Authority Control of People and Organizations on the Semantic Web

Jussi Kurki and Eero Hyvönen

Semantic Computing Research Group (SeCo)
Helsinki University of Technology (TKK) and University of Helsinki
http://www.seco.tkk.fi/
firstname.lastname@tkk.fi

Abstract. Authors and documents with identical titles are common in the digital library environment. In order to manage identities correctly, authority control is used by library and information scientists for disambiguating and cross-referencing entity names. We argue that the benefits of traditional authority control can be enhanced by using techniques and technologies of the Semantic Web, leading to simpler management of multiple languages, better linkability of resources, simpler reuse of authority registries in applications, and less work in indexing. To demonstrate our propositions, we have created a prototype of an ontology server and service called ONKI People that is used in two ways: First, it is a centralized authority service providing human end-users with efficient and easy to use authority finding and disambiguation services based on faceted semantic search and visualizations. The services are available online also as AJAX and Web Services API for machines to use. Second, the underlying RDF triple store can be used as a content resource in applications such as semantic cultural heritage portals. The paper discusses and demonstrates both use cases in a real life setting.

1 Towards Authority Control on the Semantic Web

Authority control includes the processes of maintaining author, title, and subject headings for bibliographic material (person, group or organization) in a library catalog [1, 2]. The basic problems addressed here are: 1) How how to encode names referring to the same entity in a systematic way, so that the resources can later be found by searching (e.g., names can be transliterated differently in different languages)? 2) How to guarantee that different entities are not encoded by a similar name, which would lead to confusions in information retrieval (e.g., different intances of “John Smith” encoded with the same string)?

Current methodology [1] relies on rigid syntax and rules for presenting the information. The basic sets of rules define the forms of the names (e.g. use the form *Surname, First name* for persons’ names), rules for changing the records when the names change, and so on. The main goal is to enable efficient search and retrieval based on author, title and subject names, that are often ambiguous, encoded in varying ways, and are subject to change.

Traditional authority control works well on small homogeneous environments and datasets, but as a downside, it requires much expertise from the indexers and manual work. These problems are emphasized when managing and interlinking large databases, such as authority records of different libraries in different countries based on different languages. Such tasks are becoming more and more popular on the Web. At the same time, less and less experienced people are indexing content at, e.g., various Web 2.0 sites. To cope with these trends, new kind of approaches and tools for authority control are needed on the Web. As a solution approach, we propose Semantic Web technologies[1].

A major idea of the Semantic Web is to identify entities, called resources, using Uniform Resource Identifiers[2] (URI), whose uniqueness can be guaranteed using the Domain Name Server system of the Web. The various linguistic representations of an identity are represented as literal properties attached to the URI using the Resource Description Framework (RDF)[3] for describing resources. By transforming the authority data about actors (persons, groups of people, organizations etc.) into RDF-based formats of the Semantic Web, this important data becomes part of the ever growing Web of Data[4]. Such content can be linked with related contents, transformed and enriched with simple tools and reasoning rules, and be queried using standard protocols such as the SPARQL Query Language for RDF[5]. On the semantic web, authority records are used in two ways: First, they are used in the traditional way for authority control, e.g., for finding unique identifiers for identities during indexing. Second, the authority RDF files can be exploited as a reusable *content repository* for applications, such as semantic portals for cultural heritage [3].

In the following, we first analyze problems with the traditional authority control approach. As a solution approach an authority record ontology is presented based on globally unique URIs and RDF. Two use cases of the RDF-based authority records are then demonstrated: First, we present ONKI People, a prototype of a national ontology service of authorities that is a part of the Finnish FinnONTO content infrastructure [4] and the National Ontology Service Library ONKI [5]. Second, we show how the ontological authority data has been reused in the semantic cultural heritage portal CultureSampo[6] [6] for publishing collections of museums, libraries, and archives.

