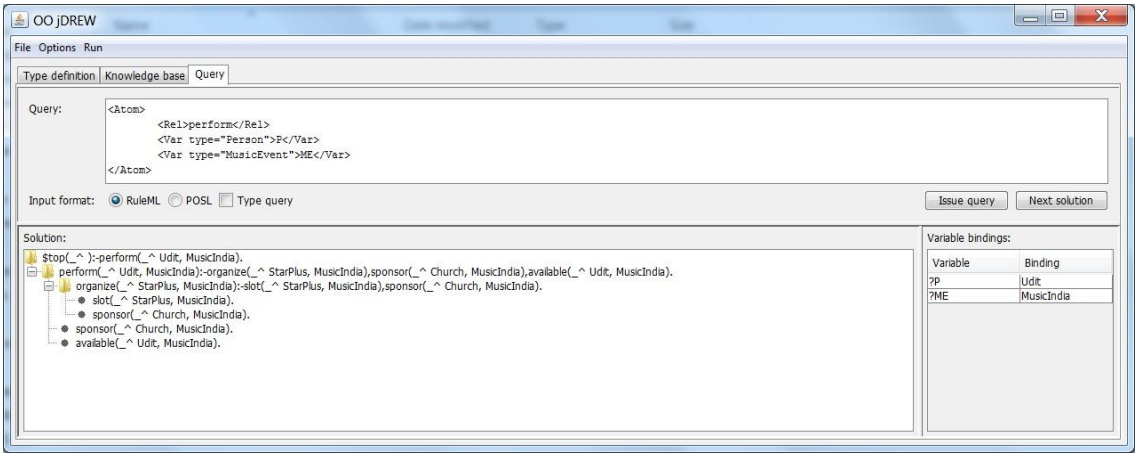


Fig 2: Snapshot of Querying OO jDREW for MusicEvent

3. DanceEvent:

Rule 1: The first rule implies that a person performs in a Dance Event if the Judge Judges a Dance Event, there is a Dance Form for a Dance Event and person likes that Dance Form.

Rule 2: The second rule implies that judges judge a Dance Event if Organizer organizes a Dance Event and Organizer assigns a judge to the Dance Event.

Rule 3: Third rule is a set of facts.

Test Result:

To test the rules, one can use the following query:

```
<Atom>
        <Rel>perform</Rel>
        <Var type="Person">P</Var>
        <Var type="DanceEvent">DE</Var>
</Atom>
```

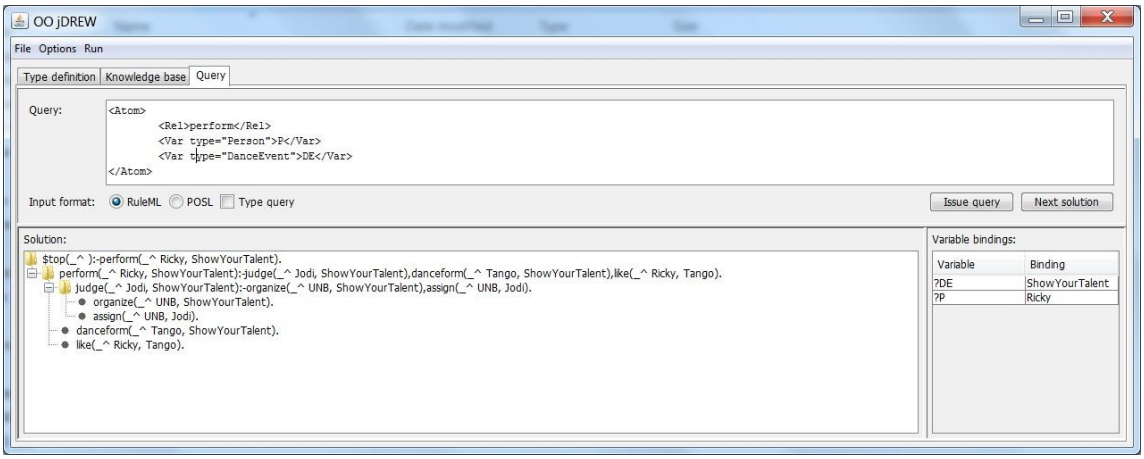


Fig 3: Snapshot of Querying OO jDREW for DanceEvent

4. PoojaEvent:

Rule 1: The first rule implies that a person attends a Pooja Event if organizer organizes a Pooja Event, Sponsor sponsors Pooja Event and person has faith in Pooja Event.

