RuleTheWeb!: Rule-based Adaptive User Experience [⋆]

Adrian Giurca[1 2], Matthias Tylkowski[2] and Martin Müller[2]

[1] Dept. of Databases and Information Technology,
[2] Binary Park
Brandenburg University of Technology,
P.O. 101344, 03013 Cottbus, Germany

Abstract. During the last years the business rules industry proliferated as rules were recognized as a practical tool for solving real-world problems. Nowadays, many research communities develop rule languages and rule systems as well as rule markup languages and interoperability tools. However, due to the necessary high level of knowledge and complexity of tools, rules are yet designed only inside of narrow and high-skilled communities. After more than a decade of research on Semantic Web, and after initiatives of the main industry players, the Web is fast evolving into a world of objects as content creators started enriching their pages with semantic annotations. This paper presents an application using simple rules to enrich the user navigation experience on the web. We show a demo of adaptive user experience based on semantic data and reaction rules aiming to enable social rules designed and shared by web users.

1 Background

Five years ago, in a blog posting [12], Ora Lassila was pointing out that Semantic Web may not be only about data but also there is significant work to do with respect of *"systems that work on users' behalf"*:

> For a long time (longer than I have worked on the Semantic Web) I have wanted to build *systems that work on users' behalf.* Semantic Web is one of the enabling technologies, a means to an end, and not the end itself. Every time I look critically at the current use of (information) technology, I cannot help but wonder how it is possible to actually get away with the approach taken today (where substantial burden is placed on the users).

The Semantic Web community developed an amazing set of knowledge representation languages such as Resource Description Framework (RDF) (See [23] for a hub of resources), and Web Ontology Language (OWL)[22], query languages such as SPARQL [26], and thousand of well established tools. However,

[⋆] This research is supported by ESF-EXIST grant No. 03EGSBB066, CatchThis: Conjoint-Analyse in Sozialen Netzwerken

most of the applications were centered on creating and querying Linked Data, i.e., to connect related data that wasn't previously linked using URIs and RDF. There is a little publicly available work with respect of building Semantic Web applications which use business intelligence to connect various web documents according with user preferences.

1.1 Application Vocabularies

Ontology experts developed a large amount of web vocabularies such as FOAF [18], DOAP [17] GoodRelations [19] just to mention some of them. There are many projects aiming to process large amount of semantic data (big data projects). Recently, initiatives such as Web Data Commons[3] published extracted semantic data from several billion web pages[4].

However, one of the main difficulties to use this data comes from the large number of vocabularies that are involved, as SPARQL queries must be aware of vocabularies. Along with the Facebook Open graph Initiative `https://developers.facebook.com/docs/opengraph/`, in June 2011, Google, Bing and Yahoo! launched a common initiative, `http://schema.org` towards *a unique web vocabulary* to be used in semantic annotations:

> A shared markup vocabulary makes easier for webmasters to decide on a markup schema and get the maximum benefit for their efforts. So, in the spirit of sitemaps.org, search engines have come together to provide a shared collection of schemas that webmasters can use.

Initiatives such as `http://getSchema.org` already report large amount of web sites using this vocabulary. We expect that, due to the increasing revenues of the content creators when using Schema.org annotations, this vocabulary will spread very fast on the Web content. Therefore our application focuses on this vocabulary although only little change would be needed to support other vocabularies too. Recently Microsoft and others announced submission to standardization of the Open Data Protocol `http://www.odata.org/`.

1.2 Business Rules

Some of our previous work (see [6]) reported on rule-based processing of semantic data annotations of HTML pages by considering annotation languages such as RDFaLite [15]. We emphasized that using rules one can significantly enrich the user interaction experiences on the Web. In addition, by offering new information in ways not originally planned, such application contributes to creation of linked data too. Specifically business rules can be successful involved when it comes to capture user's interaction with web pages towards running various business processes such as:

[3] `http://webdatacommons.org/`

[4] Notice that the extracted data does not come in standard RDF as they use an RDF triple extension, N-Quads. Basically they augment the RDF triple with another component which is the URI from where the triple was extracted

