

ABSTRACT OF DOCTORAL DISSERTATION		AALTO UNIVERSITY SCHOOL OF SCIENCE AND TECHNOLOGY P.O. BOX 11000, FI-00076 AALTO http://www.aalto.fi	
Author Eetu Mäkelä			
Name of the dissertation View-Based User Interfaces for the Semantic Web			
Manuscript submitted May 11th, 2010		Manuscript revised August 11th, 2010	
Date of the defence November 26th, 2010			
<input type="checkbox"/> Monograph		<input checked="" type="checkbox"/> Article dissertation (summary + original articles)	
Faculty	Faculty of Information and Natural Sciences		
Department	Department of Media Technology		
Field of research	Computer Science, Semantic Web, User Interfaces		
Opponent(s)	Professor Lora Aroyo		
Supervisor	Professor Eero Hyvönen		
Instructor	Professor Eero Hyvönen		
<p>Abstract</p> <p>This thesis explores the possibilities of using the view-based search paradigm to create intelligent user interfaces on the Semantic Web. After surveying several semantic search techniques, the view-based search paradigm is explained, and argued to fit in a valuable niche in the field. To test the argument, numerous portals with different user interfaces and data were built using the paradigm. Based on the results of these experiments, this thesis argues that the paradigm provides a strong, extensible and flexible base on which to build semantic user interfaces. Designing the actual systems to be as adaptable as possible is also discussed.</p>			
Keywords Semantic Web, view-based search, faceted navigation, user interface design, system design			
ISBN (printed)	978-952-60-3446-1	ISSN (printed)	1795-2239
ISBN (pdf)	978-952-60-3447-8	ISSN (pdf)	1795-4584
Language	English	Number of pages	86 + 129
Publisher Aalto University School of Science and Technology, Department of Media Technology			
Print distribution Aalto University School of Science and Technology, Department of Media Technology			
<input checked="" type="checkbox"/> The dissertation can be read at http://lib.tkk.fi/Diss/2010/isbn9789526034478/			

VÄITÖSKIRJAN TIIVISTELMÄ		AALTO-YLIOPISTO TEKNILLINEN KORKEAKOULU PL 11000, 00076 AALTO http://www.aalto.fi	
Tekijä Eetu Mäkelä			
Väitöskirjan nimi Semanttisen webin näkymäpohjaiset käyttöliittymät			
Käsikirjoituksen päivämäärä 11.5.2010		Korjatun käsikirjoituksen päivämäärä 11.8.2010	
Väitöstilaisuuden ajankohta 26.11.2010			
<input type="checkbox"/> Monografia		<input checked="" type="checkbox"/> Yhdistelmäväitöskirja (yhteenvedo + erillisartikkelit)	
Tiedekunta	Informaatio- ja luonnontieteiden tiedekunta		
Laitos	Mediatekniikan laitos		
Tutkimusala	Tietojenkäsittelytiede, semanttinen web, käyttöliittymät		
Vastaväittäjä(t)	Professori Lora Aroyo		
Työn valvoja	Professori Eero Hyvönen		
Työn ohjaaja	Professori Eero Hyvönen		
<p>Tiivistelmä</p> <p>Tämä työ selvittää mahdollisuuksia soveltaa näkymäpohjaista hakuparadigmaa älykkäiden semanttisen webin käyttöliittymien pohjana. Työ alkaa selvityksellä olemassaolevista semanttisen webin hakujärjestelmistä. Tämän jälkeen esitellään näkymäpohjaisen haun paradigma ja esitetään sen soveltuvan hyvin semanttisen webin käyttöliittymien pohjaksi. Väitteen testaamiseksi rakennettiin useita semanttisia moninäkömahakuportaaleja eri käyttötarkoituksiin ja aineistoille. Saadut tulokset osoittavat että moninäkömahakuparadigma tarjoaa toimivan, hyvin laajennettavan ja mukautuvan pohjan semanttisten käyttöliittymien rakentamiseen. Työ esittelee myös tutkimuksen aikana selvitettyjä suunnitteluperiaatteita, joiden avulla semanttisen webin tietojärjestelmistä voidaan tehdä mahdollisimman laajennettavia ja mukautuvia.</p>			
Asiasanat semanttinen web, näkymäpohjainen haku, käyttöliittymäsuunnittelu, järjestelmäsuunnittelu			
ISBN (painettu)	978-952-60-3446-1	ISSN (painettu)	1795-2239
ISBN (pdf)	978-952-60-3447-8	ISSN (pdf)	1795-4584
Kieli	englanti	Sivumäärä	86 + 129
Julkaisija Aalto-yliopiston teknillinen korkeakoulu, Mediatekniikan laitos			
Painetun väitöskirjan jakelu Aalto-yliopiston teknillinen korkeakoulu, Mediatekniikan laitos			
<input checked="" type="checkbox"/> Luettavissa verkossa osoitteessa http://lib.tkk.fi/Diss/2010/isbn9789526034478/			

Contents

Contents	i
List of Publications	iii
Author’s Contribution	vii
List of Figures	ix
1 Introduction	1
1.1 Semantic Web Technologies	2
1.2 Research Questions and Methodology	5
1.3 Thesis Contributions	8
1.4 Thesis Structure	9
2 Survey of Semantic Search Research	12
2.1 Research Directions in Semantic Search	12
2.1.1 Augmenting Traditional Keyword Search with Semantic Tech- niques	13
2.1.2 Basic Concept Location	15
2.1.3 Complex Constraint Queries	17
2.1.4 Problem Solving	19
2.1.5 Connecting Path Discovery	20
2.2 Common Methodology	20
2.2.1 RDF Path Traversal	21
2.2.2 Mapping Between Keywords and Concepts	21
2.2.3 Graph Patterns	22
2.2.4 Logics	22
2.2.5 Combining Uncertainty with Logics	23
2.3 Conclusions Drawn from the Survey	23

3	Applying View-Based Concepts to the Semantic Web	25
3.1	The View-Based Search Paradigm	25
3.2	View Projection from Ontologies	29
3.3	Complementing View-Based Search with Semantic Autocompletion	31
3.4	View-Based Search as a General Base for Semantic Interfaces	33
4	Adaptability of Semantic View-Based Interfaces	35
4.1	A View-Based Search Interface for Browsing	36
4.2	A View-Based Search Interface for Efficient Search	43
4.3	Further Interfaces	46
4.4	Adaptability of the Paradigm to Different Domains	47
4.5	Expanding the Paradigm to Heterogeneous Datasets	49
4.5.1	Problem Definition	49
4.5.2	An Event-Based Approach	51
4.5.3	Domain-Centric View-Based Search	53
4.5.4	The Search and Organize User Interface Concept	54
4.5.5	Thematic Views	58
5	Design Issues for View-Based Semantic Web Interfaces	61
5.1	The Semantic Portal Creation Tool OntoViews	61
5.1.1	The Projection and Semantic Linking Engine Ontodella	62
5.1.2	The Semantic View-Based Search Engine Ontogator	62
5.2	Content Production Architecture of Semantic Portals	63
6	Discussion	65
7	Conclusions	72

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

- I** Eetu Mäkelä, Eero Hyvönen, Samppa Saarela and Kim Viljanen. 2004. OntoViews – A Tool for Creating Semantic Web Portals. In: Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen (editors), *The Semantic Web - ISWC 2004: Third International Semantic Web Conference*, Hiroshima, Japan, November 7-11, 2004. Proceedings, volume 3298 of *Lecture Notes in Computer Science*, pages 797–811. Springer. ISBN 3-540-23798-4.
- II** Eero Hyvönen, Eetu Mäkelä, Mirva Salminen, Arttu Valo, Kim Viljanen, Samppa Saarela, Miikka Junnila, and Suvi Kettula. 2005. MuseumFinland – Finnish museums on the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web* 3, no. 2-3, pages 224 – 241. Selected Papers from the International Semantic Web Conference, 2004 - ISWC, 2004.
- III** Eetu Mäkelä, Eero Hyvönen, and Teemu Sidoroff. 2005. View-Based User Interfaces for Information Retrieval on the Semantic Web. In: Abraham Bernstein, Ion Androutsopoulos, Duane Degler, and Brian McBride (editors), *End User Semantic Web Interaction Workshop*, volume 172 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- IV** Eero Hyvönen and Eetu Mäkelä. 2006. Semantic Autocompletion. In: Richiro Mizoguchi, Zhongzhi Shi, and Fausto Giunchiglia (editors), *The Semantic Web - ASWC 2006, First Asian Semantic Web Conference*, Beijing, China, September 3-7, 2006, Proceedings, volume 4185 of *Lecture Notes in Computer Science*, pages 739–751. Springer. ISBN 3-540-38329-8.

- V** Eetu Mäkelä, Eero Hyvönen, and Samppa Saarela. 2006. Ontogator - A Semantic View-Based Search Engine Service for Web Applications. In: Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold, and Lora Aroyo (editors), *The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings*, volume 4273 of *Lecture Notes in Computer Science*, pages 847–860. Springer. ISBN 3-540-49029-9.
- VI** Eero Hyvönen, Tuukka Ruotsalo, Thomas Häggström, Mirva Salminen, Miikka Junnila, Mikko Virkkilä, Mikko Haaramo, Eetu Mäkelä, Tomi Kauppinen, and Kim Viljanen. 2007. CultureSampo – Finnish Culture on the Semantic Web: The Vision and First Results. In: Klaus Robering (editor), *Information Technology for the Virtual Museum – Museology and the Semantic Web*, pages 33–58. LIT Verlag, Berlin. ISBN 978-3-8258-0262-2.
- VII** Eetu Mäkelä, Osmo Suominen, and Eero Hyvönen. 2007. Automatic Exhibition Generation Based on Semantic Cultural Content. In: Lora Aroyo, Eero Hyvönen and Jacco van Ossenbruggen (editors), *Cultural Heritage on the Semantic Web Workshop, 6th European Semantic Web Conference, ESWC 2009, Heraklion, Crete, Greece, May 31-June 4, 2009*, pages 41–52.
- VIII** Eero Hyvönen, Eetu Mäkelä, Tomi Kauppinen, Olli Alm, Jussi Kurki, Tuukka Ruotsalo, Katri Seppälä, Joeli Takala, Kimmo Puputti, Heini Kuitinen, Kim Viljanen, Jouni Tuominen, Tuomas Palonen, Matias Frosterus, Reetta Sinkkilä, Panu Paakkarinen, Joonas Laitio, and Katariina Nyberg. 2009. 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.

In addition to these publications, this thesis references other work by the author [37, 53, 71, 75] to provide context and further information on the subjects discussed.

Of the articles making up this thesis, the relationship between publications I and II bears further notice. Almost all of the content of publication I is repeated mostly verbatim in the later expanded publication II, making up over half of that paper. However, the differences that are there are very meaningful in the sense that publication I argues the subject from a general paradigmatic and systemic viewpoint, while publication II is written from the view of a single system. As the viewpoint of this thesis is paradigmatic and system-spanning, this difference necessitates the inclusion of publication I in addition to publication II.

Author's Contribution

The author is the primary writer of all articles where his name is given first. The original idea for combining the view-based search paradigm with the Semantic Web is by professor Eero Hyvönen, who also guided all the research.

In publications I and II, the author is the designer and primary developer of the semantic view-based search and browsing interface, as well as the whole mobile user interface.

The application of the OntoViews framework to the Suomi.fi data described in III was the work of Teemu Sidoroff. Otherwise the work described in that paper is by the author.

The author co-developed the concepts in publication IV, as well as co-wrote the article. He is responsible for the implementations of semantic autocompletion in the MuseumFinland, Veturi and CultureSampo portals.

The author is the primary designer and implementer of the combined OntoViews architecture discussed in publications I, II and V, except as noted in the following. The Ontogator search engine was designed and implemented by Samppa Saarela, except for the extension to support Prova projection rules and some design refactoring which were the work of the author. Samppa Saarela also collaborated in smaller part on the design and implementation of the first version of OntoViews-C, as well as the first complete version of the MuseumFinland interface. Publication V is based on an earlier draft created in collaboration with him, but the content and tests discussed therein were done by the author. The Ontodella server infrastructure, as well as the projection and recommendation rule formats are the work of Arttu Valo and Kim Viljanen.

In publication VI, the author is responsible for the architecture utilized in CultureSampo I. He is also responsible for the underlying general architecture, as well as the underlying architecture but not the user interface of the search functionality in CultureSampo II.

Except for some comments on user interface design, the author is solely responsible for publication VII.

For the final version of CultureSampo described in VIII, the author was the manager and chief architect of the project. He is responsible for the general orientation and layout of the user interface, as well as the keyword search, “Search for Items on the Map” and “Search and Organize” views. He is also responsible for the web widget functionality. In addition, he is responsible for the current general design principles of the metadata schemas used in the portal, as well as the data translations needed to create the portal.

List of Figures

1.1	A visualization of an example RDF network	3
1.2	An example of an ontology	4
1.3	A process model of design science research methodology [58]	11
2.1	A visual formulation of a query in the GRQL interface, along with the generated query language expression [5]	18
2.2	The SEWASIE visual tool for query formulation support [11]	19
3.1	The HiBrowse interface, with three hierarchical views [60]	27
3.2	A conceptual overview of view-based querying	28
3.3	An example of view projection	30
4.1	The main search view of MuseumFinland	38
4.2	The tree view of MuseumFinland	40
4.3	Entering keywords creates a dynamic facet of matching categories	41
4.4	The item view of MuseumFinland	42
4.5	The Veturi user interface	44
4.6	Domain Centric View Based Search	54
4.7	Exhibition room visualization in CultureSampo	57
4.8	Map visualization in CultureSampo	58
4.9	Timeline visualization in CultureSampo	58

1 Introduction

The Semantic Web [3, 7] is a technology for representing data on a semantic level, allowing for web-scale intelligent integration as well as inferencing based on that data. The benefits of such a coding lie in more efficient reuse of content, interlinking of content across institution bounds and increased interoperability between software systems. Encoding semantics already in the data also eases the creation of intelligent and ideally thus more usable applications. Major application sectors are those where there is a significant need and willingness for interoperability and integration of distributedly generated content: the cultural heritage domain, the health and welfare domain, e-government, business to business communication, subcontractor networks etc [37].

However, while the formal semantic coding of information on the Semantic Web makes it possible for applications to intelligently process that information, such annotations are not clear to an average human user [59]. In addition, the sheer amount of interlinked information can also easily become overwhelming [55, 66]. In user interface research, a core challenge then is in how to enable users to harness the power of the Semantic Web, while hiding the complexity [17, 27]. This thesis covers the work of the author in trying to meet this demand.

The context of this work is the FinnONTO¹ project [37]. The aim of this project is to make uptake of the Semantic Web in Finland as cost-effective as possible. This is done by creating and providing not only common Semantic Web vocabularies, but also ready-made functionality. This dictated an additional constraint for the work presented herein: all systems designed should be as adaptable as possible, both to new content as well as differing end-user needs. Thus, a large part of the work deals with how to create modular, adaptable systems and interfaces, making maximal use

¹<http://www.seco.tkk.fi/projects/finnonto/>

of the information already coded in the Semantic Web of data.

1.1 Semantic Web Technologies

The Semantic Web is based on encoding semantic-level information in a common formal way. To facilitate this, the Semantic Web relies on a common data model, and various semantics-specifying languages layered on top of it.

Underlying everything is the RDF data model [47], which specifies how information on the Semantic Web is to be represented. The model is based on a collection of simple triplets of the form (Subject, Predicate/Property/Relationship, Object), mimicking simple factual sentences such as (“Finland”, “is a part of”, “Europe”) or (“Finland”, “is a”, “country”). In RDF, however, each subject and relationship used in a statement has a global and unique identifier, while the object can either be another entity identifier, or a literal value. By using the same identifiers in multiple triplets, a net of nodes and arcs is formed, linking the triplets together into graphs, and thus allowing for more complex forms of information to be modeled and stored.

An example of an RDF network is depicted in figure 1.1, describing some metadata about this thesis. In RDF, globally addressable entities are demarcated by URIs, in the figure shortened using the XML namespace notation [8]. In the example, “e:vbsui” is related by the property “e:author” to an individual, whose “e:name” is “Eetu Mäkelä”. The “e:title” of “e:vbsui” is “View-Based User Interfaces for the Semantic Web”. It is “e:about” something referred to by the resource “e:semanticWeb”, as well as “e:about” something referred to by “e:search”.

There are still more complexities in the RDF model, such as blank nodes, collections and containers that group resources together, as well as reification, where statements can refer to other statements. Also these constructs are represented using the triplet

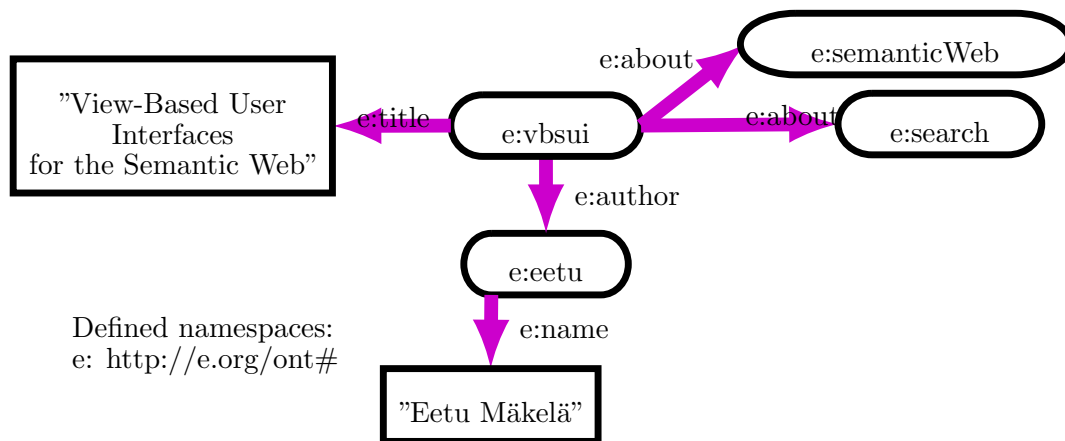


Figure 1.1: A visualization of an example RDF network

model. They are, however, not relevant to understanding this thesis, and thus will not be covered further here.

While the RDF data model provides a simple way to represent nearly any information, it does not generally specify what the used concepts and relations mean — what they entail. This is because RDF provides only a bare minimum of formal semantics [24]. For example, in the graph of figure 1.1, there is still nothing telling the computer what the blank node actually is, or what “e:author”, “e:title”, or “e:semanticWeb” mean.

