

# SPARQL SAHA, a Configurable Linked Data Editor and Browser as a Service

Eetu Mäkelä and Eero Hyvönen

Semantic Computing Research Group (SeCo),  
Aalto University and University of Helsinki, Finland  
first.last@aalto.fi, <http://www.seco.tkk.fi/>

**Abstract.** SPARQL SAHA is a linked data editor and browser that can be used as a service, targeting any available SPARQL endpoint. Besides being available as a web service, the primary differentiating features of the tool are its configurability to match the underlying data, and the fact that the usability of its user interface has been verified by dozens of non-experts using the tool in multiple multi-year projects.

## 1 Introduction

As part of the Linked Data ecosystem, both linked data browsers as well as editors are needed. The Semantic Web Wiki alone lists 45 distinct entries in its browser subcategories<sup>1</sup>, and 35 tools in its editor category<sup>2</sup>. Among such a large number of tools, it would seem hard for a new tool to distinguish itself. However, after surveying the competition it was deduced that the SPARQL SAHA linked data editor and browser presented in this paper still has a particular combination of features that contributes to others' work. The tool is available online.<sup>3</sup>

First, SPARQL SAHA operates *as a service*<sup>4</sup>, being able to be pointed against any SPARQL-compliant endpoint for both querying and updating. Further, the usability of the tool has been verified by dozens of non-experts using the tool in multiple multi-year projects. Finally, the tool is also fully multilingual.

Originally, SAHA was a dedicated application with its own storage [1]. The tool was created as a generic web-based RDF metadata editor that would 1) be easily adaptable to various schemas, 2) support multiple simultaneous users, and 3) be understandable and usable by laymen. It was only later as part of a general push toward more modular applications [2] that the tool was modified to work on top of a standard SPARQL endpoint, a major topic of this paper. In doing so, it was discovered that this also increased the usability and adaptability of

---

<sup>1</sup> [http://www.w3.org/2001/sw/wiki/Category:Special\\_Browser](http://www.w3.org/2001/sw/wiki/Category:Special_Browser)

<sup>2</sup> <http://www.w3.org/2001/sw/wiki/Category:Editor>

<sup>3</sup> This work is part of the Linked Data Finland project funded by the Finnish Funding Agency for Innovation (Tekes) of 20 partners.

<sup>4</sup> online at <http://demo.seco.tkk.fi/ssaha/>, with sources available at <http://code.google.com/p/saha/>

the tool, as now there was an additional, configurable layer of SPARQL queries mapping the raw data into its various presentation forms.

To evaluate the usability of SAHA, it has been used in a number of projects [3,4,5] by non-experts in RDF or Linked Data. The longest term and broadest of these is the BookSampo project [5], where dozens of volunteer librarians from Finnish public libraries have been indexing Finnish fiction literature using the editor since 2008. The user interface highlights presented here come mostly from the experiences gained in that project.

## 2 The User Interface of the SAHA Metadata Editor

Functionally, the main user interface of SAHA is divided into four distinct views. The first of these is the *index view*, which shows an overview of the class hierarchy used in the project along with corresponding instance counts. Clicking on any of the classes brings up the second *instance view*, which shows instances of the class under question. Together, these two views allow indexers and browsers to get an overview of the contents of the dataset.

In the BookSampo project, the instance view has also been used as simple work lists, with librarians dividing work for example alphabetically inside the instance list. This sort of ad-hoc collaboration is supported in the editor environment by a project-wide chat functionality, visible on all pages on the top right of the interface. Problems arising from multiple persons editing a resource at the same time are averted by a locking mechanism.

For quick navigation to resources of interest, all views in SAHA include a common keyword search functionality in the top part of the user interface. Typically, this search is configured to target all properties of all resources, so that for example indexers can quickly search for “author Waltari” to find books authored by Mika Waltari instead of the person record itself.

Selecting an instance moves the UI into the third view, the *browsing view*, where properties of the selected resource are visualized, along with a list showing any other resources referring to it. Clicking on any of the resources listed loads the view with that resource, enabling associative browsing of the dataset.

However, to better support contextualization and quick browsing, in all interface components, all resources also show their direct properties in a pop-up preview when the mouse is moved over them. Feedback from library indexers has shown this to be an extremely important functionality for them.