Rule 2: The second rule implies that an organizer organizes a Pooja Event if the organizer assigns a Priest to Pooja Event and sponsor sponsors Pooja Event.

Rule 3: The third rule is a fact.

Test Result:

To test the rules, one can use the following query:

```
<Atom>
        <Rel>attend</Rel>
        <Var type="Person">P</Var>
        <Var type="PoojaEvent">PE</Var>
</Atom>
```

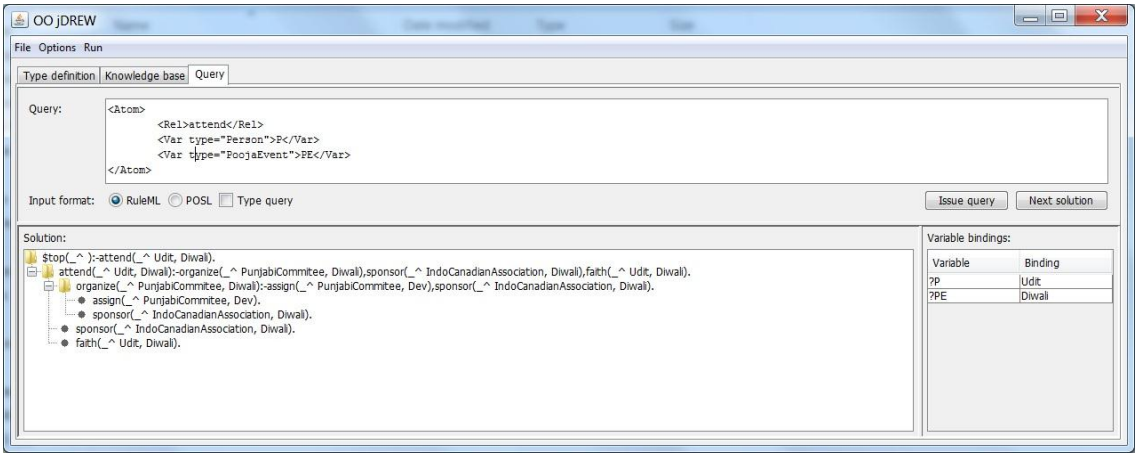


Fig 4: Snapshot of Querying OO jDREW for PoojaEvent

5. CakeCuttingEvent:

Rule 1: The first rule implies that a person attends a Cake Cutting Event if an organizer organizes the Cake Cutting Event, sponsor sponsors Cake Cutting Event and a person likes it.

Rule 2: The second rule implies that an organizer organizes Cake Cutting Event if a sponsor sponsors cake cutting event and celebrity inaugurates the cake cutting event.

Rule 3: The third rule implies that a sponsor sponsors a cake cutting event if a sponsor has funds and organizer gets profit from it.

Rule 4: Fourth rule is a set of facts.

Test Result:

To test the rules, one can use the following query:

```
<Atom>
        <Rel>attend</Rel>
        <Var type="Person">P</Var>
        <Var type="CakeCuttingEvent">CCE</Var>
</Atom>
```