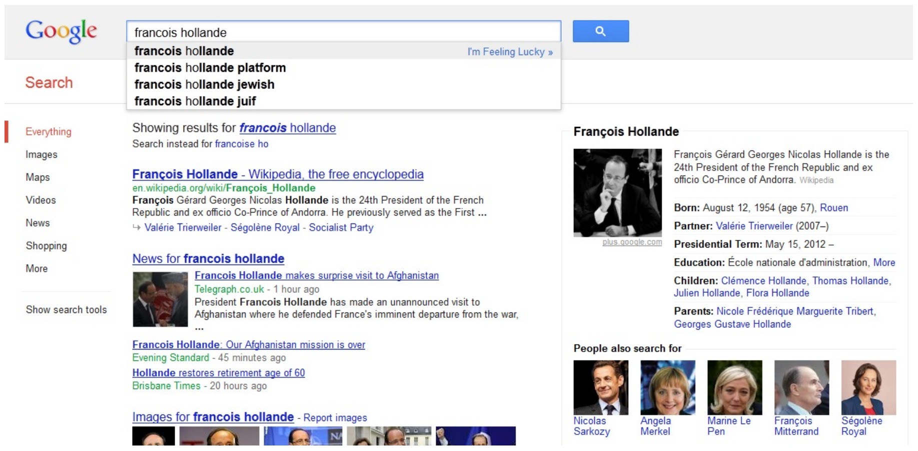
– Developing groups of navigational items that are meaningful to users. This includes the development of the most sensible set of navigational menu items, e.g., *Whenever the user clicks more than 3 times a menu item add this item to the fast access menu items.*
– Giving concepts from a vocabulary (Schema.org) on a page, show related linked data, e.g. *If the user loads financial news, then offer him a three months subscription to Financial Times.*
– Showing concepts visually, e.g. *When the user loads specific news about weather forecast, then deliver him a map of related weather events at the location.*
– Allow user to add calendar events related to visited pages, e.g. *If the user loads a conference web site then ask him to add event to calendar and show him travel opportunities.*
– Allow user to annotate the page, e.g., *If the page is loaded more than 5 minutes then on close ask user to annotate the page with tags/ratings.*

It is easily to see that such kind of rules may also involve events that occurs in the web browser, therefore RuleTheWeb! application uses both production rules and reaction rules. The readers may notice that when the rules are based on a unique vocabulary they can be far *fast shared between various actors.*

2 The Challenge

Enriching user navigation experience is not a novel paradigm. Web publishers can use various available tools such as Outbrain[5] or Linkwithin[6], just to mention some, to embed related content in their web pages. However, this experience is related to the publishers and not the readers of the web content. These application **do not include user preferences** as they embed related content only by processing the web site content and, as they are commercial products, there is no information on the models and the technology they use to create related data. However there is also research work on similar web pages mostly featuring machine learning concepts and using similarity measures (metric-based, feature-based, or probabilistic). By contrary, RuleTheWeb! employs user preferences, Semantic Web annotations and behavioral targeting to create the best related content towards a semantic navigation on the web. In addition, because the semantic annotations are extracted on the fly and rules are always up to date there is no inconsistency between cached data (such as existent crawled summaries or raw data on the server side of other solutions) and the actual status. When content creators update their web sites and the user visit them, of course, RuleTheWeb! delivers **immediate and up to date related content**. RuleTheWeb! is enriching the reader experience by considering the semantic of the visited page and user's own preferences encoded as rules.

[5] `https://www.outbrain.com/`
[6] `http://www.linkwithin.com/`



francois hollande
francois hollande platform
francois hollande jewish
francois hollande juif

Showing results for *francois* hollande
Search instead for francoise ho

François Hollande - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**François_Hollande**
François Gérard Georges Nicolas **Hollande** is the 24th President of the French Republic and ex officio Co-Prince of Andorra. He previously served as the First ...
↳ Valérie Trierweiler - Ségolène Royal - Socialist Party

News for **francois hollande**
Francois Hollande makes surprise visit to Afghanistan
Telegraph.co.uk - 1 hour ago
President **Francois Hollande** has made an unannounced visit to Afghanistan where he defended France's imminent departure from the war, ...
Francois Hollande: Our Afghanistan mission is over
Evening Standard - 45 minutes ago
Hollande restores retirement age of 60
Brisbane Times - 20 hours ago