On the Semantic Web, the further formal semantics still needed are provided by ontologies defined using RDFS [9] and OWL [50], the standard ontology languages of the Semantic Web. An ontology can be described as a formal system that describes some particular field of interest from the viewpoint of the ontology user [20]. They are usually defined as a set of classes, concepts, properties, relationships, rules and restrictions.

A sample ontology continuing the previous example is depicted in figure 1.2. Here, it is learned that the resource “e:etu” is a person, that the anonymous object is

a thesis and the two other resources are topics. The relationships used are also present as instances of the class “owl:ObjectProperty”, and their possible domains and ranges defined.

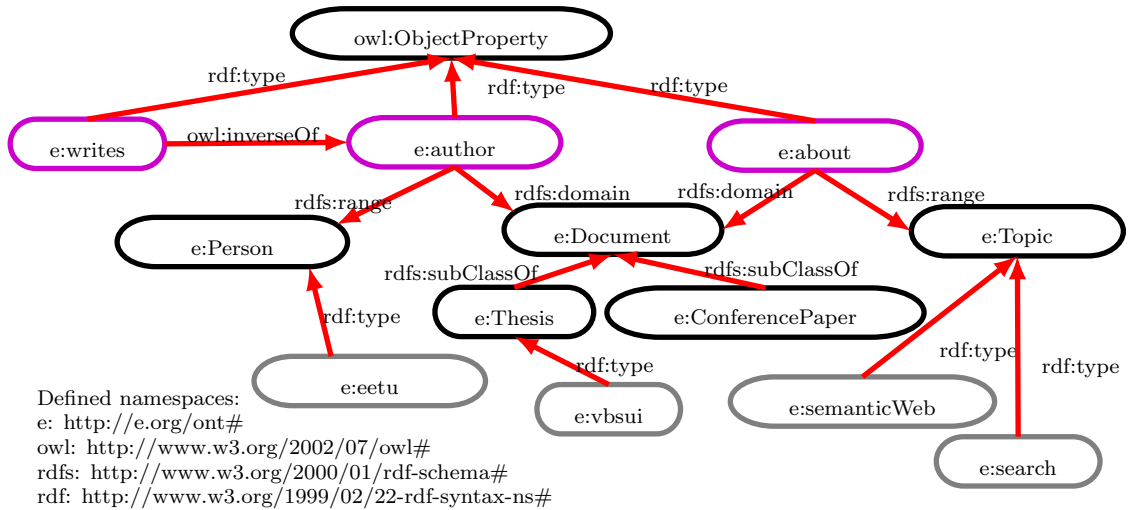


Figure 1.2: An example of an ontology

Also present, defined using the “rdfs:subClassOf” property, is a class subsumption hierarchy. Creating such a taxonomy is usually considered the first and most important step in ontology creation. This subsumption hierarchy also has semantic entailments defined in the underlying ontology language. For example, subclasses may inherit the various defined relationships of their superclasses. The “owl:inverseOf” property that has been defined between the properties “e:writes” and “e:author” can be used to do reasoning, too. Based on the existence of the property, the formal semantics of OWL define that for each triple of the form (X, “e:author”, Y), a triplet of the form (Y, “e:writes”, X) can be inferred.

With data stored in the RDF data model, and with ontologies adding formal deduction capabilities to that data, a semantic web is formed. This enables application designers to create intelligent applications more easily, as much of the intelligence needed is already encoded in the data.

1.2 Research Questions and Methodology

Retrospectively, the work described in this thesis follows the design science research methodology [29, 58], illustrated in figure 1.3. Thus, to best formalize the work in an analytical frame, the presentation of research questions and methodology here follows the outline depicted.

The work described in this thesis started from an objective-centered initiation. It stemmed from the needs of the FinnONTO project to create maximally reusable, adaptable and applicable components for a national Semantic Web infrastructure [37]. This resulted in the following objectives as research questions:

1. Seek a general user interface paradigm that:
 - (a) can be applied to as wide a variety of Semantic Web search and browsing tasks as possible.
 - (b) aligns well with Semantic Web technologies in the sense that it is easy to make maximal use of the semantics inherent in the data.
2. Identify supporting elements that make the paradigm more usable and adaptable.
3. Discover design guidelines that enable the adaptability of such systems in the context of the Semantic Web.

The methodology used was as follows. First, in order to better identify the problems, motivate research and gather theory, a survey of semantic search related research was conducted. This resulted in an understanding of the then current scope of supported semantic search and browsing behavior, as well as the conceptual capabilities of the systems surveyed. Information seeking behavior research was also consulted. This

resulted in an understanding of the breadth of possible user tasks and needs without bias to existing systems.

Based on the information gathered, hypotheses were formed that the user interface paradigm of view-based search would be able to:

1. cater to the breadth of user demands.
2. adapt to different kinds of data.
3. compete in conceptual capability with existing approaches.
4. align well with Semantic Web technologies.

Design science methodology is based on an iterative process of design, prototype building, demonstration and evaluation. Because the hypotheses stated here are mostly about adaptability, breadth and expressiveness, proving them requires that this be done in multiple contexts. Here, a multiple prototype approach was taken. User interfaces for tasks spanning different user needs were created and implemented as concrete systems. These interfaces and systems were then analyzed qualitatively and compared with respect to each other on:

1. How well the paradigm and system supported the task.
2. How hard it was to adapt the paradigm and system to the task.
3. How hard it was to adapt the paradigm and system to the data.

Qualitative and heuristic comparisons were chosen as methodology because formal testing in the scope needed was considered infeasible [29]. This is because of the following [55, 66]:

1. As regards usability testing, the functionality of a Semantic Web information system depends very much on the quality of the data, and it is very hard to separate data issues from user interface issues.
2. With regard to comparison between data sets, the same problem is evident. Different data sets on the Semantic Web differ from each other vastly in terms of quality, schema, content and inference capabilities. Thus any formal comparison of systems with regard to different data sets would by necessity target only a small subset of functionality.
3. Semantic Web information systems also differ from each other vastly in terms of scope, function and capability, so it is hard to find a baseline to compare to. In addition, most functionalities offered by Semantic Web systems are novel in the sense that their very existence is enabled by making use of semantic technologies. Thus, baseline systems also cannot be sought elsewhere. However, this also means that to prove added value, sometimes it is simply enough to demonstrate that something which was previously impossible now *can* be accomplished with a novel interface.

The lack of formal user interface or performance testing means that what is said of the usability or performance of individual interfaces rests mostly upon informed argument. For this study, this was deemed acceptable because of two reasons. First, the usability of the basic paradigm of view-based search is already well understood and proved [18, 25, 26, 61, 79, 80]. Second, the focus of this particular research is more on pure breadth of applicability – what *can* be done with the approach, as well as the iterative process of design science itself, which provides accumulating disciplinary and how to knowledge on *how* any certain task should be attempted with the methods at hand. In the case of the research presented here, these were particularly answers to the second and third research questions:

1. Identifying other user interfaces elements that could be integrated to support

the core view-based search paradigm.

2. Knowledge and comparisons on how different approaches to system and interface design affected adaptability.

1.3 Thesis Contributions

Seen as a whole, the major contributions of the works presented and discussed in this thesis are as follows:

- Identifying a strong synergy between the view-based search paradigm of information retrieval and:
 - the technological foundations of the Semantic Web (publication II)
 - forms of information retrieval on the Semantic Web (publications I and III)
- Aligning Semantic Web technologies and concepts to the paradigm in order to apply it (publications I, II, III and V)
- Furthering and tuning the paradigm for the Semantic Web with complementing user interface elements (publications I,III, IV and VII)
- Broadening the view-based search paradigm:
 - Domain-centric view-based search, which allows for more heterogeneous data (publication VII)
 - View-based constraining and visualization, which makes the paradigm more broadly applicable both to new data and to solving new problems (publication VII)

- Architectural design of easily adaptable view-based systems for the Semantic Web (publications I, II, IV and V)
- Testing the applicability and adaptability of both the paradigm and architectures (publications II,III,VI and VIII)
- The prototype systems themselves, particularly MuseumFinland² (publication II), which won the Semantic Web challenge award 2004 (second place) and the Finnish Prime Minister's commendation for the most technologically innovative application on the web 2004. The portal was also a jury nominated finalist in the Nordic digital excellence in museums awards, in the best Web based / Virtual application category. It, and its successor CultureSampo³ (publications VI and VIII) are still on the web, attracting tens of thousands of unique visitors every month.

1.4 Thesis Structure

This rest of this thesis summary is organized as follows. First, section 2 contains a survey and analysis of semantic search related research that resulted in focusing research on the view-based search paradigm.

Then follow the core contributions of this thesis. First, section 3 presents the view-based or faceted search paradigm and applies it to the Semantic Web. This paradigm is argued to both align well with core Semantic Web technologies, as well as be flexible enough to be used as a base for meeting a wide variety of user needs. Section 3.3 then presents the additional user interface element of semantic autocompletion that was created to round out view-based search for the Semantic Web. Section 3.4 draws the arguments together, and lists requirements for validating the hypothesis

²<http://www.museosuomi.fi/>

³<http://www.kulttuurisampo.fi/>

with real life tests.

The view-based search interfaces that were build to accomplish these tests are described and analyzed in section 4, while section 4.4 concerns itself with adaptability to different domains.

Section 4.5 discusses the problem of heterogeneous data with regard to view-based search, as well as our solutions.

Section 5 then deals with the implementation architectures created as part of this research, focusing on the technical adaptability of the methods developed.

Section 6 finally contains discussion on the benefits and limits of the view-based search approach. The thesis ends by listing conclusions.

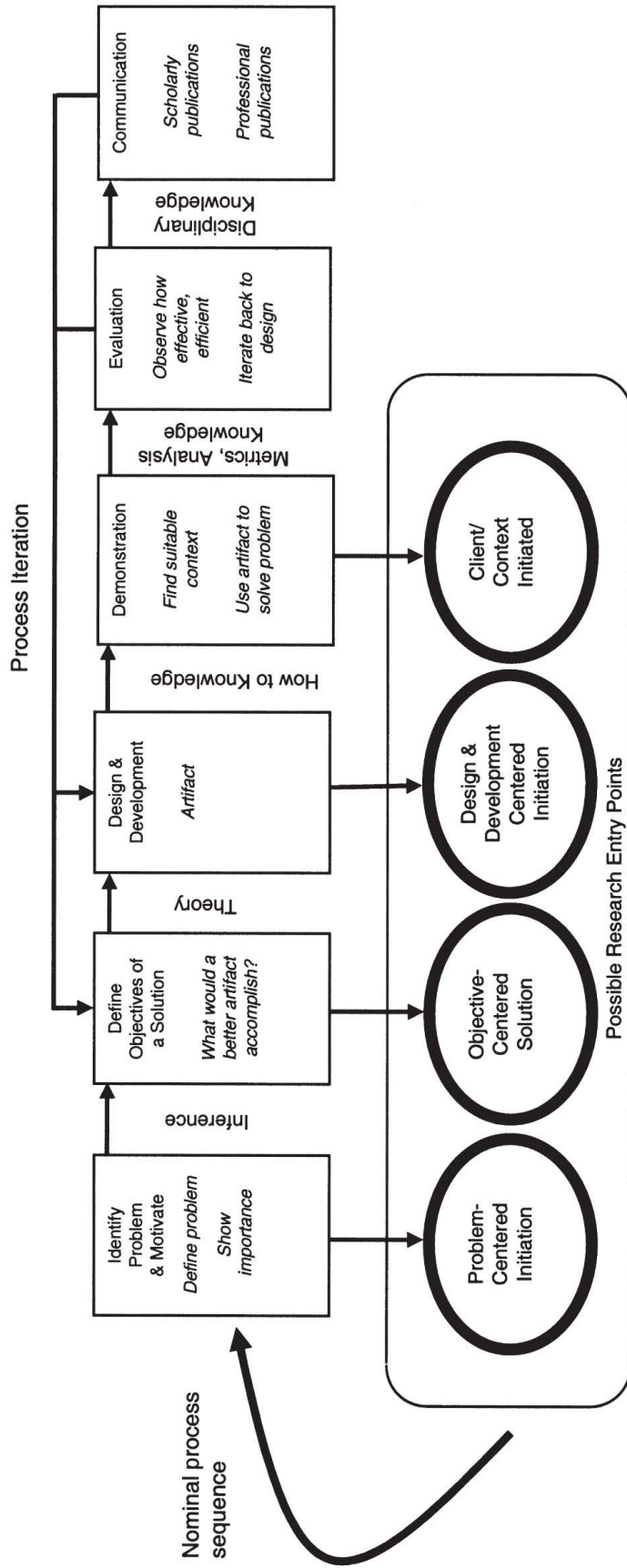


Figure 1.3: A process model of design science research methodology [58]

2 Survey of Semantic Search Research

This section of the thesis presents the results of a survey conducted in early 2005 to understand the challenges posed to information retrieval by the differences in format, breadth and depth of information on the Semantic Web as compared to the then current norm. Its function in this thesis is to provide understanding on the bases of the work, and thus has not been updated with recent publications.

For the survey, semantic search was defined as either search using semantic techniques, or search of formally annotated semantic content. The survey is based on reading and exploring some 25 different papers and approaches fitting that definition.

From the data gathered in the survey, some prevalent research directions in semantic search were identified, based on likeness of research goals. These, as well as the individual approaches that are part of them, are described in section 2.1. Besides research directions, the papers were also analyzed for common methodology. The methods used in a particular paper are noted when discussing it, but the descriptions of the common design patterns are presented in section 2.2.

2.1 Research Directions in Semantic Search

From the corpus of research used in the survey, five distinct research directions emerged. While the categories sometimes do not differ much in methodology, they seem separate and coherent enough on research goals to function as an informative clustering of the research space. The five directions are: augmenting traditional keyword search with semantic techniques, basic concept location, complex constraint queries, problem solving, and connecting path discovery. All of these are described

in detail in the following subsections.

2.1.1 Augmenting Traditional Keyword Search with Semantic Techniques

Much, particularly early research on Semantic Web enabled search deals with augmenting traditional text search with semantic techniques. This research direction differs significantly from the others presented later in the sense that it does not usually presume most of the knowledge being sought to be formally annotated. Instead, ontological techniques are used in a multitude of ways to augment keyword search, whether to increase recall or precision.

Many query expansion implementations used in keyword search make use of thesaurus ontology navigation as a step in query expansion. Particularly used is the large WordNet [19] ontology, defining synonym sets for words. The systems work as follows. First, keywords entered are located in an ontology. Then, various other concepts are located through graph traversal. Finally, the terms related to those concepts are used to either broaden or constrain the search. In Moldovan and Mihalcea [52] and Buscaldi et al. [10], terms are expanded to their synonym and meronym sets using the Boolean OR operations available in most search engines. In Clever Search [43], a particular meaning of a word in the WordNet ontology can be selected, resulting in the clarification text of that meaning being added to the search keywords with the Boolean AND operator. In the ontology navigation phase, the implementations differ mostly in what properties of the ontology are navigated and which terms are picked.

A simple manner of augmenting keyword search results is taken in the “Semantic Search” interface [23] of the TAP infrastructure. Here, besides a traditional keyword search targeted at a document database, the keywords are matched against concept labels in an RDF repository. Matching concepts are then returned alongside the

found documents. The paper also proposes a continuation of the search similar to Clever Search [43], where, if multiple concepts match the keyword, the user can select his intended meaning to constrain the search. Here, however, the idea is not to expand search terms, but to constrain results based on existing semantic annotations concerning them.

Rocha et al. [64] describes an algorithm for locating extra information relevant to a query given a starting set of documents. First, traditional text search is applied to a document collection. Then, a process of RDF graph traversal is begun from the annotations of those documents. The intent is to find concepts related to the result, such as the writer of the document or the project the document refers to in a general manner. The traversal is done by a spread activation algorithm, for the use of which the arcs in the ontology are weighed according to general interestingness. This interestingness measure is calculated by combining a specificity measure favoring unique connections in the knowledge base with a cluster measure, which favors links between similar concepts.

The CIRI [1] search system provides an ontological front-end to text search. The search is done through an ontology browser that visualizes the ontologies created for search as subsumption trees, from which concepts can be selected to constrain the search. The actual search is done through keywords annotated to these concepts as well as any subconcepts, using a traditional text search engine and Boolean logic. The search algorithm is in many ways similar to the query expansion algorithms discussed above. The main difference is in the user interface being based on direct ontological browsing, leaving out the first step of mapping a search keyword to the ontology.

2.1.2 Basic Concept Location

While much of semantic search research is directed at adding semantic annotations to data to improve search precision and recall on that data, there are other reasons for writing down information with formal semantics. Therefore, some research begins with assuming concepts, individuals and relationships, and deals with the task of efficiently finding instances of these core Semantic Web datatypes.

Usually, the data the user is interested in are individuals belonging to a class, but the domain knowledge and relationships are described mainly as class relationships in the ontology. This organization of data points to a natural way of locating information, represented for example in the SHOE [28] search system. In SHOE, the user is first given a visualization of the subsumption tree of classes in the ontology, from which he can choose the class of instances he is looking for. Then, the possible relationships or properties associated with the class are sought, and a form is presented that allows the user to constrain the set of instances by applying keyword filters to the various instance properties. When the properties point to objects, the target of the filtering will be the label of the referenced resource. Queries that can be expressed using this paradigm are for example “find all publications with a particular author name, from a particular project”. A similar approach is also taken in the ODESeW [16] portal tool.

A major drawback of the approach is that ontological knowledge is only used to produce a keyword form, and the user is still left to guess what keywords will result in the instances sought. This can be averted if the database is built in such a way that there are not too many items in a category, so they can be all shown for visual inspection. This approach is taken in many Internet directories such as the Open Directory Project directory⁴ and the Yahoo! directory⁵, where the editors are tasked

⁴<http://www.dmoz.org/>

⁵<http://dir.yahoo.com/>

with pruning the items and creating a branching category tree to hold them.

Once the search has advanced to the point where at least a single interesting instance is found, more information can be retrieved by browsing. The process is analogous to browsing web page hyperlinks. However, here the items shown are resources and the links between them are defined by their relations. In the simplest case, one concept is shown at a time, with its properties taken straight from the RDF triples. If a property points to another resource and not a literal, then clicking on that property will browse to the referenced concept. This is the approach taken for example in the SEAL portal tool [46].