2 Authority Control and Its Problems

Authority control has traditionally two main objectives [1]:

1. Find a work (e.g. a book or an article) whose the author, title or subject is known.

[1] http://www.w3.org/2001/sw/
[2] http://www.w3.org/Addressing/
[3] http://www.w3.org/RDF/
[4] http://linkeddata.org/
[5] http://www.w3.org/TR/rdf-sparql-query/
[6] http://www.kulttuurisampo.fi/

2. List all works by a given author (or subject or another attribute like genre).

These objectives are called *finding* and *collocating* objectives, respectively.

More recent work emphasizes the user's goals. Functional Requirements for Bibliographic Records[7] (FRBR) identifies the following user tasks: 1) Find an entity (this is similar to find and collocate objectives). 2) Identify an entity (confirm that the entity found corresponds to the entity sought). 3) Select an entity (select from various manifestations of an entity, e.g. a CD, DVD or book). 4) Obtain an entity. 5) Navigate (through related material).

A typical solution to meet the requirements is to build an authorized record for each document and actor. The record contains titles (and possibly their sources) and cross references. An example of an authority record is shown in Table 1, taken from a requirements document by the Functional Requirements and Numbering of Authority Records (FRANAR)[8] working group.

Authorized heading
    Mertz, Barbara
Information note/see also references:
    Barbara Mertz also writes under the pseudonyms Barbara
        Michaels and Elizabeth Peters.
    For works written under those pseudonyms, search also under:
    >> Michaels, Barbara, 1927-
    >> Peters, Elizabeth
See also reference tracings:
    << Michaels, Barbara, 1927-
    << Peters, Elizabeth

Table 1. Format for authority record by the FRANAR Working Group.

The record is identified by the authorized heading. The format of the heading is strictly defined as it glues the authorized record with the actor's actual works, such as books or articles. The additional information on the record helps the user to track related records and sources. It can also be used to disambiguate authors with similar or identical names. Automatic tools for creating authority records include clustering [7] and other name matching algorithms such as [8, 9], but even with these methods, human interaction is often required.

Authority records are often praised for their high quality. Maintaining costs are traditionally regarded as the most weighting drawback. Emerging problems include reusing records between different libraries, museums, and achieves for aggregating contents. We believe that many of these problems originate from the archaic syntax used for representing the records (e.g. the MARC formats[9]), and lack of common, shared vocabulary and repositories for authority records.

[7] http://www.frbr.org/
[8] http://archive.ifla.org/VII/d4/wg-franar.htm
[9] http://www.loc.gov/marc/

Still, one of the major problems is that an authority record is a list of literal values, where the record is identified by a selected special name. Selecting the name and its form in a systematic way in a global, distributed, multilingual, temporal environment is often a tricky problem. Furthermore, there is the problem of matching the selected authority name(s) with the actual name(s) used in the library databases, where different conventions may proliferate even within a single collection by different catalogers. In many cases, authority files are intended for human usage, and are difficult to interpret by machines—a central task on the Semantic Web.

In summary, problems of traditional authority records include the following:

1. Maintaining authority records is costly, since it requires lots of expertise and handwork.
2. Aggregating content is difficult, since different naming conventions are in use in different authority files and library databases.
3. Records evolve in time, e.g., a person may change her name, which leads to different annotations in different times.
4. Records may use complicated syntax and metadata formats that make it difficult to make contents mutually interoperable [10].
5. Records based on literal expressions do not link uniquely or straightforward with Web resources.
6. Efforts to build records are difficult to share at least on the level of the Web.

A recent approach to overcome some of the problems with authority files is the Virtual Authority Files[10] (VIAF). It attempts to aggregate the authority files of the Library of Congress, the Deutsche Nationalbibliothek, and the Bibliothéque Nationale de France under one service. The work is controlled by a central authority and is based on MARC.

3 Actor Ontology

The model for our Actor Ontology is based on existing vocabularies for describing persons, especilly FOAF[11], Relationship[12] and BIO[13] vocabularies [11]. These vocabularies are not especially designed for authority files. However, they were chosen in order to make the RDF autority files semantically interoperable with dozens of projects and tools on the web. If the vocabularies lack some properties that are essential, they can be added and format can be grown as needed. In this way extended FOAF-records are downward compatible with core FOAF records—extra information does not create problems. This approach, widely used on the Semantic Web, is more flexible than traditional database application schemas, where the fields are decided and fixed once and for all, and after that everybody have to live with them in good and in bad.