Images for **francois hollande** - Report images

François Hollande
François Gérard Georges Nicolas Hollande is the 24th President of the French Republic and ex officio Co-Prince of Andorra. Wikipedia
plus.google.com
Born: August 12, 1954 (age 57), Rouen
Partner: Valérie Trierweiler (2007–)
Presidential Term: May 15, 2012 –
Education: École nationale d'administration, More
Children: Clémence Hollande, Thomas Hollande, Julien Hollande, Flora Hollande
Parents: Nicole Frédérique Marguerite Tribert, Georges Gustave Hollande

People also search for
Nicolas Sarkozy
Angela Merkel
Marine Le Pen
François Mitterrand
Ségolène Royal

Fig. 1. Google Enriching Search Results

Big players such as Google already come up with enriching the search related content as depicted in Figure 1. This results may be related to Schema.org initiative or may not.

3 RuleTheWeb! - The Application

RuleTheWeb! is related to the W3C use cases of Linked Data Incubator [21] basically to the generic case of social recommendations[7] allowing users to benefit on the linked data recommendations with respect of the web sites they visit and the activities they perform. The related data to be offered is real-time computed by the application.

The actual implemented demo scenario included the second use case described in Section 1.2 i.e., when the current loaded page contains specific Schema.org product annotations[8], the application will suggest related product offers and related reviews, from various service providers.

Basically, when a user navigates the web using a browser employing RuleTheWeb!, they will receive recommendations as soon as the page information matches one of their rulesets. The rulesets are automatically loaded from the rule repository and the user is able to choose between various rulesets. The application uses two main categories of rulesets:

1. Rule-based user preferences. Rules are computed on top of userpreferences via logic-based conjoint analysis, [25], [7], [8], [9].
2. Social Web Rules users can create/generate/share rules. Social Web Rules forms an application field of social rules theory [4] being a basic form of

[7] `http://www.w3.org/2005/Incubator/lld/wiki/Use_Case_Social_Recommendations`
[8] See `http://schema.org/Product` and `http://getschema.org` for more examples

human interaction. Users are always motivated to (1) use publicly available rules meeting their goals - public rules are powerful because we tend to believe our friends before believing a marketing message from a brand. (2) create their own private rules and (3) share rules with the community. People like to share because (a) it brings valuable and entertaining content to others; (b) is a way of self definition; (c) is a source of growing their relationships in the community. A work in progress is a rulestore API allowing consumers to manage web rules.

3.1 The Rule Language

The rule repository stores RuleML, [24] rules while the rules in the secondary storage are JSON rules [5]. Developed in 2012, JSON rules version 2, uses document object model (DOM) event types [14] as underlining events vocabulary and a condition language build on the HTML5 DOM Core [10]. This version features five types of conditions:

1. *JavaScript Boolean conditions* - to capture any experience that can be induced by running JavaScript code in the browser as rule condition. For example, `document.getElementById('id').value=="container"` is a JavaScript Boolean condition evaluating `true` if the current document has an element with `id="container"`.
2. *Descriptions* - to offer a simple format to express conditions with respect of the current document structure. For example the description:

```
{
 "type":"input",
 "context":"$E",
 "constraints": [
   {
    "propertyName" : "id",
    "operator" : "EQ",
    "restriction" : { "type": "String", "value" : "
        postalCode"}
   },
    {
     "propertyName" : "nodeValue",
     "operator" : "MATCH",
     "restriction" :  { "type": "Regex", "value" : "/^\d
        {5}$/"}
    },
   {
    "bind" : "$V",
    "propertyName" : "nodeValue"
   }
  ]
}
```

will bound variable `$E` to the specific `input` element, if such element exists and its value encodes a postal code following a specific structure described by a regular expression (its value is bound to variable `$V`). The language keywords include names such as `tagName`, `nodeValue`, `id`, `class`, `about`, `property`, `vocab`, `typeof`, `itemscope`, `itemtype`, `itemprop` to address the corresponding DOM and HTML5 (including RDFa 1.1 Lite and Microdata) attributes.