The authors of the Haystack information management tool [40, 62] base their user interface paradigm almost completely on browsing from resource to resource. They argue this by search behavior research [73], concluding that most searching is done by means of a process called orienteering. The premise is that searchers usually don't actually themselves know or remember the specific qualities of what they are looking for, but have some idea of other things related to the sought item. The process of search is then a browsing experience in which the searcher looks for information resources that he knows are somehow related to the target. This continues iteratively, until enough additional information on the target resource has been found, and it can be located.

An example in Teevan et al. [73] is of a person searching for a particular piece of documentation. Not remembering where it is stored, she only remembers that it was referenced to in some e-mail message from a co-worker. She then scans through her mails in her inbox and, remembering the co-worker who the mail was from, finds the correct message and from there extracts the location of the document. To ease finding points of entry for orienteering, Haystack provides a simple text search interface, based on the rationale that the things people remember about resources are probably their labels or phrases contained in them.

2.1.3 Complex Constraint Queries

Many kinds of complex queries can be formulated as finding a group of objects of certain types connected by certain relationships. On the Semantic Web, this translates to graph patterns with constrained object node and property arc types. An example would be “Find all toys manufactured in Europe in the 19th century, used by someone born in the 20th century”. Here “toys”, “Europe”, “the 19th century”, “someone” and “the 20th century” are ontological class restrictions on nodes and “manufactured in”, “used by” and “time of birth” are the required connecting arcs in the pattern.

While such patterns are easy to formalize and query on the Semantic Web, they remain problematic because they are not easy for users to formulate. Therefore, much of the research in complex queries has been on user interfaces for creating complex query patterns as intuitively as possible.

Athanasios et al. [5] presents GRQL, a graphical user interface for building graph pattern queries based on navigating the ontology. First, a class in the ontology is selected as a starting point. All properties defined as applicable to the class in the ontology are then given for expansion. Clicking on a property expands the graph pattern to contain that property, and moves selection to the range class defined for that property. For example clicking the “creates” property in an “Artist” class creates the pattern “Artist \rightarrow creates \rightarrow Artifact”, and moves focus to the Artifact class, showing the properties for that class for further path expansion. The pattern can also be tightened to concern only some subclasses of a class, as in tightening the previous example to “Artist \rightarrow creates \rightarrow Painting or Sculpture”. In a similar way, property restriction definitions can be tightened into subproperties. More complex queries can be created by visiting a node created earlier and branching the expression there, creating patterns such as the one visually depicted in figure 2.1. This pattern could be used to find all artists that have either created any sculptures, or paintings

good enough to be exhibited at a museum, as well as those sculptures, paintings and museums.

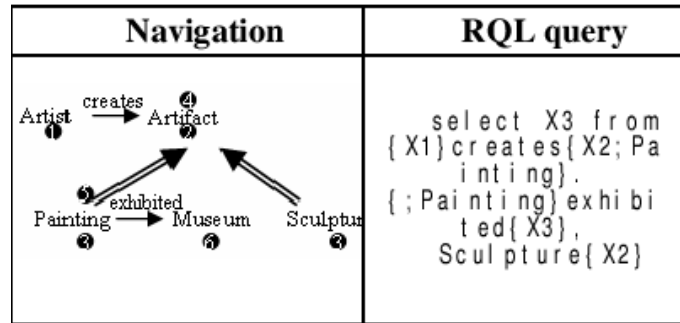


Figure 2.1: A visual formulation of a query in the GRQL interface, along with the generated query language expression [5]

Another graphical query generation interface is the SEWASIE visual tool for query formulation support [11]. Here, the user is given some prepared domain-specific patterns to choose from as a starting point, which they can then extend and customize. This is done through a clickable graphic visualization of the ontology neighborhood of the currently selected class, as shown in figure 2.2. The refinements to the query can be either additional property constraints to the classes, for example “Industry with sector Agriculture” or a replacement of a class in the pattern with another compatible one, such as a sub- or superclass.

All of the individual constraints in a complex semantic query need not be ontological. Zhang et al. [81] contains a method that allows one to treat keyword search terms as ontological classes whose instances have fuzzy membership values. A fuzzy logic formalism is then used to calculate relevance with respect to the entire query pattern formalized as a fuzzy logic statement.

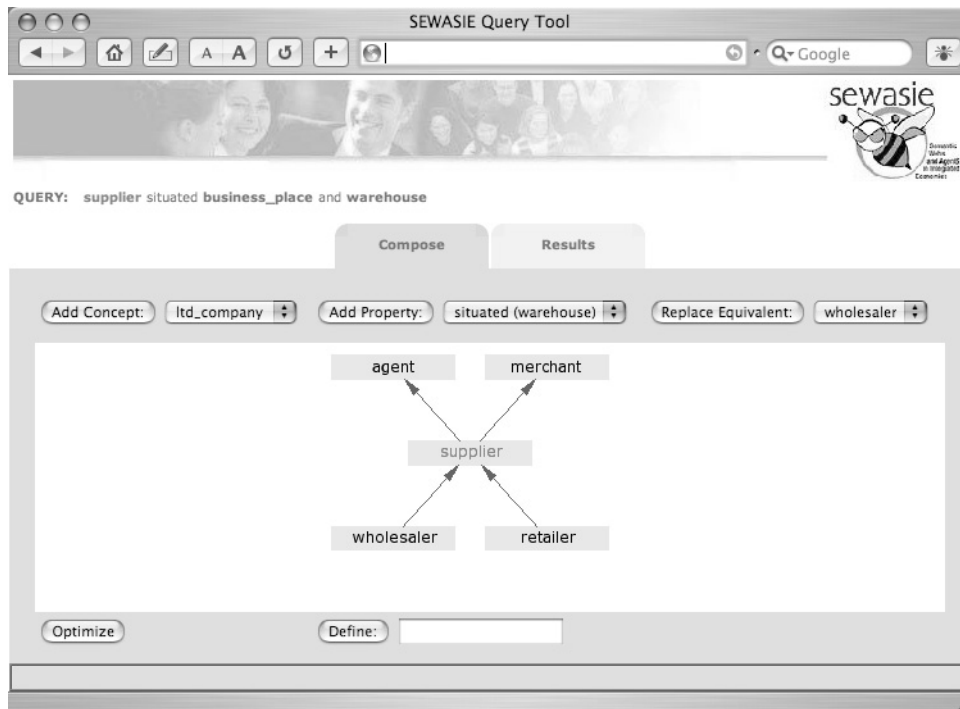


Figure 2.2: The SEWASIE visual tool for query formulation support [11]

2.1.4 Problem Solving

Describing a problem and searching for a solution by inferring one based on ontological knowledge is a use case often associated with the vision of the Semantic Web. However, current implementations are rare.

An example is the Wine Agent demonstration portal [32]. Here, the user enters information on the flavors in a dish, and the system infers a recommendation for a wine suitable to complement those flavors. The service is primarily based on restrictions and knowledge directly encoded in the OWL ontology of the portal. When a query comes in, a general purpose Description Logic reasoner is employed to perform constraint satisfaction on a combination of knowledge in the query and knowledge in the ontology. To encode the requisite knowledge in the query, the SQL-like query language OWL-QL [22] was developed.

2.1.5 Connecting Path Discovery

While usually property relations are used to traverse from an interesting resource to another, sometimes what is interesting are the connecting paths themselves. In the realized vision for the Semantic Web, a huge amount of varied semantic data will be available to be mined for semantic connections. An example of a domain where this could prove useful is the national security domain, where there is a need for finding, for example, emerging links between known terrorists and potential recruits [4].

A major problem here is how to define a measure of link interestingness in a way which cuts out uninteresting relations but is still general enough to be of use in finding complex, hidden relationships in the data. For example, “Company A and terrorist organization B are related because they both operate in the same country” is a conclusion, but not an interesting one. Anyanwu and Sheth [4] presents one take on the problem, attempting to draft an easily calculable general purpose requirement for interesting associations.

2.2 Common Methodology

In surveying semantic search related research some common methodologies appeared. Some are inherent to the RDF formalism and will probably be present in all Semantic Web applications, while others are more tied to the search domain. Identifying and understanding these common methods and how they are used in the various actual approaches provides valuable background for devising and evaluating new approaches, such as the view-based approach presented in this thesis.

2.2.1 RDF Path Traversal

Because the data model of RDF is a graph, where arcs and multiple arc paths encode information, it is natural to apply graph traversal in semantic search.

There were a couple of primary uses of network traversal found in this survey. One is finding more relevant information instances given a starting instance in the net, as in Rocha et al. [64]. Another use is in query formulation, such as in the GRQL [5] and SEWASIE [11] interfaces, where a query is constrained by navigating the classes and relationships.

Simple path traversal is also usually used when gathering all the information about an item for visualization. This is again because of the way the RDF data model works: information important to the user is also found in other resources linked to an information item, and not just the direct properties of that item. At least SEAL [46] and Semantic Search [23] both make use of graph patterns for gathering the information to be shown for an item.

2.2.2 Mapping Between Keywords and Concepts

Mapping between keywords and formal concepts is a common pattern appearing in semantic search. There are several reasons for its prevalence. The first is that commonly all knowledge available has not been formally encoded. Much research, such as the fuzzy keyword to concept mapping of Zhang et al. [81], is specifically about how to combine searching through textual material with search through formally defined information.

A second reason is that in many situations, natural language is the form of expression that comes most naturally to humans. Mapping patterns in the graph to sentences,

such as in the SEWASIE visual query tool [11] can give the user a clearer picture of what the relationships represent. On the other hand, the user may be more comfortable in expressing their queries as natural language sentences, as in the WordNet-based systems [10, 43, 52].

2.2.3 Graph Patterns

Whether described in RDF path or logical languages, graph patterns are an important concept in semantic search, used in multiple different roles. First, graph patterns are often used to formulate and encode complex constraint queries as discussed in section 2.1.3, specifying and locating interesting subgraphs in the RDF network. In Anyanwu and Sheth [4], general RDF patterns were also used to find interesting connecting paths between named resources. In result visualization, the specifications on where to fetch information relevant to the item are also usually given as graph patterns.

2.2.4 Logics

Logics and inference are integrally tied to the larger vision of the Semantic Web. For example, the web ontology language standard OWL [50] is based on Description Logics. However, only few applications are currently built solely on top of advanced logical frameworks, with the Wine Agent [32] being an exception rather than a common example. Much more commonly, applications make use of a few particular entailments as a base, and build their own functionality on top of that. For example SHOE [28], ODESeW [16], GRQL [5] and SEWASIE [11] all make use of the transitive subClassOf hierarchy, and some also the properties conferred to a class by that hierarchy.

2.2.5 Combining Uncertainty with Logics

In the research direction of augmenting text search with ontology techniques, there is a need for formalisms which allow combining uncertain annotations based on text search with the firmness of semantic annotations. As a result, several formalizations for, and experiments with fuzzy or probabilistic logics, relations and fuzzy concepts have been undertaken in that field. The method described in Zhang et al. [81] is an example.

Fuzzy logics are, however, not only useful in combining text search with ontologies. On the search method research side not directly tied to actual applications, Singh et al. [70] applies fuzzy qualifiers to complex constraint queries. In Parry [56], the idea is presented that user profiling could be used as a basis for weighting the interestingness of an ontological relation to be used in the search. In Kauppinen and Hyvönen [41], a basis is depicted for calculating overlap values for historical and current geographic places, for use in a probabilistic mapping of the concepts to one another in any ontological search.

2.3 Conclusions Drawn from the Survey

There are many common patterns found in the approaches described in this survey. On the technique level, it seems that many of the methods used are general and separable. They could probably be used in most of the systems, regardless of research direction or application domain.

It also seems that some of the research directions can be combined. First, simple concept location can be seen as a forerunner and subset of the interfaces allowing selection by more complex graph patterns. Second, while the current interfaces for

creating graph query patterns concern fairly simple patterns where the individuals and classes are the interesting information items, there is no theoretical reason for such a limitation. Because relations appear as equal partners in the underlying data model, querying for them would only need a shift in focus on the query formulation user interface level. Fuzzy logic formalisms and fuzzy concepts would allow for the inclusion of keyword search results in the queries. After finding a result set using complex constraints, graph traversal algorithms could be applied to find additional result items.

The only direction that does not neatly wrap into the others is pure inference-based problem solving. However, as already stated, many of the applications do make use of the logical entailments in one form or another, they only do not rely on them completely.

3 Applying View-Based Concepts to the Semantic Web

Based on the conclusions drawn above, it seems that complex graph matching patterns form a useful, extensible technology core for semantic search. However, a major challenge in using it is in how to provide the end-user with an intuitive interface for creating graph-based queries. This thesis is based on the argument that the so-called view-based, or faceted search paradigm [60] provides a suitable basis for creating such interfaces. In the following, this core paradigm is explained. The presentation given here expands on the short overview given in publication III. This is done in order to more fully ground and argue the research presented in this thesis.

3.1 The View-Based Search Paradigm

The core idea of view-based search is to provide multiple, simultaneous views to an information collection, each showing the collection categorized according to some distinct aspect. This is based upon a long-running library tradition of faceted classification [48]. A search in the system then proceeds by selecting subsets of values from the views, constraining the search based on the aspects selected.

The paradigm was first developed into a computer application in the HiBrowse [60] system for searching through large collections of medical texts. Figure 3.1 depicts the interface of HiBrowse as an example of what view-based search can look like for an end-user. Shown are three views, each categorizing health articles in the system according to a particular dimension. Alongside the category names are always placed the number of articles that relate to that category, so the user always knows beforehand how a particular choice will constrain the result set. The three

views in the example are: 1) the anatomy view, showing a hierarchical categorization of diseases based on the part of human anatomy they affect, 2) the therapy view, which organizes the material based on type of therapy described, and 3) the groups view, which allows for searching by affected patient group. Because these viewpoints are so vastly different, making choices from them intersects the data very efficiently, leading to quickly finding items of relevance. Also, showing all possible choices beforehand supports the user at each point in their query, as well as quietly adds to his understanding of the structure and indexing of the whole data set.

After HiBrowse, the idea of view-based search has been implemented in a number of systems. Usability studies done on these systems, such as Flamenco [18, 26, 79] and Relation Browser++ [80] have proved the usability claims made. The paradigm was proved both powerful and intuitive for end-users, particularly in drafting more complex queries. More evidence suggesting the power of the paradigm comes from more general results on the benefit of using multiple categorizations in search [25, 61].

Traditionally in view-based search systems, the views used are either flat or hierarchical tree categorizations of the search items. There are several good reasons for using such views. First, such categorizations are familiar to users, from for example library classification systems. Second, they can often be drawn up from any aspect of a collection, which allows for a uniform look and feel for the views. In this thesis, one reason for favoring tree categorizations also relates to how the paradigm is combined with the Semantic Web ontological hierarchies, described later.

Figure 3.2 shows a conceptual overview and an example of view-based querying using hierarchical categorizations. Here, on the left, the data representing a museum collection of items has been categorized according to three hierarchical views: “Location of Manufacture”, “Item Type” and “Location of Use”. The idea of view-based search, then, is that given these views, the user can apply successive constraints on any of the views in any order, with the effects of filtering immediately shown

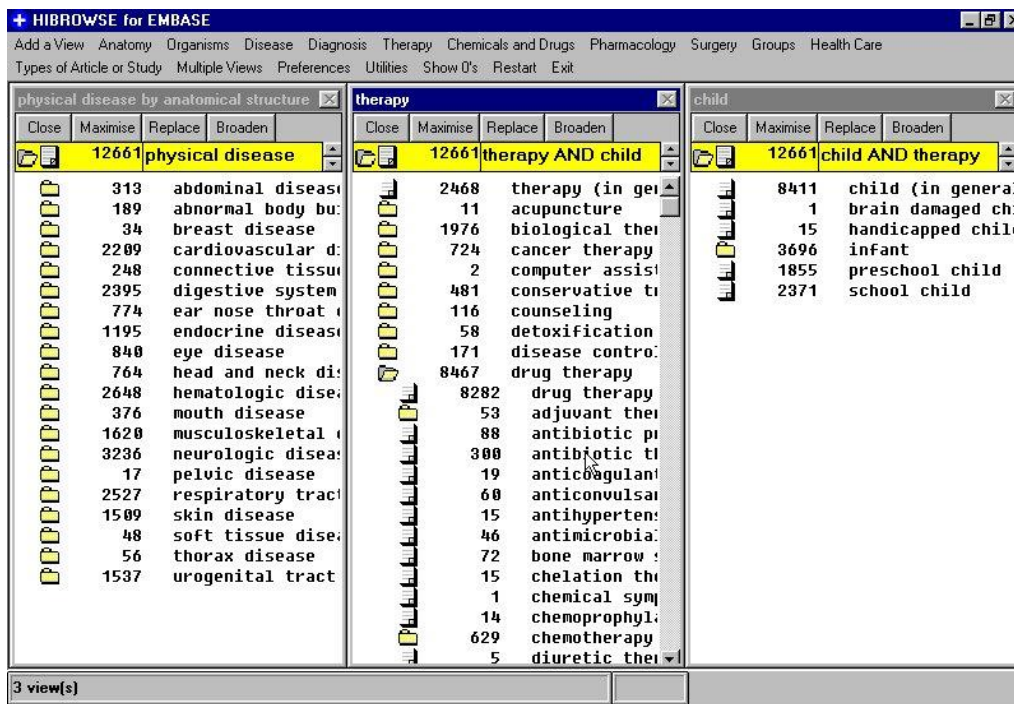


Figure 3.1: The HiBrowse interface, with three hierarchical views [60]

in all the views. Simultaneous constraints in different views are applied by simply performing an intersection operation on the results of the constraints in each view. In the example of figure 3.2, the user has selected as query constraints the category “Office Equipment” from the “Item Type” facet, and the category “Finland” from the “Location of Use” facet. Intuitively, the user is searching for any office equipment in the collection that happens to have been used anywhere in Finland.

Inside a hierarchical view, the constraint is calculated as follows. When the user selects a category c in a view v , the system constrains the search by leaving in the result set only such objects that are annotated in view v with some subcategory of c or c itself. In the figure, this is typified in the “Location of Use” view. Here, none of the objects are directly annotated as belonging to the category Finland, but some are nonetheless taken as matching, based on the implicit knowledge in the category hierarchy that Lahti and Helsinki are located in Finland.