[10] http://www.oclc.org/research/projects/viaf/
[11] http://xmlns.com/foaf/spec/
[12] http://vocab.org/relationship/
[13] http://vocab.org/bio/0.1/

FOAF vocabulary defines 12 classes and about 50 properties. We use four classes from FOAF, *Actor* and its subclasses *Person*, *Group* and *Organization*. Properties we use include: *firstName*, *surname* and *knows* (*knows* being the essence of FOAF in building the who-knows-who index). The Relationship vocabulary defines sub-properties for *foaf:knows*, we use e.g. *childOf*, *friendOf* and *siblingOf*. The BIO vocabulary defines biographical events, such as *Birth*, *Death* and *Marriage*, which have place and time. All events extend the generic *Event* class thus making the model flexible and extendable. Finally we have defined own classes and properties to cover information presented in ULAN. The classes we have defined include nationalities (*European*, *Frech*, *Japanese*,...) and roles (*Artist*, *Painter*, *Collector*,...). The properties include *teacherOf*, *patronOf*, *masterOf* etc. The key classes used in the actor ontology are summarized Table 2 and properties in Table 3.

Vocabulary	Class	Description
FOAF	*Agent*	Super-class for all FOAF actors
FOAF	*Person*	One individual
FOAF	*Group*	Group of persons or other groups
FOAF	*Organization*	Organization, i.e. company
BIO	*Event*	Super-class for biographical events
BIO	*Birth*	Class presenting birth event
BIO	*Death*	Class presenting death event
ACTOR	*Artist*	Class presenting the role artist
ACTOR	*Painter*	Class presenting the role painter (sub-class of Artist)
ACTOR	*X*	Class presenting the role X
ACTOR	*European*	Class presenting nationality European
ACTOR	*Scandinavian*	Class presenting nationality Scandinavian (sub-class of European)
ACTOR	*Finnish*	Class presenting nationality Finnish (sub-class of Scandinavian)
ACTOR	*Y*	Class presenting the nationality Y

Table 2. Classes used in actor ontology.

The Semantic Web is built around the RDF data model. It is a simple way to present data as triples of the form

(*subject, predicate, object*),

e.g. (*"Picasso", "roleIs", "Artist"*). The resulting sematic net (graph), where each triple represents an arc between two nodes in the graph, can be grown easily and infinitely by adding new triples telling e.g. more about Picasso, about the procession Artist, and about what is the meaning of the predicate property roleIs. (Properties, i.e., arcs, are at the same time nodes in the network and can be attached with metadata, too.) All content based on RDF (or other semantic web formats based on it) share the domain independent logical semantics underlying the system in use, which makes it possible for the machine to aggregate

Vocabulary	Class	Description
FOAF	*name*	Name of person, group or organization
FOAF	*firstName*	First name
FOAF	*surname*	Surname
FOAF	*knows*	Generic "knows" relationship
RELATIONSHIP	*childOf*	Sub-property of *foaf:knows*
RELATIONSHIP	*friendOf*	Sub-property of *foaf:knows*
RELATIONSHIP	*spouseOf*	Sub-property of *foaf:knows*
RELATIONSHIP	*siblingOf*	Sub-property of *foaf:knows*
ACTOR	*studentOf*	Sub-property of *foaf:knows*
ACTOR	*patronOf*	Sub-property of *foaf:knows*
ACTOR	*assistantOf*	Sub-property of *foaf:knows*
BIO	*place*	Place of *bio:Event*
BIO	*date*	Date of *bio:Event*

Table 3. Key properties used in actor ontology.

and interpret the information in different applications. Another key to enable the aggregation of the Web of Data, is to use shared, domain sepcific vocabularies and resources, such as authority indentifiers, in the content description. By using logic and by giving things global identifiers (URIs) taken from shared vocabularies (ontologies), semantic web repositories can be at the same time aggregated in an interoperable way over domain borders, and new information can be inferred. Figure 1 shows an example of the semantic network of resources around a person. Here the person with ID toimo:p12 has name "Jussi Kurki", was born in 1982 in the Place "Helsinki", works with a person whose name is "John Smith", and so on.