3. *XPath conditions* - to offer fast access to any content of the current document. For example, `$X in html/table[1]/tr[2]` will bound the variable `$X` to a collection of table data, the second row of the first table in the current document.
4. *Equality.* The traditional equality between two logical terms.
5. *Built-in predicates.* Built-in predicates do not follow any specific schema, they are simple Boolean JavaScript expressions. Failure to evaluate such a JavaScript expression is interpreted as logical `false`.

JSON Rules actions are close to JavaScript function calls as such there is very much freedom on implementing both state change actions and environment change actions. This solution covers the RIF Production Rule Dialect [13] standard actions too:

– State change actions:
    1. We experience an *assert fact action* when create a new element/attribute
    2. A *retract fact action* when we delete a DOM element/attribute
    3. A *retract all slot values* when we delete specified all attributes of a DOM Element
    4. A *retract object* action when deleting an attribute of a DOM Element
– Environment change actions:
    1. An *execute* action when we run JavaScript code.

The reader may notice that while RuleML is a large family of rule languages allowing rules to be defined in top of any vocabulary, JSON rules are defined using a specific vocabulary based on Schema.org, the Facebook Open Graph and the Document Object Model (DOM). As DOM is an universal specification for all web pages the main benefit is that such rules can be immediately shared between users. However, JSON-Rules does not aim to offer standard actions (allowing for any JavaScript function call) and its actual implementation sticks to only a set of predefined possible actions.

This way we keep close to the approach of RDF rules [2] as well on some principles of publishing rules online [3]. The JSON rule model is depicted by Figure 2.

3.2 A Simple Scenario

When users like to enrich their experience on visiting web sites discussing movies they can use RuleTheWeb! and load a specific ruleset from the rules repository (either their private ruleset or a public ruleset). For example, such a ruleset may contain rules implementing cases like below:
Whene the visited page contains Schema.org Movie annotations, then

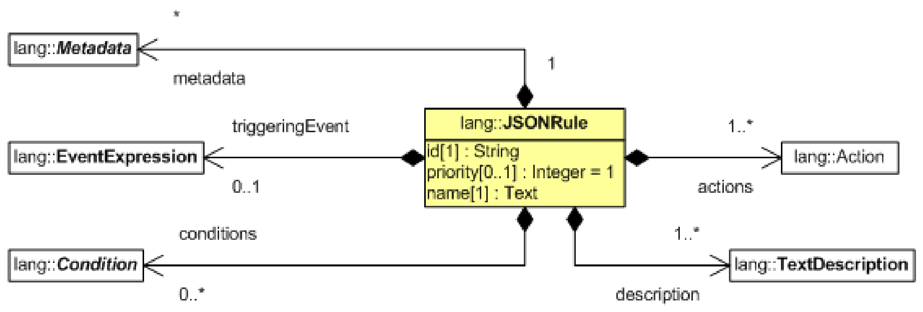


Fig. 2. RuleTheWeb: The Core Data Model

1. show related movies by the same genre.
2. show related films with the same genre and created in the same year.
3. show a trailer of the movie and background information. Also offer the sound-track, the latest news about the movie as well as statistical information.
4. show film suggestions from the same director.
5. show related film directed by the same director, in the same year.
6. show background information about the producer.
7. show background information about the music creator and offer related music composed by the same person.

The rule repository returns JSON Rules. The Example 1 shows a possible rule describing a part of the above scenario:

Example 1 (A JSON Rule).

```
{
  "id": 3,
  conditions: [
   {
    "type":"Element",
    "context":"$T",
    "constraints": [
   {
    "propertyName" : "itemscope",
    "operator" : "NEQ",
    "restriction" : { "type": "String", "value" : "null"}
   },
    {
     "propertyName" : "itemtype",
     "operator" : "EQ",
     "restriction" :  { "type": "URL", "value" : "http://
         schema.org/Movie"}
    },

   {
    "type":"Element",
    "context":"$_",
    "constraints": [
   {
    "propertyName" : "itemprop",
    "operator" : "EQ",
    "restriction" : { "type": "String", "value" : "name"}
   },
    {
     "propertyName" : "parentNode",
     "operator" : "EQ",
     "restriction" :  "$T"
    },
    {
    "bind" : "$Y",
    "propertyName" : "nodeValue"
   }
  ],
  "actions": [
    "invoke('youtube', $Y +' trailer')",
    "invoke('imdb', $Y )",
    "invoke('amazon', $Y + ' soundtrack')",
    "invoke('googlenews', $Y)",
    "invoke('wolframalhpa', $Y)"
  ]
}
```

Action `invoke` is an environment change action (included in the standard execute action of RIF-PRD). The conditions of the rule are related to the DOM content and, as usual, must be satisfied to execute the intended actions.