From the browsing view, the user can finally enter the *editing view*, which mirrors the browsing view with editable components, as show in Figure 1. The user is here editing metadata about Mika Waltari’s historical novel *Sinuhe the Egyptian*. The view shows both all properties that the resource has, as well as all properties associated with classes the resource is an instance of.

For object references, SAHA utilizes semantic autocompletion [6]. When the user tries to find a concept, SAHA uses at the same time web services to fetch concepts from configured external repositories, as well as the local project. Results are shown in one autocompletion result list regardless of origin, and their

Novels: Sinuhe the Egyptian

[\[view\]](#) | [\[rdf\]](#) | [\[config\]](#) | [\[ren\]](#)

**Luxor (place)**  
→name: Luxor

results from <http://demo>

Isla Luxor  
Ku-Luxomo  
Kwa Luxomo  
Luxorfjellet  
administrative region: M  
→Markaz Luxor  
inhabited place: Luxomni

**hierarchy** Westmoreland county | Pennsylvania | United States | North and Central America | World

**IsPartOf** [Westmoreland](#)

**name** Luxor

**TGN ID** 2090395

**type** [inhabited place](#)

**wgs84 latitude** 40.3333

**wgs84 longitude** -79.4667

**name**

**creator**  
päävastuullinen tekijä

**character in the narrative**  
select reference from Kaunokki-ontology

**concrete place**  
concrete places from the real world; select reference from Geo-ontology

**time of events**  
general expressions of time; select reference from Kaunokki-ontology

inhabited place: Luxor  
inhabited place: Luxor  
inhabited place: Luxora  
peak: Luxor Peak  
physical feature: Luxor

luxo

[\[remove\]](#) [Babylon](#) [\[edit\]](#)  
[\[remove\]](#) [Egypt](#) [\[edit\]](#)  
[\[remove\]](#) [Syria](#) [\[edit\]](#)

select reference (range unknown)

[\[remove\]](#) [ancient times](#) [\[edit\]](#)

[\[close\]](#)

**name** add new literal

[\[remove\]](#) (en) ancient times  
[\[remove\]](#) (sv) gamla tiden  
[\[remove\]](#) (fi) vanha aika

**Fig. 1.** The SAHA metadata editor, showing both semantic autocompletion as well as a pop-up preview presentation of one of the autocompletion results.

properties can also be inspected using the pop-up preview presentations. In the situation depicted in Figure 1, for example, this is extremely useful when the user must choose which of the many Luxors returned from both local and external RDF sources is the correct annotation for this book. The preview of places also includes a map visualization, but this is cut off in the figure. For geodata, SAHA also supports the editing of points, polylines, and polygons on maps.

For the purposes of the BookSampo project, the SAHA editor was improved with an inline editor feature. The idea is simple: a resource referenced through an object property can be edited inline in a small version of the editor inside the existing view. Specifically, this functionality was developed to ease the use of the auxiliary resources required in the RDF data model to group information together, such as when associating an issue number with information that a book has been published in a particular literature series. However, there seemed to be no reason to restrict the functionality, so this possibility is now available for all linked object resources. In Figure 1, this is shown for the property “time of events” whose value “ancient times” has been opened inline for editing.

### 3 Flexibility from Configurability and Using SPARQL

To be easily usable for a variety of projects, the SAHA editor provides a configuration interface with sensible defaults, with the configuration options also serializable as RDF and thus transferable along with the project data and schema.

For each property and class combination, the order in which the properties are displayed can be configured by simply dragging and dropping the fields. In addition, each object property can also be linked to external repositories from which to draw references from, while literal properties can be shown either with language-tagging support or without.

The move to using configurable SPARQL queries to populate all views considerably increased the flexibility of the editor. For example in the BookSampo project, certain metaclasses that caused confusion could be hidden from the class hierarchy. Also, while most projects wanted to group the list of resources referring to an object by the referring property, in BookSampo the librarians were so used to a pure alphabetical list that they wanted the functionality reverted.

The configurability of the SPARQL queries also came into use in tuning the scalability of the system when suddenly the BookSampo dataset grew to 1.5 times its former size as the project also started to cover children's literature. Here, response times on the now 6.6 million triple dataset decreased threefold by tuning the generic default queries to the exact data model semantics of the dataset. In the end, this resulted in performance faster than the original dedicated SAHA.