To test the idea of publishing and using authority files on the Semantic Web in practice we decided to use the Union List of Artist Names (ULAN)[14] as a starting point. ULAN consists of over 120,000 individuals and corporate bodies of art historical significance, with over 300,000 names. In addition, the dataset includes comprehensive information about relationships between actors. ULAN was transformed into FOAF/RDF format using XSL-transformations. Figure 2 shows as an example first lines of the ULAN record of "Gallen-Kallela, Akseli", a Finnish artist, with his ID and alternative names.

3.1 URI Identifiers

URIs are used for identifying things on the Semantic Web. However, in FOAF, the global identifiers are not used. A reason for this is that there are no global repositories available for distibuting and managing identifiers for contemporary ordinary persons. Instead, person or group is identified by a set of unique properties, such as the personal email address. In this way it is possible to identify two resources as the same even if they have different identifyers. The process of merging data from different sources is called "smushing". Depending on the

[14] http://www.getty.edu/research/conducting_research/vocabularies/ulan/

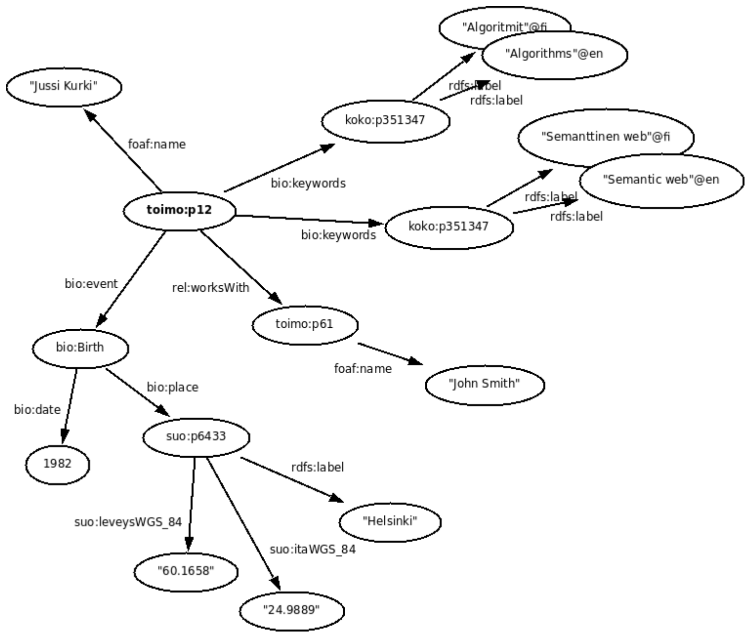

Fig. 1. Semantic web connecting persons to places, concepts, other persons etc.

Fig. 2. Different names of Finnish artist Gallen-Kallela displayed on ULAN web site.

dataset different smushing strategies are needed. For example, historical persons do not have personal email addresses but the year of birth, year of death, name, profession, nationality or some combination of these can be utilized in disambiguation.

In our Actor Ontology we are using identifiers (URIs) in the same way as in ULAN. It seems reasonable to try to identify persons by identifiers when this is possible. Using IDs is ealier and more error proof by using indirect indirect reasoning. However, using URIs and smushing hare not mutually exclusive strategies. If some entity does not have a URI or information about the entity is divided into two records, smushing is one way to try and aggregate the records and IDs in use.

We have listed below desirable requirements for the actor ontology URI identifiers:

1. URIs are persistent and do not change e.g. when a persons name changes.
2. URIs are not dependent on the language in use which makes them multilingual.
3. URIs are not dependent on the data described making them universally applicable.
4. URIs globally unique making diambibuation easy.
5. URIs can be generated uniquely in a decentralized manner.