The actions are invoked sequentially but the final result, including possible state change i.e. changes into the current DOM (the working memory), will take place at the end of all action calls and the environmental effect may be a *sum* of all actions to be executed.

When elements annotated with `http://schema.org/Movie` (bound to `$T`) have a child annotated with the property `name` (`$Y` is bound to the text node ”Francis Ford Coppola”) then the conditions are satisfied. For example when the DOM contains the below fragment all rule conditions are satisfied:

```
...
<div itemscope itemtype="http://schema.org/Movie">
...
  <span itemprop="name">Francis Ford Coppola</span>
...
</div>
...
```

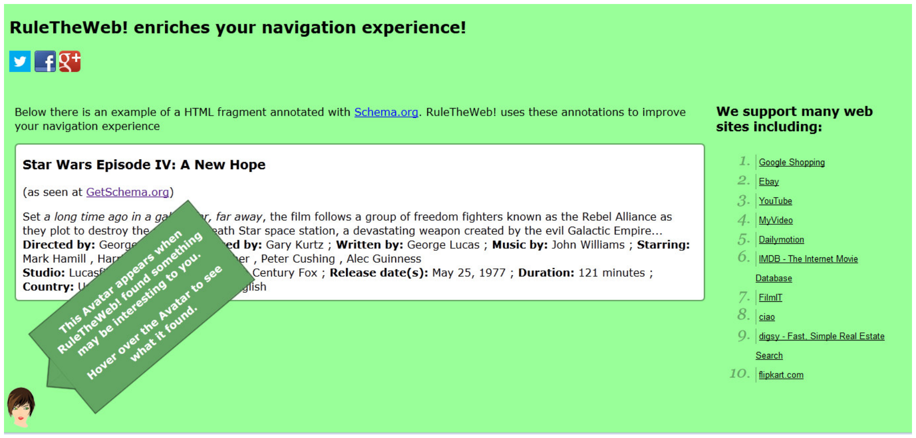
While the above example uses HTML5 Microdata annotations [16], JSON-Rules language also supports RDFa 1.1. Lite annotations [15]. This is quite simple, as RDFa 1.1. lite is very close to Microdata, basically by using `typeOf` instead of `itemtype` and `property` instead of `itemprop`.

3.3 On Rule Execution

RuleTheWeb! does not employ a rule engine. Rather, the rules are directly compiled to executable JavaScript code. Also, the way the actions are executed is part of the compiling technology which is in development.

3.4 How to use it

To install the application please visit `https://ruletheweb.org`. After a successful installation one must see an explanation page offering a brief introduction on how to use the extension.



RuleTheWeb! enriches your navigation experience!

Below there is an example of a HTML fragment annotated with Schema.org. RuleTheWeb! uses these annotations to improve your navigation experience

Star Wars Episode IV: A New Hope

(as seen at GetSchema.org)

Set *a long time ago in a ga*[illegible]*r, far away*, the film follows a group of freedom fighters known as the Rebel Alliance as they plot to destroy the [illegible]eath Star space station, a devastating weapon created by the evil Galactic Empire...
Directed by: George [illegible]**ed by:** Gary Kurtz ; **Written by:** George Lucas ; **Music by:** John Williams ; **Starring:** Mark Hamill , Har[illegible]er , Peter Cushing , Alec Guinness
Studio: Lucasf[illegible] Century Fox ; **Release date(s):** May 25, 1977 ; **Duration:** 121 minutes ;
Country: U[illegible]glish

This Avatar appears when RuleTheWeb! found something may be interesting to you.
Hover over the Avatar to see what it found.