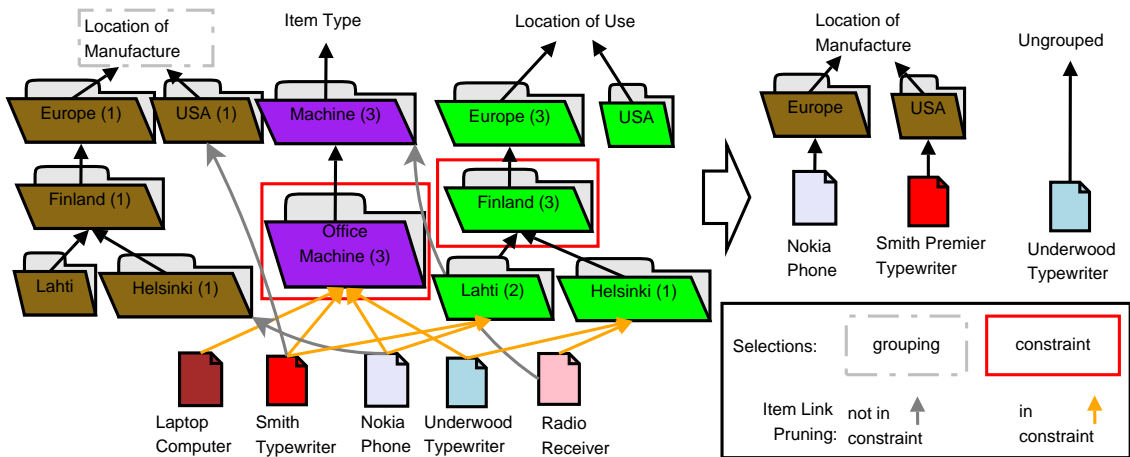


Figure 3.2: A conceptual overview of view-based querying

A core idea of view-based search is that once the result set is calculated, it is categorized according to the views and visualized in place. This can be done for example by showing the number of results in each view category beside them, as in figure 3.2 and the HiBrowse interface in figure 3.1. The result of applying this idea is a tight, beneficial loop between query constraining and result browsing. First, the user is immediately able to gauge the result set from multiple different aspects. Second, the user is given direct, accurate information on how any further selections will limit the result set. The system can also directly cut out category choices with no associated results as further selections, because selecting them would lead to an empty result set.

In addition to in place visualization, separate views can be used for organizing the results of a search. For example, on the right in figure 3.2, a flat column result grouping has been formed using the “Location of Manufacture” category tree. This has been accomplished by cutting the hierarchy on the first sublevel, and sorting the result items into these categories. Item “Nokia Phone” is bumped two levels up to its ancestor category of “Europe”, and item “Underwood Typewriter”, which was not annotated anywhere within the grouping hierarchy, is shown within the dynamically

created “un-grouped” category.

3.2 View Projection from Ontologies

In non-semantic view-based search systems, the focus on hierarchical views was brought by the prevalence of taxonomic classification systems in the collections the systems were built for. On the Semantic Web, domains are described more richly using ontologies. However, hierarchical hyponymy and meronymy relationships are still important for structuring a domain. Therefore, the ontologies used typically contain a rich variety of such elements, most often defined with explicit relations, such as “partOf” and “subclassOf”. This naturally leads to the idea of using these hierarchical structures as bases for views in view-based searching. To carry this out, this section introduces a process termed view projection. Here the process is explained in abstract terms. Details of the actual systems produced are found later, in the implementation part of this thesis.

An example of view projection using the process is given in figure 3.3. The transformation described consists of two important parts: projecting a view tree from the graph, and linking items to the categories projected. The projection of a hierarchical category tree can be done through traversing the graph by some rule, picking up relevant concepts and linking them into a tree based on the relations they have in the underlying knowledge base. Most commonly, the relations used are hyponymies and different kinds of meronymies.

In the example, the “Item Type” view is projected using a simple rule following the “subclassOf” hyponymy relationship, starting from a pair of selected roots. The rules governing projecting the “Location” meronymy tree are a little more complex. It is created by taking all instances of the class “GeographicalEntity” and its subclasses, but then creating a category tree from these instances by traversing their

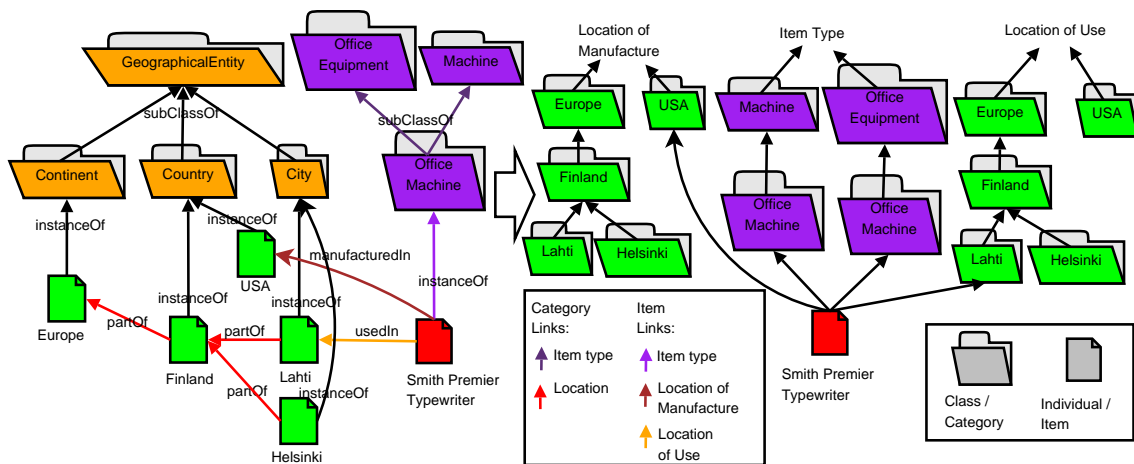


Figure 3.3: An example of view projection

“partOf” relationships.

In projecting a tree from a directed graph, there are always two things that must be considered. First, possible loops in the source data must be dealt with to produce a Directed Acyclic Graph (DAG). This usually means just dismissing arcs that would form cycles in the projection process. Second, classes with multiple superclasses must be dealt with to project the DAG into a tree. Usually such classes are either assigned to a single superclass or cloned, which results in cloning also the whole subtree below. In the example, the class “Office Machine”, in the “Item Type” view is cloned based on this rule.

The second phase of view projection is associating the actual information items with the categories. Most often, this is just a simple case of selecting a property that links the items to the categories, but it can get more complex than that here, too. As can be seen from the example in figure 3.3, the same hierarchy can also form the basis of several views, based on how linked items are selected. The geographical “partOf” hierarchy is projected into two views, based on whether the “usedIn” or the “manufacturedIn” relationship between the items and places is used. For an example

where the item linking would be more complex, consider a view categorizing items based on the type of geographical entity they were manufactured in. Here, creating the view hierarchy would be a simple case of transitively following the “subclassOf” property of the class “GeographicalEntity”. However, both a “manufacturedIn” and an “instanceOf” property would have to be traversed to link the items to the categories.

3.3 Complementing View-Based Search with Semantic Auto-completion

View-based search is based on providing visual categorizations of data from different viewpoints. This gives the user excellent contextual information for a drill-down search, where a user does not a priori know either exactly what they are searching for or do not know the collection sufficiently well to formulate efficient queries.

However, when the user does have sufficient information, the usability of the view-based paradigm benefits from applying complementary elements to support such spot search. During the work presented in this thesis, the principle of semantic auto-completion was developed for this purpose, and its combination with view-based search studied.

The different forms of semantic auto-completion developed are presented exhaustively in paper IV. Shortly, the idea of auto-completion is that a user can type in short prefix strings, for which the system then returns possible completions, thus aiding query construction. The idea of semantic auto-completion then is to extend traditional syntactic auto-completion to take into account semantic information.

For example, syntactic auto-completion for the prefix strings “Scand presid” might

return keywords Scandinavian and president, but this would not aid the user if most of the data used the keyword “Nordic” instead of “Scandinavian” or only had data on the presidents of Finland, Norway, Denmark and Sweden without explicitly mentioning Scandinavia.

With semantic autocompletion, the idea is that both the terms Nordic and Scandinavian could be linked to the same underlying annotation concept, and furthermore the system could make use of ontological information linking the countries to the whole. It could also suggest the ontologically more general “head of state” keyword in order to bring into the results the leaders of those Scandinavian countries with royal lineages. It might also span languages, e.g. matching also “Suomen presidentti”, the president of Finland in Finnish.

Semantic autocompletion can also offer other further means of constraining the query beyond keywords, such as giving a selection of the possible roles in which the keyword can appear, such as offering a choice between “place of use” and “place of manufacture” for the keyword “Finland” in relation to museum objects.

Because these semantic extensions can be much larger than syntactic extensions, it is beneficial to pre-filter the results by counting actual search hits corresponding to each extension, as in view-based search.

In order to maintain as much of the context advantages of view-based search, it is beneficial to provide enough ontological or view context for the autocompletions (e.g. that a particular hit count is specifically for “place of use: Nokia, a part of Finland”). One possibility is to visualize the matching concepts directly in the views, an approach described both in paper IV as well as later in this thesis.

Another possibility is to gather enough context information around the results themselves, thus creating an additional dynamic view to the data to complement the

static views decided by the system developers. Simple implementations of also this approach are described in publication IV as well as later here. However, this has also been a topic of further study, which resulted in a solution for providing contextual navigation for autocompleted in-place developed [71].

3.4 View-Based Search as a General Base for Semantic Interfaces

The previous sections showed a way of combining view-based search with the Semantic Web. However, there are still other requirements to be met before the paradigm can be considered useful as a general base for semantic search interfaces.

First, and most importantly, the interfaces created using the paradigm should be usable by an end-user for the tasks they need to perform on the Semantic Web. Usability studies [18, 26, 79] suggested that the paradigm is particularly useful for intuitively formulating complex queries. This, combined with the conclusions about complex queries forming a good technology core for semantic search intimate good results. However, the expressiveness of the paradigm still needs to be discussed.

View-based constraints can be seen as a limited form of complex graph constraints. At first sight, the formalism may seem restrictive compared with the more complex graph patterns formed by the interfaces presented in section 2.1.3 of the survey section. Widening the expressive power of the approach, however, is the fact that the views can be complex projections from rich ontologies. It seems that most combinatorial constraints needed can be covered by choosing the views intelligently. The difference becomes that in view-based search, much work must be done in figuring out the useful views and projecting them from the underlying ontology. However, a similar operation will probably prove necessary for the other formalisms

as well, as already apparent for example in the preselected starting point queries of the SEWASIE [11] system.

Concerning projection, the formalism should be tested on adaptability to a wide range of different ontological data. It should also be easy to extend the paradigm itself to make powerful use of the rich semantics of that data. There are few inherent restrictions here. The only real requirement of a view is that it organizes the information items of the application in some intuitive, visualizable, and constrainable way. Therefore, it should be quite possible to extend the paradigm to make use of other supporting semantic search methods.

4 Adaptability of Semantic View-Based Interfaces

While the above considerations point to a good potential for view-based search on the Semantic Web, the hypotheses still need real world verification. Combining all the requirements, the paradigm should make it possible to create powerful, efficient interfaces for varying search tasks aimed at real world ontological data.

In order to test the applicability of the paradigm to varying search tasks, search behavior research [6, 12, 14, 15, 33, 68, 73, 78] was consulted to discover prototypical information retrieval tasks and strategies.

As a first measure, the various search strategies identified in research were partitioned into two groups, designed to demarcate two different polar ends of search behavior. By designing user interfaces for these disparate objectives, much information can be gained of the applicability of the paradigm. These groups were respectively termed browsing and spot search.

The browsing agglomerate search strategy is characterized by the absence of a particular clear information need. Instead, the user is either looking to get an overview of some topic, or just looking for something interesting to explore. This agglomeration contains the information gathering and browsing strategies identified in Sellen et al. [68], the scanning, learning and recognizing strategies in Belkin et al. [6], as well as the informal search and undirected and directed viewing strategies in Choo et al. [12].

Spot searching, the second agglomerate strategy defined, relates closely to the finding behavior of Sellen et al. [68], the formal search of Choo et al. [12], as well as the teleporting strategy defined in Teevan et al. [73]. It also closely corresponds to the search, select and specify strategies of Belkin et al. [6]. It is characterized by

the need for a particular, singular piece of information without much regard to its context, and by the need to get it quickly.

It is argued here that the view-based search paradigm can adequately respond to both of these, in many cases opposite needs of searching and browsing. Additionally, there is value in being able to support them both at the same time. This is proved by the results of research into the prevalence of the orienteering search behavior, where different strategies are used intermittently [73] as well as the fact that different complete information seeking strategies may actually pick component strategies from both agglomerates [6, 15, 78]. Here the tight relationship between result browsing and query constraining in view-based search is an asset.

4.1 A View-Based Search Interface for Browsing

First, a view-based interface intended primarily for browsing was created for the MuseumFinland portal. This interface is described in detail in publications II and I, as well as shortly in III. However, because the interfaces developed are at the core of this thesis, the parts of the interface description most pertinent to the argument are repeated here.


The MuseumFinland portal is intended as a prototype virtual museum semantically combining museum artifact collections from different sources. Taking this into account, most users' information needs when coming into the portal will not be well defined. Instead, the most common use case will be to first ascertain if there is any interesting content in the collections, and if found, scan them, possibly finding other interesting items in the process. Thus, a browsing-oriented interface is appropriate. For our user interface design, after some iterations [35] we eventually settled on view-based interface similar to the Flamenco system for locating fine arts images [18, 26, 79], which had scored extremely well in user interface studies.

The main search view of MuseumFinland is depicted in figure 4.1. The design follows an iteratively advancing search paradigm, aiming to provide as many informative choices to the user as possible at each point in browsing. In the interface, the main selection views are displayed on the left. They each contain a flat list of selections, initially showing the root concepts of each hierarchical view, along with hit counts that tell how many results will be left if the user selects a particular constraint. On the right, items related to the current constraints are shown, by default organized according to the subcategories of the last selection. In this way, as many different further constraints as possible fit on the screen, as well as many different types of result items as possible. At each level, the user only needs to find one further interesting constraint to continue her search, or one interesting item to move into the item browsing part of the interface.

At all times, the user can firmly gauge the effects of possible choices by looking at the number of hits associated with the categories, and the user interface eliminates selections leading to empty result sets completely. This interaction pattern also quickly gives the user an impression on what is contained in the portal collection, and provides to the user in each step a manageable set of choices to choose from. For example, looking at the main page of MuseumFinland, the user, not really looking for anything particular, may decide that he will start by looking at items used in Europe. In the results, he then sees several chairs he likes, and decides to constrain his search to furnishing items used in Europe, and so on.


As said, the views show by default only a flat list of the root concepts of each hierarchical view. But when a user selects one of these (e.g. “tools” in the item type view), the content of that view changes to show the subcategories of his selection as further constraint possibilities (e.g. “textile tools”, “forestry tools”, “writing implements”, and so on). In this way, the user can iteratively drill down their constraints also in a single view until they are happy with the scope of objects shown.

Address <http://museosuomi.cs.helsinki.fi/?l=f&m=0&n=%2500%2516&g=c%2500%2516>



MuseoSuomi

- Suomen museot semanttisessa webissä -



Uusi haku | Ohjeet | Näytä kaikki kategoriat | Tietoa ohjelmasta | MuseoSuomi-palaute

Sanahaku: Hae tarkenna hakua

Esinetyyppi [kaikki](#) > [työvälineet \(koko luokittelu\)](#)

- [tekstiilikasityövälineet](#) (219),
- [kansanlaakinnan työvälineet](#) (1),
- [luokittelemattomat työvälineet](#) (36),
- [maaloustyövälineet](#) (7), [metallityövälineet](#) (1),
- [pilkkomis ja hienontamisvälineet](#) (4),
- [kirjoitusvälineet](#) (9), [metsätyövälineet](#) (4),
- [työkahvit](#) (22)

Materiaali [\(koko luokittelu\)](#) [\(ryhmittele kohteet\)](#)

[materiaalit](#) (241)

Valmistaja [\(koko luokittelu\)](#) [\(ryhmittele kohteet\)](#)

[henkilöt](#) (9), [tuotemerkit](#) (2),

[yritykset](#) (38)

Valmistuspaikka [\(koko luokittelu\)](#) [\(ryhmittele kohteet\)](#)

[Afrikka](#) (2), [Etela-Amerikka](#) (1),

[Eurooppa](#) (84)

Valmistusaika [\(koko luokittelu\)](#) [\(ryhmittele kohteet\)](#)

[aikakaudet](#) (90), [vuosisadat](#) (89)

Käyttäjä [\(koko luokittelu\)](#) [\(ryhmittele kohteet\)](#)

[henkilöt](#) (54), [laitokset](#) (1),

[yritykset](#) (3)

Käyttöpaikka [\(koko luokittelu\)](#) [\(ryhmittele kohteet\)](#)

[Eurooppa](#) (71)

Käyttötilanne [\(koko luokittelu\)](#) [\(ryhmittele kohteet\)](#)

[harrastus- ja kansalaistoiminta](#) (4),

[kohteelle tehtävät toimenpiteet](#) (17),

[maalatous ja karjanhoito](#) (2),

[ruoan- ja juomanvalmistus](#) (3),

[toimijoiden yleiset prosessit](#) (2),

[elinkeinot](#) (9), [valmistusteknikat](#) (179)

Kokoelma [\(koko luokittelu\)](#) [\(ryhmittele kohteet\)](#)

[Espoon kaupunginmuseon kokoelmat](#) (54),

[Kansallismuseon kokoelmat](#) (193),





[Lahden kaupunginmuseon kokoelmat](#) (50)

Hakuehdot


Kategoria: Esinetyyppi > [työvälineet](#) [\(ryhmittele kohteet\)](#) [\(poista\)](#)

Kohteet ryhmiteltyinä kategorian [työvälineet](#) mukaisesti
(näytä ilman ryhmittelyä)

[tekstiilikasityövälineet](#), kohteet 1-4/219 [\(ryhmittele kohteet\)](#)

			
kehräpuu, kuosali (NBA SU4527 50)	kehrulauta, kehräpuu, kuezsel, kuosali (NBA SU5069 26)	rukinlapa (ECM 100 1)	sneldde, varttinänlumppio, varttinäpyörä (NBA SU2449 7)

[kansanlaakinnan työvälineet](#), kohteet 1-1/1 [\(ryhmittele kohteet\)](#)



suonrauta:suoneniskentärauta (ECM 2711 1) (edellinen) / (seuraava)

[luokittelemattomat työvälineet](#), kohteet 1-4/36 [\(ryhmittele kohteet\)](#)





			
nappikoukku:näpätuskoukku (ECM 3594 264)	kietkamläbda, komsiolihna (NBA SU4922 32)	palohosat:palohosat (ECM 614 1)	luontilasta (NBA SU4135 166)

Figure 4.1: The main search view of MuseumFinland

Showing only one flat level of each hierarchical grouping supports the interaction pattern wanted. However, sometimes this limits the overview gained in a harmful way for answering questions about the result set as a whole. The user interface of MuseumFinland therefore also provides an alternate view to the material and the facets of the application. Clicking the link “whole facet” (“koko luokittelu”) on any facet brings up a tree view of the whole facet with the number of items in each category calculated according to current constraints. This tree view gives the user

an overview of the distribution of items in the result over a wished dimension. By judicious use of this view, complex questions about the result set can be answered. For example, a collection manager may want to know how well their collection covers tools manufactured at different times. For this, she can select the “Time of manufacture” whole facet view after constraining the query as described before. The resulting display is shown in figure 4.2. From the result and the visual cues, such as graying out categories with no hits, it is easy to see several things. For example, while there is a balance in items relating to the two world wars (“I maailmasota” with 11 items and “II maailmansota” with 9 items), there are no items from 1700–1749 and only one from 1750–1799. Also, there are two items that could only be reliably dated as being manufactured at some point in the 18th century, explaining the total of 3 items for the category “1700-luku”.