The last requirement conforms to the notion of Universally Unique Identifier[15] (UUID) that are quaranteed to be unique. An example of a random unique URI is as follows:

`http://seco.tkk.fi/onto/toimo/p05fe15c7da2fdecd387985f92af6c5484e930fd1`

URIs are used in indexing time for storing references to actors in an unambiguous and persistent way. During information retrieval (IR), IDs are needed for disambiquating semantically end-user queries. When querying e.g. books authored by “John Smith”, the system can for example show the end-user a list of known potential John Smiths to select from, and then use the corresponding ID for precise IR.

In order to facilitate such services, a centralized ontology ID service, used in a distributed manner by application systems is needed. In addition, a possibility for generating new unique IDs in a decenttralized way, and smushing them together later on, is necessary.

In the following we present an centralized ontology service called ONKI People. It addresses the question of resolving the URIs for entities expressed by a name. ONKI People is a centralized repository of persons and organizations that offers services for searching as well as disambiguating people and organizations.

4 ONKI People

The key features of ONKI People are a multifaceted search component and a graph visualizer component.

[15] http://www.ietf.org/rfc/rfc4122.txt

4.1 Faceted Search

Search starts when the user types one or more keywords to the search box and hits enter (Figure 3). The search covers all different forms of names as well as nationalities, roles and other properties related to actors. The user can refine the search by selecting categories from the facets on the left hand side.

Search is multilingual and enhanced by the underlying ontology on nationalities and roles. User can type 'German' or 'Deutsche', assuming that the ontology on the nationalities is multilingual, to receive the same results. Also typing 'European' would yield to a search on German, French, Italian, etc. actors.

Because all forms of names are indexed, the user may type 'Ruys Pablo' or 'Picasso' to find the famous artist. Also different scripts are supported since all data is encoded using Unicode. Query results on the other hand are localized and based on the language in use. In the result list, a localized summarising heading is shown for each actor. The headings are of the same form as in ULAN. An example is given below in with an English and a Finnish heading:

```
Gallen-Kallela, Akseli (Finnish painter and graphic artist, 1865-1931)
Gallen-Kallela, Akseli (suomalainen taiteilija, 1865-1931)
```

Fig. 3. ONKI People showing the search results for keyword "napoleon".

4.2 Social Graph Visualization

If the user clicks on an actor from the result list (see previous section), the social network of that actor, i.e., a circle of related actors is displayed. For example,

Figure 4 depicts the social network of Napolean I, emperor of France. The user can further click on any neighbouring nodes in the graph to see the social network of that actor. The graphs are rendered as SVG[16] images. Nodes are positioned by a simple algorithm which places direct contacts around the actor, friends of friends on the second level, and so on.

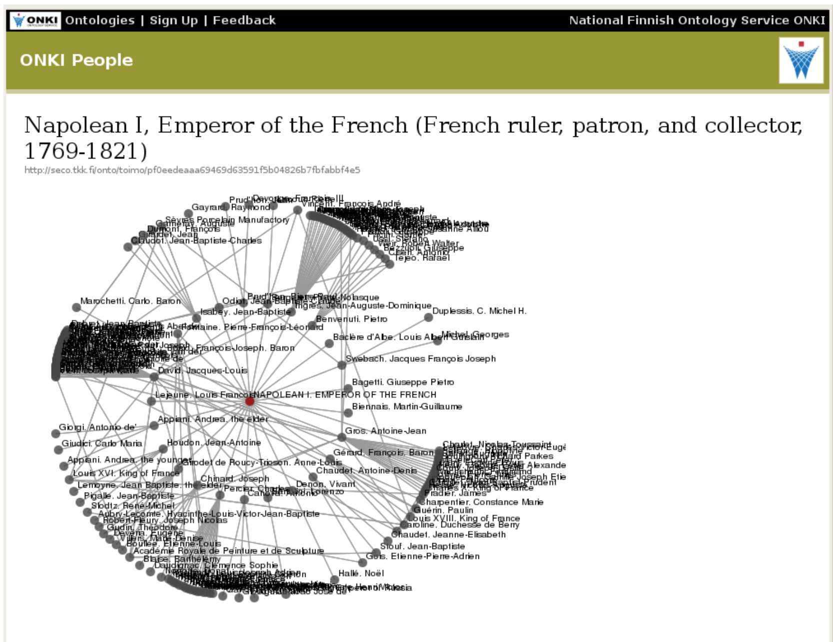

Fig. 4. Displaying the social circle of Napoleon I in ONKI People.