We support many web sites including:

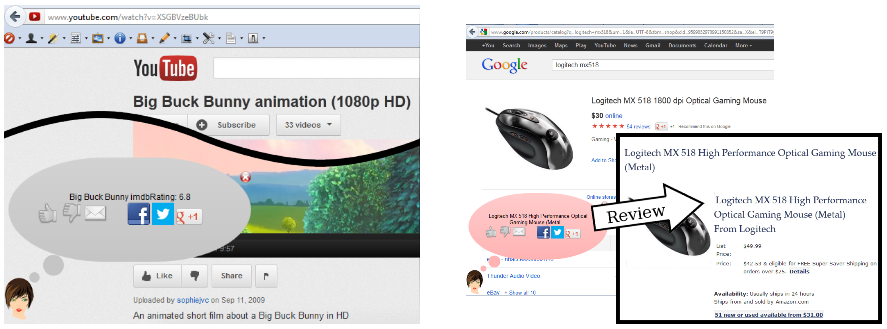
1. Google Shopping
2. Ebay
3. YouTube
4. MyVideo
5. Dailymotion
6. IMDB - The Internet Movie Database
7. FilmIT
8. ciao
9. digsy - Fast, Simple Real Estate Search
10. flipkart.com

Fig. 3. RuleTheWeb! introduction page

The Figure 4 illustrates the application by showing activities on web sites such as Google Shopping and YouTube. When watching a movie or a trailer on YouTube one can receive additional information of that movie from imdb.com. When navigating on Google Shopping and search for a desired product, RuleTheWeb! offers additional information like reviews or specifications.

3.5 Architecture

The application architecture is depicted by Figure 5. All rulesets are loaded by the application from a rule repository based on RuleML serializations.


Big Buck Bunny animation (1080p HD)
Big Buck Bunny imdbRating: 6.8
Logitech MX 518 1800 dpi Optical Gaming Mouse
Review
Logitech MX 518 High Performance Optical Gaming Mouse (Metal)
Logitech MX 518 High Performance Optical Gaming Mouse (Metal) From Logitech
Fig. 4. Related news on YouTube pages and helping on buying products

The Client The client components are depicted by Figure 5.

Browser current window. The current DOM, containing semantic annotations, acts as a facts provider: the semantic data is extracted and these are the facts to be matched with rule conditions.

RuleTheWeb. It compiles rules to JavaScript code and execute them under usual conditions. The rule execution result is sent to the Service Layer towards executing the actions. The execution result is used by the Formatter module to create the desired presentation.

Secondary Storage. The secondary storage combines two different kinds of rules: Shared rules from the rule repository and private rules that the user created himself. The user can decide to upload his custom created rules to the rule repository to publicly share them with other users.

The server side The server side has two main components: (a) A rule repository infrastructure with the main role of serving user-defined rulesets and (b) a service processor with the main role to process rule actions.

3.6 Notes on Implementation

RuleTheWeb! is implemented as a standard client-server application invoking a server service from a web based application (the client) under the same specific session. A sequence diagram, depicted by Figure 6, describes the basic execution process. The client-server communication is Ajax based.

The `Ruleset` object is serialized to the browser secondary storage. As usual, a ruleset is a collection of rules designed to fulfill a goal.

The `PageData` object implements the logic of rules working memory, i.e. it manages the facts on which the rules are matched.

The `Log` object implements the logic of actions to be executed. Basically, when a ruleset is active, the Log object stores all actions effects to be performed. Notice that the actions will not be executed immediately when they are fired by some rule but at the end of the entire ruleset execution. Therefore this object will also deal with the order of action effects.