To balance the scales, and support quick spot searching when the user knows what he is looking for, MuseumFinland includes semantic keyword searching functionality. This functionality is seamlessly integrated with view-based search in the following way: First, the search keywords are matched against category names in the facets as well as text fields in the metadata. Then, a new dynamic view is created in the user interface. This view contains all categories whose name or other defined property value matches the keyword. Intuitively these categories tell the different interpretations of the keyword, and by selecting one of them a semantically disambiguated choice can be made. This also solves the search problem of finding relevant categories in views that contain thousands of categories. The view in figure 4.3 includes a keyword search view for the word “nokia”. Matched are, for example, the categories Nokia (the telephone company), Nokia (the place) and Nokia-Mobira (an earlier incarnation of the telephone company). A result set of object hits is also shown. This result set contains all objects contained in any of the categories matched as well as all objects whose metadata directly contains the keyword. The hits are grouped by the categories found.



Figure 4.2: The tree view of MuseumFinland

At any point during the view-based search the user can select any hit found by clicking on its image. This moves the user interface into the individual item view, and a mode of browsing the results complementary to view-based search. The individual item view is shown in figure 4.4. The example depicts a special part, a distaff (“rukinlapa” in Finnish) used in a spinning wheel. On the left and center of



Figure 4.3: Entering keywords creates a dynamic facet of matching categories

the view are the detailed metadata about the item stored in the database. At the bottom center, the views are again shown, this time in an inverted form, showing all the hierarchy paths to the current item. Clicking on any category here starts a new search for items referring to just that particular category. The idea is that once a user has found an item interesting in some respect in the virtual museum exhibition, they can easily find others like it in that same regard.

This loop back to the search view is however not the only way in which the portal supports browsing based on an interesting item as a starting point. On the right of the item view there is a collection of semantic links coupling other items directly to the one currently viewed. These allow for lateral direct browsing between items in the portal database as a complementary means of navigation. The idea here is that the view-based search can also be seen only as the starting point for finding one interesting item. The rest of the user experience can consist of “wandering the museum halls” from an object to another related one.

The semantic browsing component of the view is organized as follows: First, a heading is shown describing the rule linking the items together. Then, a subheading

MuseoSuomi
- Suomen museot semanttisessa webissä -

Uusi haku | Takaisin hakusivulle | Ohjeet | Tietoa ohjelmasta | MuseoSuomi-palautte
Espoo (180) << | Bemöle (14) | >> Järvenperä (9)
(←) ruginlapa (←) jämsivuolin

rukinlapa



Valmistuspaikka: Suomi
Valmistusaika: 1793
Käyttöpäikka: Suomi,Bemöle,Espoo,Suomi,Vanhakartano,Espoo,Suomi
Asiasana: KEHRUU, KORISTEVEISTO, PUUMERKKI, VUOSILUKU
Museokokoelma: Museokokoelma
Vastuumuseo: Espoon kaupunginmuseo
Asiasanasto: Espoon kaupunginmuseon sanasto
Esineen numero: ECM.100.1
ID: 1001

Esimetyyppi:

- työvalineet (299) > tekstikkasivovalineet (219) > kehruun ja langanvalmistuksen työvalineet (63) > kehruvalineet (59) > kuontalonpitimet (3) > **rukinlapat** (1)

Valmistuspaikka:

- Eurooppa (2541) > **Suomi** (2239)

Valmistusaika:

- ajakaudet (3024) > historiallinen aika (3029) > **usi aika** (3013)
- vuosadat (3012) > **1700-luku** (123)

Käyttöpäikka:

- Eurooppa (2232) > **Suomi** (2227)
- Eurooppa (2232) > Suomi (2227) > Etelä-Suomen lään (1999) > Uusimaa-Nyland (670) > Espoo (512)
- Eurooppa (2232) > Suomi (2227) > Etelä-Suomen lään (1999) > Uusimaa-Nyland (670) > Espoo (512) > Bemöle (14)

Käyttötilanne:

- valmistustekniikat (1587) > tekniikan työ (39) > veisto (32) > **koristeveisto** (8)
- valmistustekniikat (1587) > tekstilityö (886) > kuuntely (74) > **kehruu** (64)

Kokoelma:

- Espoon kaupunginmuseon kokoelmat (1190) > **Museokokoelma** (1129)

Samaa käyttöpaikka

Bemöle:

- jämsivuolin
- opetusvaline peli
- opetusvaline peli
- opetusvaline peli
- opetusvaline peli

Espoo:

- kuvalakia kuvakirja kangasta
- lammilaisen hyyhähainen lenkki
- neuletakki naisen neuletakki
- hartiavaate naisen pitsinen hartiavaate
- purun yläosa, jalkineen purun yläosa

Suomi:

- ruokalina ruokalina, damaari
- katalina katalina, etupistokirjontaa
- pöytälina pöytälina, kirjoitu
- pöytälina ristipistokirjontainen pöytälina
- katalina batistilina, kirjoitu

Esiineseen liittyvään paikkaan liittyviä muinaismuistoja

Espoo:

- Rovkiot
- Puolustusarvukset
- Rovkiot
- Rovkiot

Samaan aiheeseen liittyviä esineitä

ajan_kasitteet:

- hevoslomi
- arkkivaatearkku
- takki vanupeite
- veistos pienoisveistos
- pezukarttu kunkka

kehruu:

- jakkara kehrusjakkara
- rullatuoli rullateline
- puola langapuola
- puola langanalla
- loukkupellavaloukku

Figure 4.4: The item view of MuseumFinland

shows a semantic property or properties of the current item that are shared with other items in the collection. These items with common elements are then shown as the actual links.

While most of the link groups are based on the same categorization used in the view-based search, such as “Common location of use” (“Samaa käyttöpaikka”), some rules go beyond the view definitions to capture other complex associations between the items. For example in figure 4.4, under the heading “Items related to the same topic” (“Samaan aiheeseen liittyviä esineitä”), there are other items related to “Concepts of time” (“ajan_kasitteet”). This is possible despite the fact that the views do not contain any “Concepts of time”. In the underlying metadata RDF graph it has been

annotated that the distaff has a year carved into it, and thus it can be found in the rule doing the semantic linking.

Comparing the interaction patterns of the MuseumFinland virtual exhibition with the physical experience of visiting a museum provides further perspective on the interface. The view-based search can be equated with choosing or building a physical museum exhibition dynamically by selecting items dealing with a certain topic or a combination of topics. The semantic browsing from item to item and item group to item group can be equated with wandering between the exhibits in the exhibition selected. However, here one is not limited to a particular way of ordering the items as in a physical space, but can change axis at will.

4.2 A View-Based Search Interface for Efficient Search

Spot searching, most often currently realized through keyword searching, provides the user with a fast way to reach their goal. It requires that the user knows what they are looking for, and additionally knows how to describe it in the terms the search engine requires. Yellow pages service directories is a domain where one can often expect users to know what they are looking for. There are no guarantees, however, that the user can formulate their queries accordingly. The view-based yellow pages search portal Veturi, described shortly in publication III, was created to address this search problem. The portal contains some 220,000 real-world services from both the public and private sectors, annotated semantically with a SUMO-based [57] service ontology.

The user interface of Veturi is based on on-the-fly semantic autocompletion (see publication IV and section 3.3 here) of keywords into categories, made possible by the use of AJAX⁶ techniques. This user interaction pattern tightly integrates keyword

⁶Asynchronous JavaScript and the XMLHttpRequest -object, which allow for mak-

searching with the specificity, semantic disambiguation, and context visualization capabilities of the view facets, as described in the following.

Figure 4.5 depicts the search interface of the Veturi portal. The five view facets used in the portal describe the following aspects of the services provided: Consumer (“Kuluttaja” in Finnish), Producer (“Tuottaja”), Target (“Mitä”), Process (“Prosessi”), and Location (“Paikka”). These facets are located at the top, initially marked only by their name and an empty keyword box. Typing search terms in the boxes immediately opens the corresponding facet to show matching categories. The results view below the facets also dynamically updates to show relevant hits, defined by the current search constraints in other facets and a union of all the categories in the current facet matched by the keyword. If there is a need for more specificity or an alternate selection, a single category can be selected from the facet. After such a selection, the facet again closes, showing only the newly selected constraint, with the results view updating accordingly.

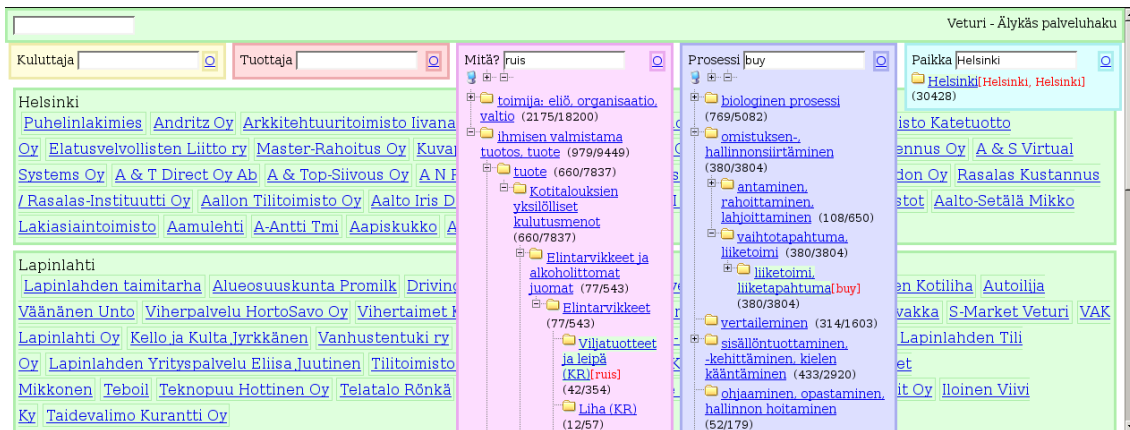


Figure 4.5: The Veturi user interface

The user is guided in formulating his query by focusing the views on clearly identifiable distinct variables of the service. For users more familiar with the portal and its

ing HTTP calls to the server in the background while viewing a page. See e.g. <http://en.wikipedia.org/wiki/AJAX>.

service description model, a globally effective keyword search box is provided in the upper left corner of the interface for quick, undifferentiated searches. Because in the service model used the contents of the views seldom overlap, most queries can be adequately and precisely replied to simply by typing the service need in plain text in the global keyword box, e.g. “car repair helsinki”, with possible disambiguation done through selections in the facets.

The example search depicted in figure 4.5 shows a user trying to find out where he can buy rye bread in Helsinki. He has already selected Helsinki as the place for the services he requires. Now, he is in the process of describing the actual service. In the view “Mitä?” (service target), the user has typed in the word “ruis” (rye). While the annotation ontology used does not contain different grains, the textual description of the category “Viljatuotteet ja Leipä (KR)” (grain products and bread) contains a reference to rye, resulting in a category match. In this way, existing textual material can be used to augment incomplete ontologies to at least return some hits for concepts that have not yet been added to the ontology.

In the interface, the matched categories are shown directly in their hierarchical contexts. This allows for quick evaluation of the relevance of the hits, as well as reveals close misses. For example, a user may enter the common-language keyword “vitamin”, while she actually meant the whole category of dietary supplements was meant. As a side effect of viewing the trees, the user is also guided on the content of the collection and how it is indexed in the system. The trees can also be opened and navigated freely without using keywords for an alternate form of navigation and familiarization with the indexing concepts and facets.

The search query entered in the view “Prosessi” (Process) divulges an additional feature of the portal: multi-language support. Typing in the word “buy” matches the appropriate “liiketoimi, liiketapahtuma” (business transaction), even though the word for “buy” in Finnish would be “ostaa”.

On selecting an individual service from the results, the user is taken to an item page similar to the one in MuseumFinland, with lateral links to other services in the collection. Here, however, the services are linked using more specific rules. For example, the item page for a hotel shows nearby restaurants and nightclubs, and the item page for a car repair service contains links to nearby taxi companies.

In summary, the Veturi interface provides a powerful tight coupling between the keyword and categorization approaches to service discovery. The fact that the Veturi search can be started, and usually also completed simply by typing in keywords provides the users with a familiar entry point to the system. Still, the semantic firmness inherent in the categories is transferred into a sense of security for the user. Users more familiar with a category-based approach are catered for, too, with the added benefit of having multiple viewpoints to choose from, in contrast to the single categorization approaches commonly in use.

4.3 Further Interfaces

Together, the two portals presented prove the applicability of the view-based search paradigm to two polar ends of search needs, as well as highlight how it is possible to tweak the systems to cater to both at the same time.

In addition to these two main portals, additional view-based search interfaces were created, spanning a wide variety of different search and browsing tasks. These are: a mobile version of MuseumFinland (publication II), a standalone museum exhibit system about university promotions called Promoottori [36], CultureSampo I and II (publication VI), the health promotion portal HealthFinland [72], the e-learning portal Orava [45] and its successor Opintie.

Of these, HealthFinland is interesting in that user feedback on its interface eventu-

ally resulted in creating separate browsing and searching interfaces for it. Yet both utilized the same underlying view-based search engine. The MuseumFinland mobile interface on the other hand demonstrated the possibility for view-based interfaces to meet the strict screen-space and interaction constraints of mobile devices, as well as integrated geolocation-based searching to the system. The Orava interface in turn was originally created by a group of students⁷ as a software engineering project at the University of Helsinki Department of Computer Science. The intent was to test how usable our view-based portal creation tool was for outside users trying to create a new semantic portal with it. The result was an alternate search/browsing interface sitting between MuseumFinland and Veturi.

Taken together, these additional interfaces, residing in various spaces between the two extremes of spot search and browsing, give additional weight to the argument on the versatility of the view-based paradigm.

4.4 Adaptability of the Paradigm to Different Domains

During the course of this thesis, the view-based search paradigm was applied to the following eight separate singular domains:

1. museum artifact data and photographs in MuseumFinland (publication II),
2. yellow pages service directory information and health service information in the Veturi portal (publication III),
3. educational and historical videos in the Orava and Opintie systems [45],
4. ontology information in the ONKI ontology browser test (unpublished)
5. health promotion information in HealthFinland [72],

⁷<http://www.cs.helsinki.fi/group/orava/>

6. a database of photographs relating to university promotion events in the Promoottori system [36],
7. the dmoz.org open directory project⁸ website directory (unpublished), and
8. link library data of the Suomi.fi⁹ e-government portal [69].

Of these, particularly interesting here are the open directory project data and the Suomi.fi data, because they are both originally single hierarchy classifications of links to information items. The case studies concerning these tell how such data can be converted for view-based search.

In the case of the dmoz.org data, the top level categories in the single hierarchy actually made workable, if not perfect views. Thus, the portal ended with the views Arts, Business, Computers, Games, Health, Home, Kids and Teens, News, Recreation, Reference, Regional, Science, Shopping, Society, Sports, and World.

Unfortunately, the same was not true of the singular topic classification of Suomi.fi. Instead, for the semantic version of the Suomi.fi portal¹⁰, new views were built up from scratch to complement the existing view [69]. The new views concerned the type of the content, the language of the content, the target group, and their status in life, and any specific spatial region where the data was useful.

Together, these varied application domains speak for the applicability of the paradigm to varied data on the Semantic Web. Yet, the single classification materials also highlight a possible restriction on the usability of the paradigm. There may be some data where only the single classification is sensible, or any other categorizations produced do not intersect efficiently with it. This may be true for example with homogeneous

⁸<http://www.dmoz.org/>

⁹<http://www.suomi.fi/>

¹⁰Available at <http://demo.seco.tkk.fi/suomifi/>

collections of articles, where a single topic hierarchy cannot be efficiently separated into sensible views.

4.5 Expanding the Paradigm to Heterogeneous Datasets

Thus far, all work presented has focused on a single domain in turn. After the work on the MuseumFinland portal and publication pipeline however, our work moved onto a wider eCulture application called CultureSampo¹¹, described in publications VI and VIII. The core idea here was to expand into other cultural content than museum items, and thereby explore the area of semantic cross-domain interoperability and multi-domain user interfaces for vastly heterogeneous datasets.

4.5.1 Problem Definition

Paper VI presents some of the problems encountered while integrating heterogeneous data for the CultureSampo portal. However, the exposition there is incomplete, so a more complete version is given below.

First, data integration problems arise with regard to properties. These stem both from the inherent heterogeneity of the data, as well as from modeling differences in the original databases. For example, even inside a domain, one museum collection may use a general “place of creation” property, while another uses the more distinct “place of manufacture”. In a collection of paintings on the other hand, these might be “place of painting” versus “place of creation”.

Sometimes, these matters of generality are even more complicated. For example,

¹¹Portal available at <http://www.kulttuurisampo.fi/>

the schema for Finnish Museums Online¹², an aggregator service in itself, contains a field “place of acquirement/place of discovery”, which irrevocably combines these two fields found separate in other collections. Conversely, properties with the same name do not always mean the same thing. The property “color” in a museum database usually describes the coloring of the objects, while in a particular photography database it is a binary predicate with options “color” and “monochrome”.