The network can be used to disambiguate actors with similar names. Also roaming social connections is in itself interesting to the user.

4.3 ONKI Selector

ONKI Selector[17] is a ready-to-use service for creating "mash-up" applications cost-efficiently. The selector on actor ontology is shown in Figure 5. The selector can be integrated into an existing web-based indexing system with one or two lines of Javascript code. ONKI Selector can be seen as a specialized input element. It returns either URIs or labels (or both) directly into a desired field. It works in a cross-domain manner, though this is invisible to the user. The end-user does not need to concern himself whether he is using a local input element

[16] http://www.w3.org/Graphics/SVG/About
[17] http://www.yso.fi/onkiselector/

or a cross-domain selector widget. Using the selector, one can add authority control into one's own system with minimal effort.

Fig. 5. ONKI Selector on actor ontology.

4.4 Generic Machine Interfaces

Besides the human oriented web interface and ONKI Selector interface, ONKI People can be published through various machine interfaces. These include a cross-domain AJAX Javascript interface using DWR[18] and a Web Service interface using CXF[19].

4.5 Key Points of the Implementation

ONKI People was implemented in Java on top of the Spring framework[20]. The application follows the Model-View-Controller (MVC) pattern where display logic is separated from the data model. JSP[21] and XSLT[22] were used as a view layer. The faceted search engine is backed by Lucene[23] indexing. In the visualizer component, SVG graphs are rendered directly to a HTTP response to avoid the need of caching and disk operations. Other optimizations include compression of HTTP packets for faster page load times.

18 http://directwebremoting.org/
19 http://cxf.apache.org/
20 http://springframework.org/
21 http://java.sun.com/products/jsp/
22 http://www.w3.org/TR/xslt
23 http://lucene.apache.org/

5 Applications

We have tested our data in the semantic cultural heritage portal CultureSampo[24] [6]. Authority data has a two key functions in the portal. First, it acts as a supporting element for search and resource linking. Second, the data in itself can be mined using a method we call *relational search*. In the following, we fist introduce the idea of relational search. In the subsequent section, we show how authority data can be utilized by using simple rule-based recommendations.

5.1 Relational Search

Semantic association identification has been studied in national security applications [12]. We have adapted this notion to the cultural heritage domain. The idea is to make it possible for the end-user to formulate queries such as "How is X related to Y" by selecting the end-point resources. The result is a set of semantic connection paths between X and Y. The data used in CultureSampo is the same that is used in ONKI People, i.e., the RDF-transformed ULAN dataset. An example of relational search is shown in Figure 6, showing the social connections between Akseli Gallen-Kallela, the Finnish artist, and Napoleon I.

Fig. 6. Finding social paths between actors.

The search is implemented as an uniformed breadth-first search. As a generic graph search, any relations—not just social connections—could be searched. The graph is built directly from the RDF-graph. RDF-resources are nodes (i.e. actor entities that have URI) and properties (i.e. *friendOf*) are edges.

The implementation is in Java. The graph is stored as a memory based adjacency list. To minimize memory consumption, graph nodes have only a minimal

[24] http://www.kulttuurisampo.fi

set of fields: an URI and a list of children. At this point, all relationships are basically reduced to "knows" and all data is reduced to URIs. Serialized on the disk, the whole graph takes about 10MB of memory.

Even the longest paths (12 steps) can be found in less than 0.5 seconds. This is explained partly by the structure of the ULAN data. The graph has a strongly connected component of about 12,000 actors containing major artists, such as Picasso and Donatello. At the same time, thousands of others, especially contemporary artists, don't have many connections in the underlying RDF graph.