In the BookSampo project, operating on top of a SPARQL endpoint also allowed the creation of custom reports, basically SPARQL queries whose results are linked to the SAHA editor. For example, queries have been crafted to list multiple books having the same name and author as candidates for merging, as well as to list orphaned resources no longer referenced from any primary item.

### 4 SAHA as a Linked Data Browser

In addition to its use as an editor, SAHA can also be used as a simple linked data browser and publishing platform. That is, the resource view of SAHA supports content negotiation and is able to present either an RDF view or the human-oriented browser view in response. In this mode, the tool has been used to publish for example a World War I -related data collection [3]. Further, the tool was chosen as the linked data browser to be used for the national Linked Data Finland LDF.fi portal<sup>5</sup>, after experimenting with multiple other choices such as URIBurner<sup>6</sup> and Pubby<sup>7</sup>. Here, the primary reasons for selecting the tool were: 1) the pop-up preview functionality of the tool provided a better user experience than was available from competitors, 2) by configuring the SPARQL queries, the tool could be adapted to the sometimes quirky data modeling conventions used by some of the datasets to be published and 3) the tool supported linked data

<sup>5</sup> <http://www.ldf.fi/>

<sup>6</sup> <http://uriburner.com/>

<sup>7</sup> <http://wifo5-03.informatik.uni-mannheim.de/pubby/>

browsing of material also in cases where the IRIs did not match their publication location, also an unfortunate case in some of the datasets.

## 5 Related Work

As stated earlier, there are both many other editors as well as browsers for linked data. Based on the survey conducted for this work, of the editors the closest to SAHA would be SKOSjs<sup>8</sup>, which also works as a web application on top of SPARQL endpoints, and features multilanguage support. However, where SAHA is directed mostly at indexing instances, SKOSjs is geared towards managing a SKOS vocabulary hierarchy. Thus, the two tools actually complement each other.

Of browsers, the closest matches seemed to be Pubby<sup>9</sup> and Graphity [7], which can both operate on top of a SPARQL endpoint to publish data, and where Graphity also had ways by which to configure the presentation of that data. For both tools however the SPARQL endpoint is fixed in configuration, thus disallowing using the tool as a service.

## References

1. Kurki, J., Hyvönen, E.: Collaborative metadata editor integrated with ontology services and faceted portals. In: Workshop on Ontology Repositories and Editors for the Semantic Web (ORES 2010), the Extended Semantic Web Conference ESWC 2010, CEUR Workshop Proceedings, Vol. 596 (June 2010)
2. Mäkelä, E., Viljanen, K., Alm, O., Tuominen, J., Valkeapää, O., Kauppinen, T., Kurki, J., Sinkkilä, R., Kansala, T., Lindroos, R., Suominen, O., Ruotsalo, T., Hyvönen, E.: Enabling the semantic web with ready-to-use web widgets. In Nixon, L.J.B., Cuel, R., Bergamini, C., eds.: FIRST. Volume 293 of CEUR Workshop Proceedings., CEUR-WS.org (2007) 56–69
3. Lindquist, T., Hyvönen, E., Törnroos, J., Mäkelä, E.: Leveraging linked data to enhance subject access - a case study of the University of Colorado Boulder's World War I collection online. [8]
4. Hyvönen, E., Alm, O., Kuittinen, H.: Using an ontology of historical events in semantic portals for cultural heritage. In: Proceedings of the Cultural Heritage on the Semantic Web Workshop at the 6th International Semantic Web Conference (ISWC 2007). (November 12 2007)
5. Mäkelä, E., Hypén, K., Hyvönen, E.: Improving fiction literature access by linked open data -based collaborative knowledge storage - the BookSampo project. [8]
6. Hyvönen, E., Mäkelä, E.: Semantic autocompletion. In Mizoguchi, R., Shi, Z., Giunchiglia, F., eds.: ASWC. Volume 4185 of Lecture Notes in Computer Science., Springer (2006) 739–751
7. Jusevičius, M., Smirnovas, A., Šėporaitis, J.: Graphity – a generic linked data platform. In: Linked Enterprise Data Patterns Workshop 2011, Proceedings. (2011)
8. World Library and Information Congress: 78th IFLA General Conference and Assembly, Helsinki, Finland, August 11-17, 2012. Proceedings, World Library and Information Congress (2012)

---

<sup>8</sup> <http://github.com/tkurz/skosjs>

<sup>9</sup> <http://wifo5-03.informatik.uni-mannheim.de/pubby/>