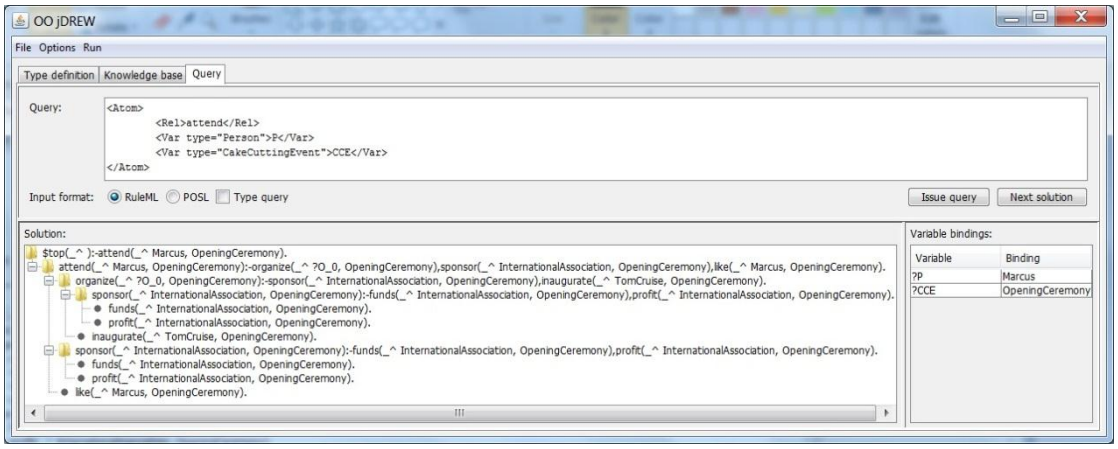


Fig 5: Snapshot of Querying OO jDREW for CakeCuttingEvent

6. Vertical Association:

If we look at the taxonomy of schema.org, we can see that Event, Person and Organization are subsumed by Thing while ComedyEvent, MusicEvent, DanceEvent, PoojaEvent and CakeCuttingEvent are subsumed by Event.

Test Result:

To test the rules, one can use the following query:

```
<Atom>
        <Rel>subsumes</Rel>
        <Var>X</Var>
        <Var>Y</Var>
</Atom>
```

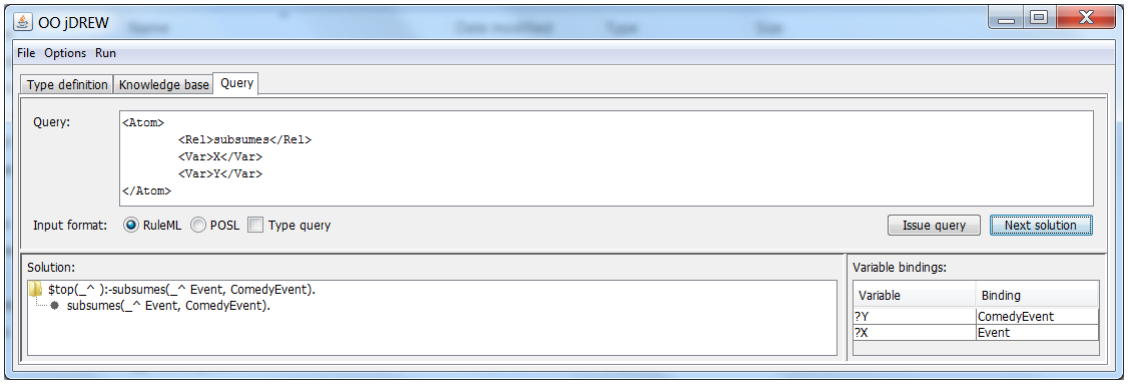


Fig 6: Snapshot of Querying OO jDREW for Vertical Association

3.4 Building the Web-Site

Microdata is an HTML specification used to nest semantics within existing content on web pages. Search engines, web crawlers, and browsers can extract and process Microdata from a web page and use it to provide a richer browsing experience for users. Search engines benefit greatly from direct access to this structured data because it allows search engines to understand the information on web pages and provide more relevant results to users. Microdata uses a supporting vocabulary to describe an item and name-value pairs to assign values to its properties. Microdata helps technologies such as search engines and web crawlers better understand what information is contained in a web page, providing better search results. Microdata is an attempt to provide a simpler way of annotating HTML elements with machine-readable tags than the similar approaches of using RDFa and Microformats.

A collection of available vocabularies are:

1. DataVocabulary.org

2. Schema.org

While both the websites purpose is to provide web developers with vocabularies to add semantics to the items on their web pages, Data-vocabulary.org is behind Schema.org in the available number of classes. Schema.Org provides many more and wide varieties of classes for the web developers to choose from to suit their requirements. For this reason, Schema.Org is preferred to Data-Vocabulary.org. Moreover, Data-vocabulary has been integrated into Schema.org making latter the standard for Microdata. However, Data-vocabulary.org can still be used to refer to the vocabularies in Microdata. For these reasons, we chose Schema.Org vocabulary to use with Microdata.

3.5 Planning of the Website

Authors came up with an idea that, instead of choosing few HTML pages with Microdata, they would make their own website containing HTML5 Microdata using Schema.org so that it gives them more authority and expressivity on the pages while letting us experiment more with the different schemas. Thus they have decided to make a website on behalf of Indian Student Association at University of New Brunswick. This website represents ISA (Indian Student Association) and it has information about its whereabouts, purpose, goals, and activities. Most importantly, ISA is celebrating Diwali, an Indian festival and is organising various events on this occasion. Students and any other page visitors can browse through the website, learn about the events, and even register for few of them if they're interested. Each of the pages are built using Microdata so that the search engines refine the results in an optimized fashion, which will give better chances for the users who are unfamiliar with the website to bump into it when they search on internet for events related to Diwali and UNB. The aim of building this website is to show how search engines and web developers can work in collaboration using HTML5 Microdata to provide the users with more accurate and better search results.