5.2 Portal Recommendations

CultureSampo builds recommendations using SPARQL-queries[25]. A page about the artist Akseli Gallen-Kallela, based on ULAN, is shown in Figure 7. In addition to the data about the artist, paintings and drawings created by Gallen-Kallela are automatically shown, based on various museum collections (i.e. works of which creator is annotated by the artist URI). Recommendations shown in the example are dynamically built by the following rule:

```
SELECT ?item WHERE {
   ?item kulsa-schema:creator $this
}
```

Parameter *$this* is replaced by an actor URI. Return value *?item* is a list of related item URIs. Recommendations can be made more complex by adding further rules. For example, the following rule would return all works of an artist's pupils.

```
SELECT ?item WHERE {
   ?student actor:studentOf $this
   ?item kulsa-schema:creator ?student
}
```

Recommendations are dynamically calculated and easily configurable. Rebooting the server is the only requirement for new rules to become visible. Though designed for recommendations, the SPARQL endpoint of CultureSampo could be used for other purposes as well.

6 Discussion

This paper argued that Semantic Web technologies are useful in authority control, especially when dealing with heterogeous, multilingual contents created by non-professional indexers in a distributed manner—an evermore typical situation on the Web. In contrast to traditional authority control and database search, indexing and IR on the Semantic Wed is based on URIs, a semantic RDF network connecting them, shared ontologies, and logical reasoning based on globally

[25] http://www.w3.org/TR/rdf-sparql-query/

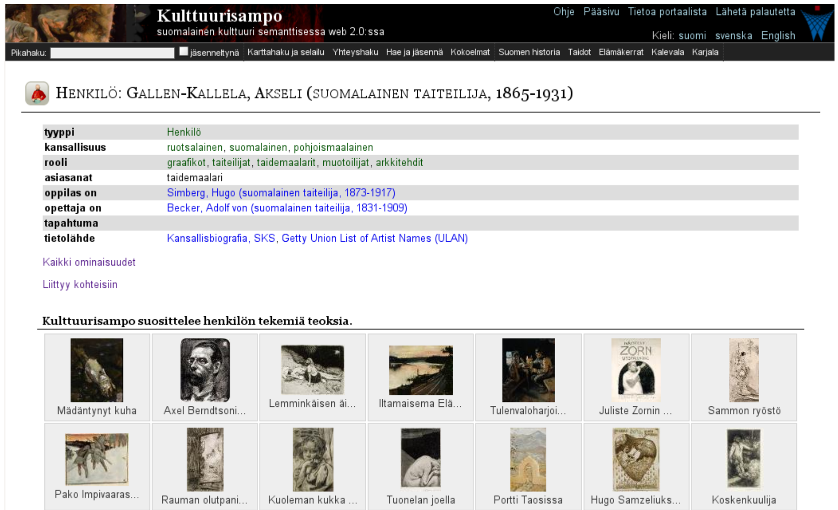

Fig. 7. Kulttuurisampo displays the record of Finnish artist Akseli Gallen-Kallela and on the same page recommends works by him.

shared semantic interpretations and standards. Benefits of the approach were discussed and demonstrated in two practical use cases and implementations: an ontology service and a semantic portal for cultural heritage.

ONKI People is a unique authority ontology service for actors based on the Semantic Web approach. The solution builds, however, on various pieces of related work, some of which are listed below. The faceted (view-based) search paradigm used in ONKI People has been discussed e.g. in [13, 14]. The idea in ONKI People to support mash-up usage of ontologies as services in legacy systmes is based on [4, 15, 5]. In [16], another system is presented using an integrable autocompletion widget. Relational search (association identification) is discussed e.g. in [12].

Acknowledgements This work is part of the National Semantic Web Ontology project in Finland[26] (FinnONTO, 2003-2010), funded mainly by the National Technology and Innovation Agency (Tekes) and a consortium of 38 organizations. The work is also partly funded by the FP7 EU Project SmartMuseum[27].