Also, with regard to user interfaces and traditional view-based search in particular, even after thorough unification of properties (potential views), there are just too many of them left. In the final CultureSampo portal for example, there are about 200 truly semantically different properties among the 20 or so different content types of the portal. Another problem with regard to view-based search here is that the degree by which these properties are shared across content type and between original databases varies wildly. For example, in the data for CultureSampo, the property “color” is stored only for one collection of paintings and only a part of the museum item collections, even if it would apply to other objects as well.

These same problems of integration also apply to the values of the properties, i.e. different collections may use different vocabularies, such as one designating an item as “man-made” while another uses “crafted by hand”. Also the annotation level of granularity may differ, such as one collection making a distinction between a chalice and a goblet, while another would classify them both just as drinking vessels. In CultureSampo, this last problem was much diminished, because Finnish libraries and Museums have a long tradition of drafting and making use of common vocabularies. However, all these vocabularies were still special to a single field such as fiction literature as opposed to museum artifacts, or works of fine art.

In MuseumFinland, all these integration problems were sidestepped by defining a limited common schema and vocabulary. We then required all participants to map

¹²<http://suomenmuseotonline.fi/en>

their data to that schema, as well as requiring any missing data to be filled. While this worked for a single domain and a controlled set of content producers, these requirements had to be loosened for CultureSampo.

4.5.2 An Event-Based Approach

Paper VI documents the basis of our first approach at solving problems of view-based search for heterogeneous data sets, more fully expanded in [65]. Here, the idea was to map the schemas to a more primitive homogeneous representation based on events and thematic roles. For example, consider the following metadata about a painting and a person:

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix person: <http://www.yso.fi/onto/person/> .
@prefix time: <http://www.yso.fi/onto/time/> .
@prefix place: <http://www.yso.fi/onto/place/> .
@prefix cs: <http://www.kulttuurisampo.fi/data/> .

cs:Kullervo_departs_for_war
  dc:creator person:A.Gallen-Kallela ;
  dc:date time:1901 ;
  dc:spatial place:Helsinki .

person:A.Gallen-Kallela
  cs:placeOfDeath place:Stockholm ;
  cs:timeOfDeath time:1931 .
```

Using mapping rules, the following corresponding event descriptions were generated:

```
@prefix e: <http://www.yso.fi/onto/event/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix person: <http://www.yso.fi/onto/person/> .
@prefix time: <http://www.yso.fi/onto/time/> .
```

```
@prefix place: <http://www.yso.fi/onto/place/> .
@prefix cs: <http://www.kulttuurisampo.fi/data/> .
```

```
cs:painting_event_45
  rdf:type e:painting_event ;
  e:agent person:A.Gallen-Kallela ;
  e:patient cs:Kullervo_departs_for_war ;
  e:time time:1901 ;
  e:place place:Helsinki .
```