3.6 Implementation of the Website

The authors have built the website in four phases. Below is the detailed description on what different phases comprised of.

3.6.1 Phase 1: Writing HTML Code

Just as the case with any other website, writing code is the first and foremost thing we had to do to make the webpages. We went with the traditional way of writing the code i.e. writing HTML code in notepad++ for each of the pages. At this point, we had not yet tried to include Microdata. Instead we focused on the external appearance of the website. We also tried to keep the website as simple as possible for two reasons; 1. The primary goal of the website is to showcase the functionalities of HTML5 Microdata and benefits of using it in the webpages. 2. Avoiding usage of heavy Cascade style sheets would conserve the bandwidth and save us time which could be efficiently spent on the actual goal. For this reason, we decided to spend fewer resources on rotating icons and fading links and more on functionalities. This approach has rendered the website appearance simple but the purpose is still served.

We divided the website into 6 pages, main page being the home page, and then the rest being the subpages. Homepage consists of the basic information about the website and its location, and latest news on current affairs. We made sure this page has some information about Diwali celebrations and then we linked this piece to an entirely new page called Events which has few sub categories for a convenient browsing. The sub categories are namely Pooja Event, Dance Event, Music Event, Comedy Event and Cake Cutting Event. Each of these webpages consists of information about relevant event with a basic description about it and its timing. Few events which require attendants to pay the cover charge have it mentioned on their respective pages.

3.6.2 Phase 2: Applying Microdata

After writing the code and designing the content of the website, we carefully went through each text item to find those that can be applied with Microdata. To decide which text item fits the bill, all the classes and properties of Schema.Org had to be studied thoroughly. Sometimes we encountered a situation where we had to add more content to the webpage to make the usage of Microdata significant enough. This proved to be tricky because it is was difficult to keep adding the information for the sake of using Microdata on it while keeping the meaning of the page intact. But it had to be done to emphasize on the role and importance of Microdata. We successfully used microdata on almost all the possible items. We also experimented with adding microdata content in different tags and surprisingly, it worked on all the tags. We are yet to find an HTML tag that doesn't support Microdata in it. At this point, it might be safe to say all HTML tags are compatible with Microdata.

Example of Microdata used in Div Tag:

```
<div itemscope itemtype="http://schema.org/person">
<span itemprop="name">Anil Kumar,<br /></span>
<span itemprop="jobTitle">President, <br /></span>
<span itemprop="workLocation">780, Windsor Street,<br />
ISA Building,<br />
Fredericton, NB<br />
E3B 5A3<br /></span>
<span itemprop="telephone">(506) 321654<br /></span>
<span itemprop="email">anil@unb.ca<br /></span>
<br /></div>
```

The above piece of Microdata code is an extract from the home page. It gives semantics to the text it is applied to. The piece of text "Anil Kumar" in the second line is a fictitious name of the President of ISA. The "Itemscope" tells the browser that the following piece of code contains Microdata. Itemprop="name" tells the browser the property of the text Anil Kumar as being name. Thus it adds semantics to the content. Now the browser knows what the text actually refers to.

3.6.3 Phase 3: Putting Schema.org to use

Schema.org provides vocabulary for Microdata. It has wide variety of vocabularies and properties to fit almost any requirement but not completely. There are few situations where a certain suitable vocabulary is not found and a new one is to be made. Schema.org provides a mechanism to make entire new classes and attributes. This got our interest and we decided to test and experiment with these new class creations. Hence we decided to make new classes on Schema.org besides using the existing ones. Below is the taxonomy for the classes we've used from Schema.org in making the website.