References

1. Taylor, A.: Introduction to Cataloging and Classification. Library and Information Science Text Series. Libraries unlimited (2006)

[26] http://www.seco.tkk.fi/projects/finnonto/
[27] http://www.smartmuseum.eu

2. Tillett, B.: Authority control: State of the art and new perspectives. In: International Conference on Authority Control, Haworth Press, Binghamton, NY (2004)
3. Hyvönen, E.: Semantic portals for cultural heritage. In Staab, S., Studer, R., eds.: Handbook on Ontologies (2nd Edition), Springer–Verlag (2009)
4. Hyvönen, E., Viljanen, K., Tuominen, J., Seppälä, K.: Building a national semantic web ontology and ontology service infrastructure—the FinnONTO approach. In: Proceedings of the ESWC 2008, Tenerife, Spain, Springer–Verlag (2008)
5. Viljanen, K., Tuominen, J., Hyvönen, E.: Ontology libraries for production use: The Finnish ontology library service ONKI. In: Proceedings of the ESWC 2009, Heraklion, Greece, Springer–Verlag (2009)
6. Hyvönen, E., Mäkelä, E., Kauppinen, T., Alm, O., Kurki, J., Ruotsalo, T., Seppälä, K., Takala, J., Puputti, K., Kuittinen, H., Viljanen, K., Tuominen, J., Palonen, T., Frosterus, M., Sinkkilä, R., Paakkarinen, P., Laitio, J., Nyberg, K.: CultureSampo—Finnish culture on the Semantic Web 2.0. Thematic perspectives for the end-user. In: Museums and the Web 2009 Proceedings, Archives & Museum Informatics, Toronto (2009) http://www.archimuse.com/mw2009/papers/hyvonen/hyvonen.html.
7. French, J., Powell, A., Schulman, E.: Using clustering strategies for creating authority files. Journal of the American Society for Information Science **51**(8) (jun 2000) 774–786
8. Galvez, C., Moya-Anegon, F.: Approximate personal name-matching through finite-state graphs. Journal of the American Society for Information Science and Technology **58**(13) (2007) 1960–1976
9. Borgman, C., Siegfriend, S.: Getty's synoname and its cousins: A survey of applications of personal name-matching algorithms. Journal of the American Society for Information Science and Technology **43**(7) (1992) 459–476
10. Baca, M., ed.: Introduction to metadata. The Getty Research Institute, Los Angeles (2008)
11. Aleman-Meza, B., Bojars, U., Boley, H., Breslin, J., Mochol, M., Nixon, L., Polleres, A., Zhdanova, A.: Combining rdf vocabularies for expert finding. In Franconi, E., Kifer, M., May, W., eds.: ESWC. Volume 4519 of Lecture Notes in Computer Science., Springer (2007) 235–250
12. Sheth, A., Aleman-Meza, B., Arpinar, F.S., Sheth, A., Ramakrishnan, C., Bertram, C., Warke, Y., Anyanwu, K., Aleman-meza, B., Arpinar, I.B., , Kochut, K., Halaschek, C., Ramakrishnan, C., Warke, Y., Avant, D., Arpinar, F.S., Anyanwu, K., Kochut, K.: Semantic association identification and knowledge discovery for national security applications. Journal of Database Management **16** (2005) 33–53
13. Pollitt, A.S.: The key role of classification and indexing in view-based searching. Technical report, University of Huddersfield, UK (1998) http://www.ifla.org/IV/ifla63/63polst.pdf.
14. Hearst, M., Elliott, A., English, J., Sinha, R., Swearingen, K., Lee, K.P.: Finding the flow in web site search. CACM **45**(9) (2002) 42–49
15. Tuominen, J., Frosterus, M., Viljanen, K., Hyvönen, E.: ONKI SKOS server for publishing and utilizing SKOS vocabularies and ontologies as services. In: Proceedings of the ESWC 2009, Heraklion, Greece, Springer–Verlag (2009)
16. Hildebrand, M., van Ossenbruggen, J., Amin, A., Aroyo, L., Wielemaker, J., Hardman, L.: The design space of a configurable autocompletion component. Technical Report INS-E0708, Centrum voor Wiskunde en Informatica, Amsterdam (2007)