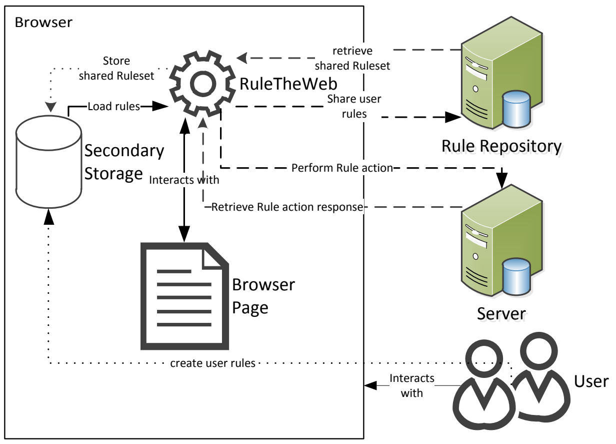


Fig. 5. RuleTheWeb! Architecture

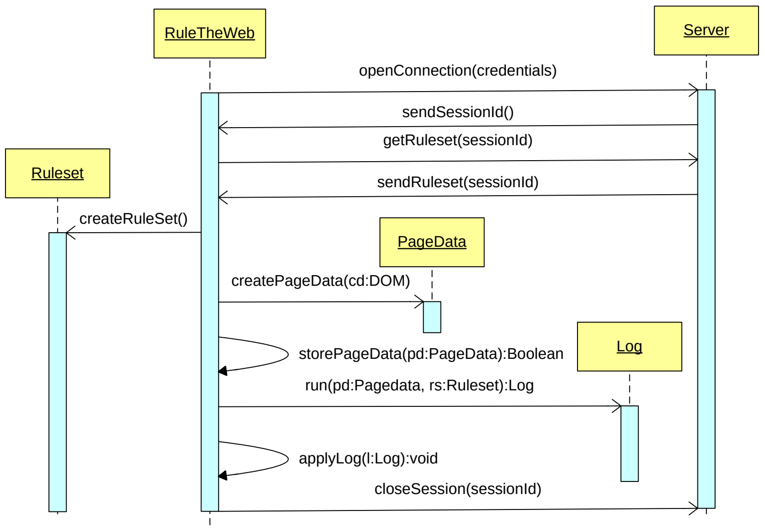


Fig. 6. RuleTheWeb: Basic RuleSet Execution

The RuleTheWeb! Firefox demo uses the storage only available for extensions[9]. In addition, there are two other kinds of storages that the application is using: (a) The session storage[10], used to store the state of the application (disabled or enabled). This storage remains valid over the browser session and settings are restored when the browser is restarted, and (b) the DOM Storage[11], persistent as long as the actor stays on the same page.

4 Conclusion and Future Work

This demo as a proof-of-concept gave insight how rules can be used together with semantic annotated content on web pages to enrich the user web surfing experience. There is an ongoing work to define a complete data model of capturing user preferences by investigating the capabilities of data collection offered by modern communication tools such as online media and social networking. Such model aims to capture most of widely accepted preference properties with respect to behavioral economics concepts [11] such as heuristic, i.e. consumers make decisions based on approximate rules and not strict logic (see also [20] for an interesting use case).

Because this application uses rulesets created by third parties according with the user profile they store, future work will offer users to change and store their own rules. While writing rules typically requires professional expertise, our goal is to allow users to write simple rules while experts may contribute to complex rules as well as to rule curation.

In addition, for the content creators, the application will be offered as a standalone JavaScript application to be added to the web pages whereas a browser extension with respect of these pages will no longer be necessary.

Acknowledgements

We would like to gratefully acknowledge Prof. Daniel Baier, head of department of Marketing and Innovation and Prof. Ingo Schmitt, head of department of Databases and Information Technology for their useful insights.

References

1. Y. Aytar, M. Shah, J. Luo. Utilizing Semantic Word Similarity Measures for Video Retrieval, IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA.
2. T. Berners-Lee, D. Connolly, E. Prud'homeaux, Y. Scharf. Experience with N3 rules, W3C Rules language Workshop April 2005, `http://www.w3.org/2004/12/rules-ws/paper/94/` .

[9] `https://developer.mozilla.org/en/Storage`
[10] `https://developer.mozilla.org/en/Session_store_API`
[11] `https://developer.mozilla.org/en/DOM/Storage`