```
cs:death_event_41
  rdf:type e:death_event ;
  e:patient person:A.Gallen-Kallela ;
  e:time time:1931 ;
  e:place place:Stockholm .
```

The idea was to use events as a harmonizing representation format underlying the heterogeneous data. However, while this worked sufficiently well as an underlying data model for reasoning and recommendation, it created problems on the user interface level.

For the CultureSampo II prototype described in paper VI, a view-based search interface was prepared that used this event schema directly, i.e. the views were “event type”, “event location”, “agent”, “patient”, and “event time”. When user tests were conducted on this prototype, they resulted in a verification of the usefulness of the basic view-based search paradigm [38]. The event-based views themselves, however, were criticized as unintuitive. Based on this, as well as interviews with personnel from organizations doing indexing for CultureSampo, it became apparent that while events may be a good base for tying content together, they are not intuitive to the users.

While bringing events to the fore, the approach fractured and distributed the metadata of the original primary objects into various events and into different roles in those events. This meant that traditional and well-understood attribute-value pair visualizations could no longer be applied to the original objects. Instead, complex

visualization were needed that placed them in relation to all the events that touched them. These in turn were considered both by users and annotators as vastly less clear and usable than the original primary object-oriented metadata.

Thus, we had to rethink our approach. We returned to the original noisy data in traditional schemas and traditional integration approaches concerned with property and vocabulary mapping, and focused on expanding and modifying the paradigm on the user-interface level to cope with the results.

4.5.3 Domain-Centric View-Based Search

Paper VII presents our current solution to view-based semantic search for heterogeneous data. This approach, termed domain-centric view-based search, is based on the realization that even when the amount of different properties grows quickly, the amount of domains grows much slower. That is, many of the properties share the same range of values, such as places, times, or people.

It is then possible to modify the view projection algorithms so that they create views based on domains and not properties, i.e. they create a single “place” view instead of “place of use”, “place of manufacture” and so on. However, this does not solve the whole problem, because if only a simple “place” view is shown, it gets harder for people to understand the actual links between the items and the places shown. Also the expressive power of the interface is diminished, as one can no longer e.g. search for items made in Japan but used in Europe.

These problems were solved by two measures. First, in the presentation, for each item an explanation is included of the property-concept relationships that place that item in the result set. Second, the properties were brought back to the views, but in a different form. Now, a view consists of two selectors: one for selecting the domain

concept and another for limiting the role (property) that the concept has in relation to the search items. Here, the user is free to search both with and without specifying a role, actually increasing the expressiveness of the view-based search paradigm.

Figure 4.6 shows a sample domain centric view. Here, the user has used a semantic autocomplete functionality integrated into the view to locate Japan in the place hierarchy. This constrains the search to all items in any way related to Japan. However, she has also been shown a tree view (based on subproperty relationships) of the predicates that link content to Japan. Here, the user has selected “by place of manufacture”, thus ending up with only objects manufactured in Japan.

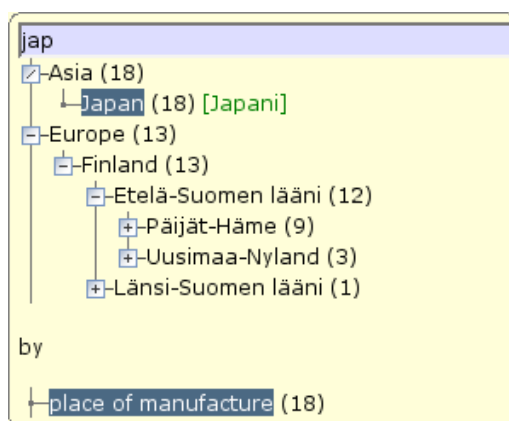


Figure 4.6: Domain Centric View Based Search

4.5.4 The Search and Organize User Interface Concept

In CultureSampo, the work on view-based interfaces for heterogeneous data yielded not only a technical solution, but an important argument for shifting focus in semantic search from items themselves to using them as lenses to wider topics. This development is detailed fully in publication VII, but because it is so important, the core argumentation and solutions are repeated here.

Traditionally, Internet search has been about finding a document or documents that answer the question posed by the searcher. Semantic Web search systems have mostly also held this viewpoint [31], using properties and concepts in domain ontologies to locate search objects annotated with them. For semantically annotated content analogous to text documents, this works adequately, but for qualitatively different material, it creates problems. To understand why, one must take a step back to look at information needs.

Classifications of information needs [2, 6, 12, 15, 39, 78] agree that there is a major partition between look-up queries like “For my meal, I need a *white wine* with a *spicy flavor*” and more general information needs such as “tell me all about *spicy white wines*”. The former focuses on selecting, fact finding, and question answering, while the latter deals with the more general objective of learning and investigation, containing in addition to searching also tasks such as comparison, interpretation, aggregation, analysis, synthesis, and discovery [49]. Depending on domain, at least a significant part (22% [14]), or even the majority (70% [77], 67% [12]) of inquiries for information relate to learning as opposed to spot queries.

Despite this, search research has only recently begun to move to this expanded domain, termed exploratory search [49]. We propose that a major reason for this is that as long as the information is encoded only inside documents, learning and investigation searches are adequately catered for by the same functionality as fact finding, i.e. locating all matching documents and then perusing each for relevant data [39].

For semantically annotated content other than information documents, the situation is different. Often the useful information is not the object itself, but the relation between the object and the ontological resources associated with it. Now, for question answering such as what wine to have with a particular food, the answer is still a particular object with particular characteristics, and the old paradigm still works.

For the more general type of queries, on the other hand, typical Semantic Web object databases fall short, as they contain no singular exposition about, e.g. “French spirits”.

However, if looked at from another perspective, the data contains ample information to answer someone wanting to know about French liquors. It is merely encoded differently, distributed across the multiple object annotations and ontologies. To pull this information out, one must move the focus from individual items to the set of objects with particular properties as a whole, and even further. What one actually wants is to look at the combination of the domain concepts “French” and “spirits” through the lens of the items.

Actually, if an interface capable of such can be created, the pieced nature of the information becomes an advantage, as the pieces can be combined to shed light on a much wider variety of topics than anyone could write an explanatory article on. This capability is even further enhanced if the database contains material of multiple different kinds. For example in the cultural heritage domain, with suitable material, one could learn not only about 19th century Finnish crafts, 19th century Finnish paintings etc., but actually of the 19th century Finland as a whole.

Based on this analysis, I argue that to support exploratory search tasks, Semantic Web application designers need to shift focus from object location to the creation of structured, domain-centric presentations based on those items.

An interface for doing exactly this is also described in publication VII. Here, taking cue from actual museum exhibitions, the user is presented with a user interface for organizing their own virtual exhibition. In essence, this interface is an elaboration of view-based search. Here, one area and some views of the user interface are geared specifically toward result set selection. Another area and views on the other hand focus on different ways of informatively organizing the result set according to the

various view dimensions, such as in a one to two-dimensional matrix, on a map, or on a timeline.

While details of this (apart from domain-centric view constraining) are left to publication VII, some example exhibitions generated are shown here to illustrate the possibilities of this approach. Figure 4.7 shows an exhibit on Japanese items imported into Finland organized into rooms by date of manufacture and item type. Here, the user can instantly see the rise in production of high technology goods in Japan in the later parts of the 20th century. Figure 4.8 on the other hand shows how a dedicated map visualization of the results can be useful in gauging the distribution of churches in southern Finland. Finally, figure 4.9 displays how a dedicated timeline visualization of items related to a particular keyword can be used to discern if there was a change in beard styles in Finland near the end of the 19th century.

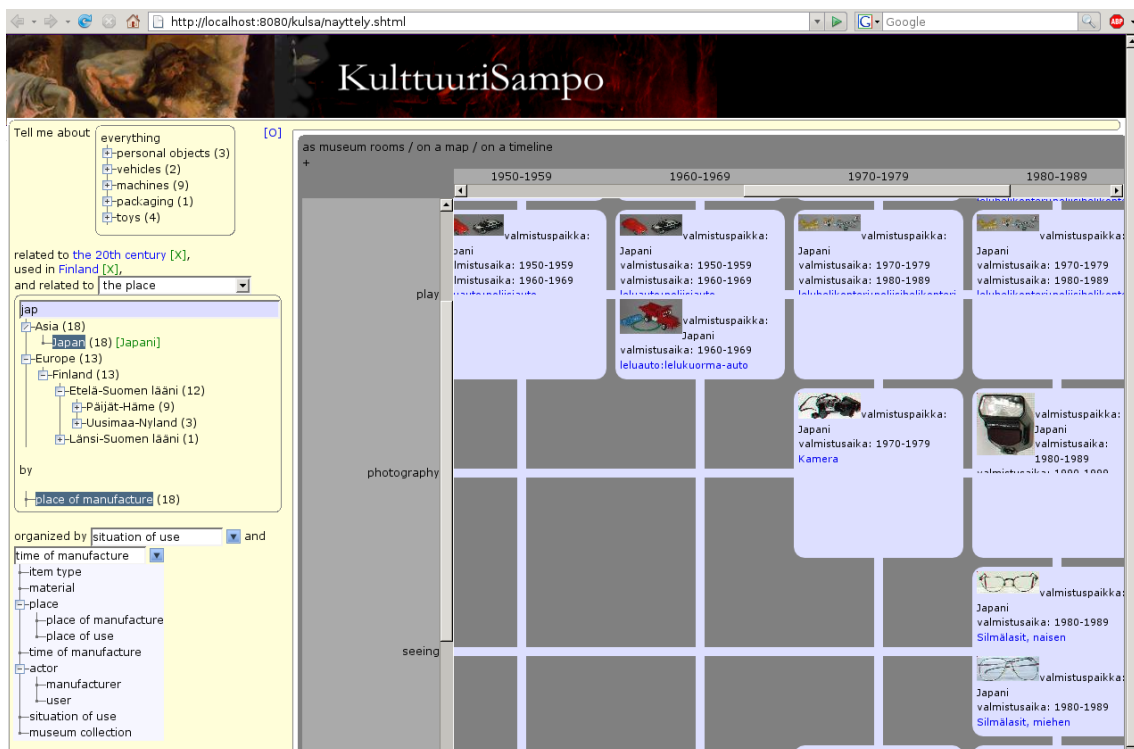


Figure 4.7: Exhibition room visualization in CultureSampo

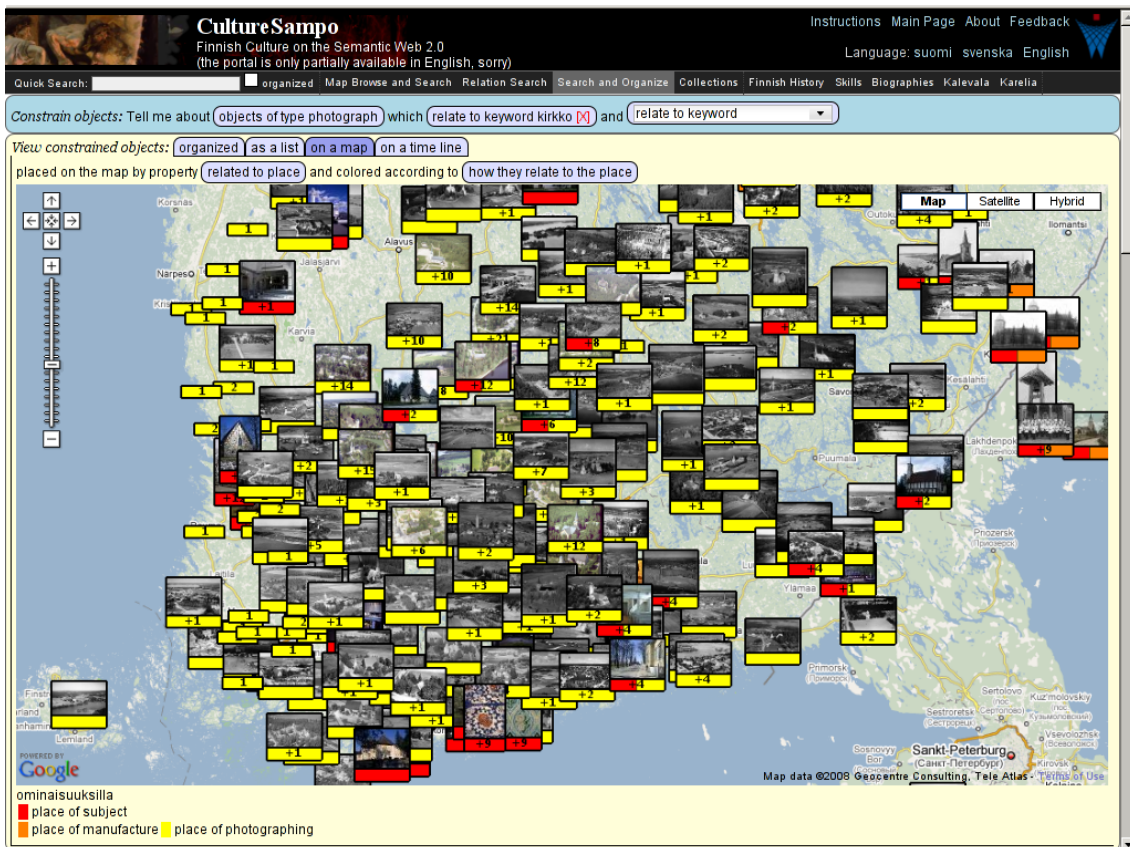


Figure 4.8: Map visualization in CultureSampo

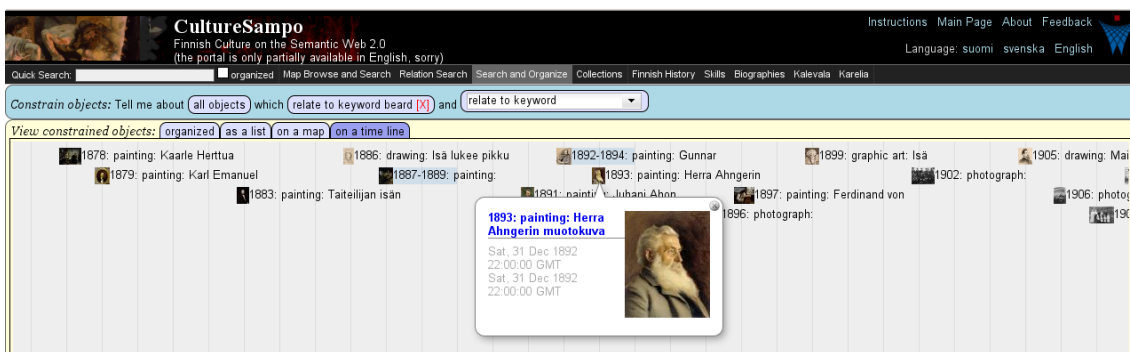


Figure 4.9: Timeline visualization in CultureSampo

4.5.5 Thematic Views

While the exhibition generation view presented before is very powerful, it can also be quite overwhelming for a first time user. To address this, we provide for the Cul-

tureSampo portal not only a single massively user-configurable view-based interface, but also a selection of expert pre-selected views to the data, based on thematic viewpoints.

Apart from one, these views, described in more detail in publication VIII, provide pre-selected and pre-configured subsets of the complete search and organize functionality. For example, in the history view of CultureSampo, a preselected query returns only historical events, and the user is left with choosing from pre-configured timeline and list view visualizations. Because they are based on common general functionality, it is easy to add more of these views based solely on the recommendations of content access specialists. In this way, these views are very closely comparable to traditional physical exhibitions, with items and information pre-selected and pre-organized into various forms of thematically interesting and informative displays by cultural heritage institution curators. The idea here is to again transfer some work from the user into the hands of experts, as was done in selecting the views to be projected in MuseumFinland. Here, however, we go further, selecting and organizing whole interface elements into thematic views simpler than the complete exhibition generation interface.

As said, there is one view which cannot be described as offering a subset of the search and organize functionality, and is thus interesting when discussing the limitations of the view-based search approach. This is the relational search view [44]. Here, the user can enter the names of information resources, and is returned with a description of how they are related to each other.

The key difference between this and the other views is that here the interesting information items are the paths between content items and not a set of those content items themselves organized in some way. Now, while the view-based search paradigm itself could be pivoted to consider paths as information items, it is hard to think of good visualization views that would categorize different paths in any meaningful

way. In essence, it would seem that this functionality is truly best left as orthogonal to view-based search.

To some extent these approaches can be combined for added benefit, such as providing additional information on relations between information items by visualizing them inside the views of view-based search. We have later had success with such approaches with for example visualizing the movements of people between places, as well as visualizing on a map the import and export patterns of different types of items [42].

5 Design Issues for View-Based Semantic Web Interfaces

As discussed in the context of this work, after discovering general paradigms for doing search and browsing on the Semantic Web, it was important that the software components developed to manifest those paradigms would be as configurable and reusable as possible.

5.1 The Semantic Portal Creation Tool OntoViews

In order to do this, the OntoViews¹³ framework was built. The major design principles underlying this tool were to make it 1) easily adaptable to new underlying domain ontologies, 2) easy to extend and adapt to new user interfaces and interaction patterns, 3) as modular as possible and 4) uphold a clear separation between the major components of the system. In accordance with these guidelines, OntoViews consists of three major components: 1) OntoViews-C, the user interface and interaction controller, 2) Ontogator, the view-based search engine, and 3) Ontodella, an SWI-Prolog-based¹⁴ logic server capable of both view projection and item recommendation generation. All components were designed to be as independent from each other as possible, with interfaces between components based on formalized RDF/XML representations. Thus, for example, Ontogator can be integrated as the view-based search component for any system that can produce and parse either RDF or XML.

These components and their implementations are discussed in detail in publications

¹³The tool is available for free use, under the MIT open source license at <http://www.seco.tkk.fi/projects/semweb/dist.php>.

¹⁴<http://www.swi-prolog.org/>

II and V. Only the main principles and discussion concerning them will be presented here.

5.1.1 The Projection and Semantic Linking Engine Ontodella

A crucial part of the adaptability of a view-based search system on the Semantic Web is the flexibility and ease of use of the component responsible for projecting views from the underlying ontology knowledge. In designing the original OntoViews architecture, it was decided to use Prolog-based logic rules as a basis for projection. The power of Prolog allows formulating complex rules when necessary, but the most common case, where projection is based on simple transitive properties, can also be easily and shortly encoded. A similar rationale was also applied to the semantic linking of items with each other, utilized in the semantic browsing part of the user interface.

Because both the projection rules and the semantic linking rules of OntoViews are pure Prolog, the Ontodella logic server component [75] is mostly just a thin wrapper over the core SWI-Prolog engine. Its responsibilities are loading an RDF data model into the engine, organizing projection, listening on an HTTP port for semantic link generation requests, and serializing various results produced by the system to RDF/XML for output.

5.1.2 The Semantic View-Based Search Engine Ontogator

The search engine of OntoViews, Ontogator, described in detail in publication V, is a general-purpose extensible view-based RDF search engine. It was originally built specifically for tree hierarchies, and while further revisions opened the system a bit to support non-hierarchical categorizations, the interface and optimizations in the

system remain specifically tied to tree hierarchy based querying.

The main value in Ontogator lies in the work done on its application programming interface (API), which highlights the requirements a generic stateless view-based search engine service must meet in order to adapt to different tasks and requirements. Summarizing from publication V these are: 1) how categories are identified in the interface, 2) using established Semantic Web standards in queries and result serialization, 3) extensibility of the engine with custom functionality, and 4) scalability, both in indexing efficiency and interface options.

5.2 Content Production Architecture of Semantic Portals

Thus far, only the system architecture of our semantic portals has been discussed. However, the content production architectures of the portals are equally important. These consist of the schemas, vocabularies, data production pipelines, and supporting infrastructure that are necessary for getting data into the portals. During the course of this work, two content production architectures were created, one for MuseumFinland and one for CultureSampo. In addition, a separate distributed content creation architecture was created for the HealthFinland portal [72].

The earlier, single domain content production architecture of MuseumFinland is discussed exhaustively in publication II. Here, the main work was in creating the ontologies to be used as common vocabularies, as at the time no suitable ones were readily available. Another innovation was that the content providers were not forced to use a single unified terminology in their own databases. Instead, they could provide a mapping which related their own terms to the common ontology space of the portal. In addition, it was found that in mapping the original literal values found in museum databases to ontology concepts, most mappings could be done automatically based on simple rules, with only a small fraction (3,75% to 8,57%) of

the values needing manual disambiguation because of homonymy.

The MuseumFinland content production architecture was created in isolation, as there was no common infrastructure to build on. In contrast, the content production architecture of CultureSampo relies heavily on the FinnONTO infrastructure, particularly the core system of mutually mapped ontologies collectively termed KOKO. As discussed in publication VIII, this system is a plug-in architecture, where domain ontologies curated by parties of interest are joined together by the common Finnish Upper Ontology YSO. It is this well-curated ontology infrastructure that in the end makes it possible for CultureSampo to intelligently relate together content from its vastly heterogeneous sources.

Other parts of the CultureSampo content production architecture are similar to the MuseumFinland one. Any local terminology not already linked to KOKO is mapped to KOKO concepts, while manual disambiguation and correction of content is made possible by common components in the FinnONTO infrastructure such as the SAHA metadata editor [74] and ONKI ontology servers [76]. However, the CultureSampo architecture also includes many components for inferring additional information about the items, such as inferring ontological places of photography from place names featured in photograph titles.

A difference between the content production pipelines in MuseumFinland and CultureSampo is also the need for mapping between heterogeneous content schemas in the latter. As discussed in section 4.5, in the published version on the web I decided to do this mapping by traditional class and property mappings. In practice this means that for each new data source, in addition to any local terminology, any local properties and object types also have to be mapped to the existing schema space of CultureSampo. Experience has shown that it has been quite easy to do this by hand thus far. With the growth of CultureSampo however this may become problematic, as the number of properties in the common schema has already grown past 200.

6 Discussion

As stated in section 1.2, the research questions of this thesis were:

1. Seek a general user interface paradigm that:
 - (a) can be applied to as wide a variety of Semantic Web search and browsing tasks as possible.
 - (b) aligns well with Semantic Web technologies in the sense that it is easy to make maximal use of the semantics inherent in the data.
2. Identify supporting elements that make the paradigm more usable and adaptable.
3. Discover design guidelines that enable the adaptability of such systems in the context of the Semantic Web.

These resulted in hypotheses that the user interface paradigm of view-based search would be able to:

1. cater to the breadth of user demands.
2. adapt to different kinds of data.
3. compete in conceptual capability with existing approaches.
4. align well with Semantic Web technologies.

The interfaces created as part of this thesis confirm the hypothesis that view-based search presents a versatile and powerful paradigm for creating interfaces on the Semantic Web. The paradigm could be applied to solve differing search tasks spanning

the full range between the browsing and spot searching strategies. Still, the requirements of the two modes were so different that no one interface could be made to be equally supportive of both. So, while the interfaces created as part of this research do support both modes as much as possible, they differ in which one is prioritized over the other, with the MuseumFinland interface most geared toward browsing, and the Veturi interface most geared toward spot search.

As an indication of the usefulness of the approach, MuseumFinland, the oldest of the portal interfaces, has received several public awards. These include the Semantic Web challenge award 2004 (second place) and the Finnish Prime Minister's commendation for the most technologically innovative application on the web 2004. The portal was also a jury nominated finalist in the Nordic digital excellence in museums awards, in the best Web based / Virtual application category.

Analyzing the interfaces, a number of benefits persist through all of them, explaining the power of the approach:

- Because the collection is visualized along different categorizations, the user is able to immediately familiarize herself with the contents of collection, as well as how it is organized in the database.
- Showing possible constraints in multiple views simultaneously allows the user to start constraining their search with the aspect most natural to them, and continuing from there. This is a particular strength when compared with classifications based on a single hierarchy, such as the ones used in the Yahoo!¹⁵ and Open Directory Project¹⁶ directories.
- Because the views show categorizations of the items, and the categories shown are used as search constraints, any information linked to them, such as hit counts, is immediately useful in evaluating further constraining actions.

¹⁵<http://dir.yahoo.com/>

¹⁶<http://dmoz.org/>

- Visualizing results from multiple viewpoints is an intuitive, simple way to present how the result set fits into the possibly complex larger context of the domain. It also allows the user to answer questions about sets of items, not just individuals.
- In contrast to keyword searches, the semantic firmness inherent in the categorizations and constraining is transferred into a sense of security in the user of locating all the relevant items.
- Provided that the views intersect efficiently with each other, only a few selections are needed to achieve a wanted narrow result set.

The versatility of the paradigm with regard to client device and environment concerns was confirmed in creating the MuseumFinland Mobile interface. The paradigm also proved to be sufficiently extensible, testified to by the easy and tight integration of semantic autocompletion, geolocation-based searching as well as context visualization. The Veturi version of keyword-search also makes use of simple ontology navigation to match keywords from ontology entities to categories. Extending the search with lateral semantic browsing functionality could also be done without notable semantic disconnect, based on a natural flow from result set constraining to browsing. Having the view categories double as rules linking the items together provided additional ease.

The extension of view-based search to domain-centric view-based search showed how the approach can be applied to heterogeneous data. The search and organize user interface concept that grew out of this on the other hand yielded an important argument for shifting focus in semantic search from items themselves to using them as lenses to wider topics. The view-based exhibition generation interface developed in CultureSampo was able to cater to this new focus very well.

While the results clearly show that the view-based approach is a good approach to search on the Semantic Web, it still needs to be oriented with respect to other available approaches.

On the browsing side, an advantage of the view-based search paradigm is the ability to visualize many choices while still preserving their semantic context. A new user will quickly get to know the collection and the way it is organized, as well as be able to locate interesting further choices at each decision point. A drawback is that in the end, the approach is a query constraining method maintaining a result set and query state. This makes it hard to provide view-based browsing as just one equal browsing alternative among many, an often wanted quality in a versatile browsing application. Instead, like in MuseumFinland, a view-based browsing interface can be used as a starting point for further browsing, familiarizing the user with the collection, and allowing them to hone in on an interesting starting point for further exploration.

On the spot search side, all the benefits of the approach apply. Of particular interest, however, is how these qualities compare with those of the other formalisms for creating complex graph patterns discussed in section 2.1.3 of the survey section. The paradigm seems more intuitive than the alternatives presented, as in a well-crafted application, most of the complexity has already been hidden by the designer of the view projection. Based on the same argument, expressive power should also not fall notably below the other query forms. Unfortunately, formal usability testing between these interfaces has not been possible. This is mostly because no stable, obtainable full implementations of the other interfaces exist, and thus any testing would require considerable implementation work and resources not currently available.