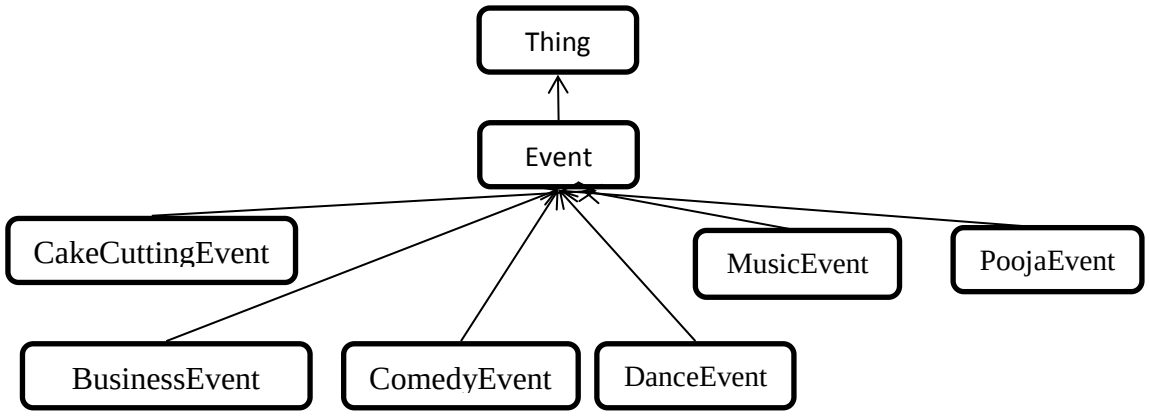


Fig 7: Taxonomy Diagram for Schema.Org

The classes MusicEvent, BusinessEvent, ComedyEvent, and DanceEvent are the existing events directly inherited from Schema.org while the other two, namely PoojaEvent and CakeCuttingEvent are new classes defined by us. Also additionally, we went on to defining new properties for these classes.

Below is an extracted structured data using Google Rich Snippet for the Pooja Event page of the website.

Fig 8: Structured data for PoojaEvent from Google Rich Snippet Tool

The extracted structured data from Google snippet page shows the details of the new class we've linked to Schema.org. Its attributes are currently non-existing on schema.Org but we've submitted this sub-class of event to them explaining the need for making this addition to their vocabulary given its importance in Hindu culture. They haven't responded yet but if they do, we plan on extending the properties to few more to make it more significant.

Another new class defined by us is CakeCuttingEvent. This event basically describes about the properties of the cake. This one, though may not seem as significant as the other user defined class, it still helped us understand the inside mechanics in a better way.

Fig 9: Structured data for CakeCuttingEvent from Google Rich Snippet Tool

The properties are self-explanatory for most part except startdate. The first part of it refers to the date of the event while the second, following the separator T refers to the time of the Event.

3.6.4 Phase 4: Bringing the Website Live

This was perhaps the most simplest of all the phases, yet tricky. We had to find webhosting sites that offer hosting services without interfering with the content of the page with their advertisements. Most of the webhosts gave us hard time letting us use our own layouts. They allowed us to use their predefined layouts which turned out to be a problem because with the limited predefined space on a page, we were forced to cut down the information, which would cost us some items with microdata which is the whole point of the website. After a long search, we came across Page.tl which gave us permission to make websites with our own layout but the ad content was relatively more on it. One way to get rid of these ads is to install adblocker on the browser. Thus the website went live and we've tested all the Microdata and rich snippet codes using Google Rich Snippet Tool. While it is working on the rich snippets, it is yet not showing in the search engine page. We wrote to Google about it and they responded it takes around 6-7 weeks for the site to show on the search results.

Chapter 4

Conclusion

1. We queried the Schema.org Rules and Facts, in which we employed the subClassOf definition of the RDFS formatted type hierarchy in OO jDREW for vertical association.
2. We defined RuleML 1.0 Facts and Rules.
3. We also developed a website containing HTML5 Microdata to demonstrate how Schema.org works for the optimized search results.

Chapter 5

Future Implementation

1. We plan on extending the scope of the website using more of the rules and facts created in Rulebase.
2. We plan on making more user defined classes on Schema.Org and use them in the website.
3. We also plan on submitting the new classes to Schema.Org as a proposal to integrate them with the existing classes.

Chapter 6

References

[1] Schema.org (Type Hierarchy):
http://schema.org/docs/full.html

[2] Schema.RDFS.org:
http://schema.rdfs.org/

[3] OO jDREW (Taxonomy Querying System)
http://www.jdrew.org/oojdrew/docs/TaxonomyQuerying.html

[4] http://ruleml.org/1.0/#Appendix-1

[5] Team 5 of Fall 2011
http://extoojdrew.weebly.com/index.html

[6] Microdata HTML: http://en.wikipedia.org/wiki/Microdata_(HTML)

[7] RDFS: http://en.wikipedia.org/wiki/RDFS