3. H. Boley: Are Your Rules Online? Four Web Rule Essentials. Advances in Rule Interchange and Applications, International Symposium, RuleML 2007, Orlando, Florida, October 25-26, 2007, pp. 7-24, http://www.cs.unb.ca/~boley/papers/RuleEssentials.pdf.
4. T.R. Burns, and T. Dietz (1992) Cultural Evolution: Social Rule Systems, Selection, and Human Agency. International Sociology 7:250-283.
5. A. Giurca and E. Pascalau (2008). JSON Rules. In G. J. Nalepa and J. Baumeister (Eds.) Proceedings of 4th Knowledge Engineering and Software Engineering, KESE 2008, collocated with KI 2008, Spetember 23, 2008, Kaiserlautern, Germany, CEUR Workshop Proceedings Vol 425.
6. A. Giurca, E. Pascalau (2009). Building Intelligent Mashups. Tutorial at 32nd Annual Conference on Artificial Intelligence (KI 2009), September 15-18, 2009, Paderborn, Germany, https://docs.google.com/View?id=dcff8ncf_181gxb3ss65.
7. A. Giurca, I. Schmitt, and D. Baier. Performing Conjoint Analysis within a Logic-based Framework. Proc of IEEE Federated Conference on Computer Science and Information Systems, (FedCSIS2011), Szczecin, Poland, 18-21 September, 2011.
8. A. Giurca, I. Schmitt, and D. Baier.Can Adaptive Conjoint Analysis perform in a Preference Logic Framework? The 8th workshop on Knowledge Engineering and Software Engineering (KESE8) at the ECAI 2012 Montpellier, France, August 27-31, 2012.
9. A. Giurca, I. Schmitt, and D. Baier. Adaptive Conjoint Analysis. Training Data: Knowledge or Beliefs? A Logical Perspective of Preferences as Beliefs. Proc of IEEE Federated Conference on Computer Science and Information Systems, (FedCSIS2012), Wroclaw, Poland, 9 - 12 September, 2012.
10. A. Le Hors, P. Le Hgaret, L. Wood, G. Nicol, J. Robie, M. Champion, S. Byrne. Document Object Model (DOM) Level 3 Core Specification, Version 1.0, W3C Recommendation 07 April 2004, http://www.w3.org/TR/DOM-Level-3-Core/.
11. D. Kahneman, and A. Tversky (1979). Prospect theory: An analysis of decisions under risk. Econometrica 47 (2): 263-291.
12. O. Lassila. Semantic Web Soul Searching, Blog posting , March 19, 2007. http://www.lassila.org/blog/archive/2007/03/semantic_web_so_1.html last retrieved: June 10, 2012.
13. C. de Sainte Marie, G. Hallmark, A. Paschke. RIF Production Rule Dialect, W3C Recommendation 22 June 2010, http://www.w3.org/TR/rif-prd/.
14. D. Schepers, J. Rossi, B. Höhrmann,P. Le Hgaret, T. Pixley. Document Object Model (DOM) Level 3 Events Specification, W3C Working Draft 31 May 2011, http://www.w3.org/TR/DOM-Level-3-Events/ .
15. Manu Sporny. RDFa 1.1. Lite, W3C Candidate Recommendation, http://www.w3.org/TR/rdfa-lite/
16. * * * HTML5. A vocabulary and associated APIs for HTML and XHTML, http://www.whatwg.org/specs/web-apps/current-work/multipage/microdata.html. last retrieved: June 20, 2012.
17. * * * DOAP: Description of a Project, https://github.com/edumbill/doap/wiki last retrieved: June 10, 2012.
18. * * ** Friend Of A Friend, http://semanticweb.org/wiki/FOAF, last retrieved: June 20, 2012.
19. Good Relations: a Web vocabulary for e-commerce, http://www.goodrelations-vocabulary.org/, last retrieved: June 20, 2012.
20. L. Lee, S. Frederick and D. Ariely (2006), Try It, Youll Like It: The Influence of Expectation, Consumption, and Revelation on Preferences for Beer. Psychological Science. Vol. 17, No. 12: 10541058.

21. Library Linked Data Incubator Group: Use Cases. `http://www.w3.org/2005/Incubator/lld/wiki/UseCases`, last retrieved: June 20, 2012.
22. * * * Ontology Web Language, W3C, `http://www.w3.org/OWL/`, last retrieved: June 10, 2012.
23. * * * Resource Description Framework, W3C, `http://www.w3.org/RDF/`, last retrieved: June 10, 2012.
24. * * * The RuleML Initiative, `http://ruleml.org`, last retrieved: June 10, 2012.
25. I. Schmitt, and D. Baier. Logic Based Conjoint Analysis using the Commuting Quantum Query Language, Proc. of Conference of the German Classification Society (GfKl2011), August 31 to September 2, 2011, Frankfurt am Main, Germany.
26. * * * SPARQL 1.1 Query Language, W3C Working Draft 05 January 2012, `http://www.w3.org/TR/sparql11-query/`, last retrieved: June 15, 2012.