During the time frame of this research, other implementations of view-based search for the Semantic Web have also surfaced. The Longwell RDF browser¹⁷ provides a general view-based search interface for any data. However, it supports only flat, RDF-property-based views. The SWED directory portal [63] is a semantic view-based search portal for environmental organizations and projects, with an interface very similar to MuseumFinland, but lacking semantic keyword search, the whole classification view, and semantic browsing functionality. Also, the view hierarchies are not projections from full-fledged ontologies, but are manually crafted using the W3C SKOS [51] schema for simple thesauri. The portal does, however, support distributed maintenance of the portal data. The Seamark Navigator¹⁸ by Siderean Software Inc. is a commercial implementation of view-based semantic search. It also, however, only supports simple flat categorizations. Later systems, offering their own expansions to the paradigm are [54], mSpace [67], /facet [30] and Exhibit [34].

Regarding the OntoViews architecture, the implementation has also proved a success. The system has been used to create altogether five different user interfaces. At the same time, the projection functionality has been tested on eight vastly different data sets. The system was also tested to scale well to hundreds of thousands of items using the dmoz.org material.

The user interface, interaction, and control component of OntoViews, called OntoViews-C, was found to be eminently portable, extensible, modifiable, and modular, as seen in the multiple user interfaces that could be designed using it. This flexibility was a direct result of building the application on top of Apache Cocoon¹⁹, with its concepts of transformers and pipelines. This was further confirmed by the fact that a previously tried and abandoned servlet-based approach did not share these qualities.

¹⁷<http://simile.mit.edu/longwell/>

¹⁸<http://siderean.com/products.html>

¹⁹<http://cocoon.apache.org/>

The modular architecture also allowed all the transformer components to be made available for use in other web applications as web services. In this way, other web applications could make use of the actual RDF data contained in the system, querying the Ontogator and Ontodella servers directly for content data in RDF/XML-format. This also provided a way of distributing the processing in the system to multiple servers.

The use of XSL Transformations [13] in most of the user interface and query transformations made it simple to carry out changes in layout and functionality. For example, creating the MuseumFinland Mobile interface and the ONKI browser interface both took less than three days of implementation work. A probable explanation for the ease XSLT brings lies in the prevalence of trees in both the queries, query results, and UI visualizations, as XSLT is specifically designed for processing such hierarchically structured documents.

However, there are also some problems in using XSLT with RDF/XML. In general, the same RDF triple can be represented in XML in different ways but an XSLT template can be only tied to a specific representation. In the OntoViews system, this problem was avoided because the RDF/XML serialization formats used by each of the sub-components of the system were known, but in a general web service environment, this could cause complications. The core search engine components of OntoViews would however be unaffected even in this case, because they handle their input with true RDF semantics.

The use of XSLT also led to some complicated transformation templates in the more involved areas of user interaction logic, for example, (sub)paging and navigating the search result pages. Therefore, in the next evolution of the system used in CultureSampo II (publication VI), the interaction logic was pushed back to Java code, with only the layout done using templates.

The framework also started to run into problems in interfaces that required tight integration between the server and browser, like the AJAX-powered Veturi interface. For every update of the interface, a whole pipeline had to be constructed, and there were no easy facilities for maintaining application state. For example, when navigating tree hierarchies in the Veturi interface, most queries are just opening further branches in a result tree already partially calculated. In OntoViews, however, the whole visible tree needed to be recalculated and returned, because the view-based search was isolated into a separate component.

These requirements also had effect inside the Ontogator search engine. A move was needed from an expectation of monolithic queries and responses to providing all sorts of view-based search related services, tightly integrated with an outside query execution controller that directs the query as the application demands. Another barrier for expansion in Ontogator was its history as a purely tree hierarchy view-based search engine, with grouping into tree categories tightly coupled into the implementation.

Our solution to this, implemented in the version of CultureSampo described in publication VIII, was to partition the view processing into two completely separate components. First, a result set is generated based on pluggable selector components. Then, this result set is fed via a standard interface to configured visualization components. While most views in traditional view-based search implement both functions and operate on the general search result set, this also allowed a more dynamic flow of data and functionality. For example, a list view could be hooked to a result set provided by a map view in order to display additional information on items related to a particular place.

This separation also made it possible to reuse some of the components in tasks other than view-based search, and led to a larger paradigm of reusable components, termed Semantic Web widgets [53].

7 Conclusions

This thesis presented the view-based search paradigm as a viable basis for querying on the Semantic Web, especially coupled with the idea of view projection from ontologies. Through testing with actual implementations, the paradigm was found to be very flexible and extensible in creating a wide range of interfaces suitable for different tasks in different environments.

When other choices are available, the paradigm should especially be considered when:

- there is a need to express complex combinatorial queries intuitively
- there is a need for visualizing result sets, not just individual items
- the contents are complex enough that the choice of a constraining viewpoint is useful
- there is use in allowing the users to gently familiarize themselves with the contents and organization of the portal.

Then also, the following drawbacks should be weighed:

- The paradigm is overarching, in the sense that it is hard to integrate other search functionality other than as subservient to the views.
- Other browsing functionality is similarly affected. However, the paradigm can be used to lead into separate browsing user interfaces.
- Some data may not contain the material needed to produce useful views.

When deciding upon views to be projected, the following considerations apply:

- A good view should both visually organize, as well as be able to be used to constrain the query in an intuitive manner.
- Views should provide as many separate viewpoints as possible to the data. Views with overlapping semantics are possible, but can be confusing.
- For quick operation, the views should intersect efficiently with each other, so that selecting constraints from different viewpoints quickly narrows down the result set.

References

- [1] Eija Airio, Kalervo Järvelin, Pirkko Saatsi, Jaana Kekäläinen, and Sari Suomela. 2004. CIRI - An Ontology-based Query Interface for Text Retrieval. In: Eero Hyvönen, Tomi Kauppinen, Mirva Salminen, Kim Viljanen, and Pekka Ala-Siuru (editors), Proceedings of the 11th Finnish Artificial Intelligence Conference STeP 2004, September 1-3, Vantaa, Finland.
- [2] Lorin W. Anderson and David A. Krathwohl (editors). 2000. A taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. Allyn & Bacon, Boston, Massachusetts. ISBN 978-0801319037.
- [3] Grigoris Antoniou and Frank van Harmelen. 2004. A Semantic Web Primer (Cooperative Information Systems). The MIT Press. ISBN 0262012103.
- [4] Kemafor Anyanwu and Amit Sheth. 2003. ρ -Queries: enabling querying for semantic associations on the semantic web. In: WWW '03: Proceedings of the 12th international conference on World Wide Web, pages 690–699. ACM, New York, NY, USA. ISBN 1-58113-680-3.
- [5] Nikolaos Athanasis, Vassilis Christophides, and Dimitris Kotzinos. 2004. Generating On the Fly Queries for the Semantic Web: The ICS-FORTH Graphical RQL Interface (GRQL). In: Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen (editors), The Semantic Web - ISWC 2004: Third International Semantic Web Conference, Hiroshima, Japan, November 7-11, 2004. Proceedings, volume 3298 of *Lecture Notes in Computer Science*, pages 486–501. Springer. ISBN 3-540-23798-4.
- [6] Nicholas J. Belkin, Pier Giorgio Marchetti, and Colleen Cool. 1993. Braque: design of an interface to support user interaction in information retrieval. *Information Processing and Management* 29, no. 3, pages 325–344.

- [7] Tim Berners-Lee, Jim Hendler, and Ora Lassila. 2001. The Semantic Web. *Scientific American* 284, no. 5, pages 34–43.
- [8] Tim Bray, Dave Hollander, and Andrew Layman (editors). 1999. Namespaces in XML. World Wide Web Consortium. URL <http://www.w3.org/TR/REC-xml-names/>. W3C Recommendation.
- [9] Dan Brickley and R.V. Guha (editors). 2004. RDF Vocabulary Description Language 1.0: RDF Schema. World Wide Web Consortium. URL <http://www.w3.org/TR/rdf-schema/>. W3C Recommendation.
- [10] Davide Buscaldi, Paolo Rosso, and Emilio Sanchis Arnal. 2005. A WordNet-based Query Expansion Method for Geographical Information Retrieval. In: Carol Peters (editor), *Working Notes for the CLEF 2005 Workshop*.
- [11] Tiziana Catarci, Paolo Dongilli, Tania Di Mascio, Enrico Franconi, Giuseppe Santucci, and Sergio Tessaris. 2004. An Ontology Based Visual Tool for Query Formulation Support. In: Ramon López de Mántaras and Lorenza Saitta (editors), *Proceedings of the 16th European Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22-27, 2004*, pages 308–312. IOS Press. ISBN 1-58603-452-9.
- [12] Chun Wei Choo, Brian Detlor, and Don Turnbull. 2000. Information seeking on the Web: An integrated model of browsing and searching. *First Monday* 5, no. 2. URL http://firstmonday.org/issues/issue5_2/choo/index.html.
- [13] James Clark (editor). 1999. XSL Transformations (XSLT) Version 1.0. URL <http://www.w3.org/TR/1999/REC-xslt-19991116>. W3C Recommendation.
- [14] P. F. Cole. 1958. The analysis of reference query records as a guide to the information requirements of scientists. *Journal of Documentation* 14, no. 4, pages 197–207.

- [15] Colleen Cool and Nicholas J. Belkin. 2002. A Classification of Interactions with Information. In: Harry Bruce, Ray Fidel, Peter Ingwersen, and Pertti Vakkari (editors), *Emerging frameworks and methods; Proceedings of the 4th international conference on conceptions of Library and Information Science (COLIS4)*, pages 1–15. Libraries Unlimited, Greenwood Village, CO. ISBN 978-1-59158-016-4.
- [16] Oscar Corcho, Asunción Gómez-Pérez, Angel López-Cima, V. López-García, and María del Carmen Suárez-Figueroa. 2003. ODESeW. Automatic Generation of Knowledge Portals for Intranets and Extranets. In: [21], pages 802–817.
- [17] Duane Degler, mc schraefel, Jennifer Goldbeck, Abraham Bernstein, and Lloyd Rutledge (editors). 2008. *SWUI 2008: Semantic Web User Interaction at CHI 2008—Exploring HCI Challenges*.
- [18] Jennifer English, Marti A. Hearst, Rashmi Sinha, Kirsten Swearingen, and Ping Yee. 2002. Flexible search and navigation using faceted metadata. Technical report, University of Berkeley, School of Information Management and Systems.
- [19] Christiane Fellbaum (editor). 1998. *WordNet. An electronic lexical database*. The MIT Press, Cambridge, Massachusetts. ISBN 978-0-262-06197-1.
- [20] Dieter Fensel. 2004. *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Springer, Heidelberg, 2nd edition. ISBN 978-3540003021.
- [21] Dieter Fensel, Katia P. Sycara, and John Mylopoulos (editors). 2003. *The Semantic Web - ISWC 2003, Second International Semantic Web Conference, Sanibel Island, FL, USA, October 20-23, 2003, Proceedings, volume 2870 of Lecture Notes in Computer Science*. Springer. ISBN 3-540-20362-1.

- [22] Richard Fikes, Patrick Hayes, and Ian Horrocks. 2004. OWL-QL—a language for deductive query answering on the Semantic Web. *Web Semantics: Science, Services and Agents on the World Wide Web* 2, no. 1, pages 19 – 29.
- [23] Ramanathan V. Guha, Rob McCool, and Eric Miller. 2003. Semantic search. In: *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 700–709. ACM Press. ISBN 1-58113-680-3.
- [24] Partick Hayes (editor). 2004. *RDF Semantics*. World Wide Web Consortium. URL <http://www.w3.org/TR/rdf-mt/>. W3C Recommendation.
- [25] Marti A. Hearst. 2000. Next Generation Web Search: Setting Our Sites. *IEEE Data Engineering Bulletin* 23, no. 3, pages 38–48. Special issue on Next Generation Web Search.
- [26] Marti A. Hearst, Jennifer English, Rashmi Sinha, Kirsten Swearingen, and Ping Yee. 2002. Finding the flow in web site search. *Communications of the ACM* 45, no. 9, pages 42–49.
- [27] Tom Heath, John Domingue, and Paul Shabajee. 2006. User Interaction and Uptake Challenges to Successfully Deploying Semantic Web Technologies. In: *SWUI 2006: 3rd International Semantic Web User Interaction Workshop*, Athens, GA, USA.
- [28] Jeff Heflin and James Hendler. 2000. Searching the web with SHOE. In: *Artificial Intelligence for Web Search, Papers from the workshop, AAAI 2000*, pages 35–40. AAAI Press, Menlo Park, CA. WS-00-01.
- [29] Alan R. Hevner, Salvatore T. March, Park Jinsoo, and Sudha Ram. 2004. Design Science in Information Systems Research. *MIS Quarterly* 28, no. 1, pages 75 – 105.
- [30] Michiel Hildebrand, Jacco van Ossenbruggen, and Lynda Hardman. 2006. /facet: A Browser for Heterogeneous Semantic Web Repositories. pages 272–285.

- [31] Michiel Hildebrand, Jacco van Ossenbruggen, and Lynda Hardman. 2007. An analysis of search-based user interaction on the Semantic Web. Technical report, Centrum voor Wiskunde en Informatica (NL).
- [32] Eric I. Hsu and Deborah L. McGuinness. 2003. Wine Agent: Semantic Web Testbed Application. In: Diego Calvanese, Giuseppe De Giacomo, and Enrico Franconi (editors), *Description Logics*, volume 81 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- [33] Isto Huvila and Gunilla Widén-Wulff. 2006. Perspectives to the classification of information interactions: the Cool and Belkin faceted classification scheme under scrutiny. In: *IiX: Proceedings of the 1st international conference on Information interaction in context*, pages 144–152. ACM, New York, NY, USA. ISBN 1-59593-482-0.
- [34] David F. Huynh, David R. Karger, and Robert C. Miller. 2007. Exhibit: lightweight structured data publishing. In: *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 737–746. ACM, New York, NY, USA. ISBN 978-1-59593-654-7.
- [35] Eero Hyvönen, Miikka Junnila, Suvi Kettula, Samppa Saarela, Mirva Salmiinen, Ahti Syreeni, Arttu Valo, and Kim Viljanen. 2003. Publishing Collections in the "Finnish Museums on the Semantic Web" Portal – First Results. In: *Proceedings of XML Finland 2003*.
- [36] Eero Hyvönen, Samppa Saarela, and Kim Viljanen. 2004. Application of Ontology Techniques to View-Based Semantic Search and Browsing. In: Christoph Bussler, John Davies, Dieter Fensel, and Rudi Studer (editors), *The Semantic Web: Research and Applications, First European Semantic Web Symposium, ESWS 2004, Heraklion, Crete, Greece, May 10-12, 2004, Proceedings*, volume 3053 of *Lecture Notes in Computer Science*, pages 92–106. Springer. ISBN 3-540-21999-4.

- [37] Eero Hyvönen, Kim Viljanen, Eetu Mäkelä, Tomi Kauppinen, Tuukka Ruotsalo, Onni Valkeapää, Katri Seppälä, Osma Suominen, Olli Alm, Robin Lindroos, Teppo Käsälä, Riikka Henriksson, Matias Frosterus, Jouni Tuominen, Reetta Sinkkilä, and Jussi Kurki. 2007. Elements of a National SemanticWeb Infrastructure – Case Study Finland on the Semantic Web. In: ICSC '07: Proceedings of the International Conference on Semantic Computing, pages 216–223. IEEE Computer Society, Washington, DC, USA. ISBN 0-7695-2997-6.
- [38] Thomas Häggström. 2007. Toimintakeskeisen semanttisen moninäkökymähaun toteutus ja evaluointi kulttuurialan portaalisovelluksessa. Master's thesis, Helsinki University of Technology (TKK).
- [39] Bernard J. Jansen, Brian Smith, and Danielle Booth. 2007. Learning as a Paradigm for Understanding Exploratory Search. In: Proceedings of the SIGCHI 2007 Exploratory Search and HCI workshop.
- [40] David R. Karger, Karun Bakshi, David Huynh, Dennis Quan, and Vineet Sinha. 2005. Haystack: A Customizable General-Purpose Information Management Tool for End Users of Semistructured Data. In: Proceedings of the CIDR Conference, pages 13–26.
- [41] Tomi Kauppinen and Eero Hyvönen. 2007. Modeling and Reasoning about Changes in Ontology Time Series. In: Raj Sharman, Rajiv Kishore, and Ram Ramesh (editors), *Ontologies – A Handbook of Principles, Concepts and Applications in Information Systems*, volume 14 of *Integrated Series in Information Systems*, pages 319–338. Springer-Verlag, Berlin. ISBN 978-0-387-37019-4.
- [42] Tomi Kauppinen, Kimmo Puputti, Panu Paakkarinen, Heini Kuittinen, Jari Väätäinen, and Eero Hyvönen. 2009. Learning and Visualizing Cultural Heritage Connections between Places on the Semantic Web. In: Proceedings of

the Workshop on Inductive Reasoning and Machine Learning on the Semantic Web (IRMLeS2009), The 6th Annual European Semantic Web Conference (ESWC2009).

- [43] Peter M. Kruse, Andre Naujoks, Dietmar Roesner, and Manuela Kunze. 2005. Clever Search: A WordNet Based Wrapper for Internet Search Engines. ArXiv Computer Science e-prints [arXiv:cs/0501086](http://arxiv.org/abs/cs/0501086).
- [44] Jussi Kurki and Eero Hyvönen. 2007. Relational Semantic Search: Searching Social Paths on the Semantic Web. In: Poster Proceedings of the International Semantic Web Conference (ISWC 2007), Busan, Korea.
- [45] Teppo Käsälä and Eero Hyvönen. 2006. A Semantic View-based Portal Utilizing Learning Object Metadata. In: Proceedings of the Semantic Web Applications and Tools Workshop, ASWC2006.
- [46] Alexander Maedche, Steffen Staab, Nenad Stojanovic, Rudi Studer, and York Sure. 2001. SEAL - A Framework for Developing SEMantic Web PortALs. In: BNCOD 18: Proceedings of the 18th British National Conference on Databases, pages 1–22. Springer-Verlag, London, UK. ISBN 3-540-42265-X.
- [47] Frank Manola and Eric Miller (editors). 2004. RDF Primer. The World Wide Web Consortium. URL <http://www.w3.org/TR/rdf-primer/>. W3C Recommendation.
- [48] Amanda Maple. 1995. Faceted Access: A Review of the Literature. Technical report, Working Group on Faceted Access to Music, Music Library Association. URL <http://www.musiclibraryassoc.org/BCC/BCC-Historical/BCC95/95WGFAM2.html>. BCC95/WG FAM/2.
- [49] Gary Marchionini. 2006. Exploratory search: from finding to understanding. Communications of the ACM 49, no. 4, pages 41–46.

- [50] Deborah L. McGuinness and Frank van Harmelen (editors). 2004. OWL Web Ontology Language Overview. World Wide Web Consortium. URL <http://www.w3.org/TR/owl-features/>. W3C Recommendation.
- [51] Alistair Miles and Sean Bechhofer (editors). 2009. SKOS Simple Knowledge Organization System Reference. World Wide Web Consortium. URL <http://www.w3.org/TR/skos-reference/>. W3C Recommendation.
- [52] Dan I. Moldovan and Rada Mihalcea. 2000. Using WordNet and Lexical Operators to Improve Internet Searches. *IEEE Internet Computing* 4, no. 1, pages 34–43.
- [53] Eetu Mäkelä, Kim Viljanen, Olli Alm, Jouni Tuominen, Onni Valkeapää, Tomi Kauppinen, Jussi Kurki, Reetta Sinkkilä, Teppo Kansälä, Robin Lindroos, Osma Suominen, Tuukka Ruotsalo, and Eero Hyvönen. 2007. Enabling the Semantic Web with Ready-to-Use Web Widgets. In: Lyndon J. B. Nixon, Roberta Cuel, and Claudio Bergamini (editors), *Proceedings of the Workshop on First Industrial Results of Semantic Technologies, co-located with ISWC 2007 + ASWC 2007, Busan, Korea, November 11th, 2007*, volume 293 of *CEUR Workshop Proceedings*, pages 56–69. CEUR-WS.org.
- [54] Eyal Oren, Renaud Delbru, and Stefan Decker. 2006. Extending Faceted Navigation for RDF Data. pages 559–572.
- [55] Jacco Van Ossenbruggen, Alia Amin, and Michiel Hildebr. 2008. Why Evaluating Semantic Web Applications Is Difficult. In: [17].
- [56] David Parry. 2004. A fuzzy ontology for medical document retrieval. In: *ACSW Frontiers '04: Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation*, pages 121–126. Australian Computer Society, Inc., Darlinghurst, Australia.

- [57] Adam Pease, Ian Niles, and John Li. 2002. The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications. In: Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web.
- [58] Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee. 2007. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems* 24, no. 3, pages 45 – 77.
- [59] Frederik Pfisterer, Markus Nitsche, Anthony Jameson, and Catalin Barbu. 2008. User-Centered Design and Evaluation of Interface Enhancements to the Semantic MediaWiki. In: [17].
- [60] A. Steven Pollitt. 1998. The key role of classification and indexing in view-based searching. *International Cataloguing and Bibliographic Control* 27, no. 2.
- [61] Dennis Quan, Karun Bakshi, David Huynh, and David R. Karger. 2003. User Interfaces for Supporting Multiple Categorization. In: Matthias Rauterberg, Marino Menozzi, and Janet Wesson (editors), *Human-Computer Interaction INTERACT '03: IFIP TC13 International Conference on Human-Computer Interaction, 1st-5th September 2003, Zurich, Switzerland*. IOS Press. ISBN 1-58603-363-8.
- [62] Dennis Quan, David Huynh, and David R. Karger. 2003. Haystack: A Platform for Authoring End User Semantic Web Applications. In: [21], pages 738–753.
- [63] Dave Reynolds, Paul Shabajee, and Steve Cayzer. 2004. Semantic information portals. In: *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 290–291. ACM, New York, NY, USA. ISBN 1-58113-912-8.

- [64] Cristiano Rocha, Daniel Schwabe, and Marcus Poggi Aragao. 2004. A hybrid approach for searching in the semantic web. In: WWW '04: Proceedings of the 13th international conference on World Wide Web, pages 374–383. ACM, New York, NY, USA. ISBN 1-58113-844-X.
- [65] Tuukka Ruotsalo and Eero Hyvönen. 2007. An Event-Based Approach for Semantic Metadata Interoperability. In: Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux (editors), The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007, volume 4825 of *Lecture Notes in Computer Science*, pages 409–422. Springer. ISBN 978-3-540-76297-3.
- [66] mc schraefel, Jennifer Golbeck, Duane Degler, Abraham Bernstein, and Lloyd Rutledge. 2008. Semantic web user interactions: exploring hci challenges. In: CHI '08: CHI '08 extended abstracts on Human factors in computing systems, pages 3929–3932. ACM, New York, NY, USA. ISBN 978-1-60558-012-X.
- [67] mc schraefel, Max Wilson, Alistair Russell, and Daniel A. Smith. 2006. mSpace: improving information access to multimedia domains with multi-modal exploratory search. *Communications of the ACM* 49, no. 4, pages 47–49.
- [68] Abigail J. Sellen, Rachel Murphy, and Kate L. Shaw. 2002. How knowledge workers use the web. In: CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems, pages 227–234. ACM, New York, NY, USA. ISBN 1-58113-453-3.
- [69] Teemu Sidoroff and Eero Hyvönen. 2005. Semantic E-government Portals - A Case Study. In: Proceedings of the ISWC-2005 Workshop Semantic Web Case Studies and Best Practices for eBusiness SWCASE05.

- [70] Shailendra Singh, Lipika Dey, and Muhammad Abulaish. 2004. A Framework for Extending Fuzzy Description Logic to Ontology Based Document Processing. In: Jesús Favela, Ernestina Menasalvas Ruiz, and Edgar Chávez (editors), *Advances in Web Intelligence, Second International Atlantic Web Intelligence Conference, AWIC 2004, Cancun, Mexico, May 16-19, 2004*. Proceedings, volume 3034 of *Lecture Notes in Computer Science*, pages 95–104. Springer. ISBN 3-540-22009-7.
- [71] Reetta Sinkkilä, Eetu Mäkelä, Tomi Kauppinen, and Eero Hyvönen. 2008. Combining Context Navigation with Semantic Autocompletion to Solve Problems in Concept Selection. In: Khalid Belhajjame, Mathieu d’Aquin, Peter Haase, and Paolo Missier (editors), *First International Workshop on Semantic Metadata Management and Applications, SeMMA 2008, Located at the Fifth European Semantic Web Conference (ESWC 2008), Tenerife, Spain, June 2nd, 2008*. Proceedings, volume 346 of *CEUR Workshop Proceedings*, pages 61–68.
- [72] Osma Suominen, Eero Hyvönen, Kim Viljanen, and Eija Hukka. 2009. HealthFinland—A national semantic publishing network and portal for health information. *Web Semantics: Science, Services and Agents on the World Wide Web* 7, no. 4, pages 287 – 297. Semantic Web challenge 2008.
- [73] Jaime Teevan, Christine Alvarado, Mark S. Ackerman, and David R. Karger. 2004. The perfect search engine is not enough: a study of orienteering behavior in directed search. In: *CHI ’04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 415–422. ACM, New York, NY, USA. ISBN 1-58113-702-8.
- [74] Onni Valkeapää, Olli Alm, and Eero Hyvönen. 2007. An Adaptable Framework for Ontology-based Content Creation on the Semantic Web. *Journal of Universal Computer Science* 13, no. 12, pages 1835–1835.
- [75] Kim Viljanen, Teppo Käsälä, Eero Hyvönen, and Eetu Mäkelä. 2006. ONTODELLA—A Projection and Linking Service for Semantic Web Appli-

- cations. In: DEXA '06: Proceedings of the 17th International Conference on Database and Expert Systems Applications, pages 370–376. IEEE Computer Society, Washington, DC, USA. ISBN 0-7695-2641-1.
- [76] Kim Viljanen, Jouni Tuominen, and Eero Hyvönen. 2009. Ontology Libraries for Production Use: The Finnish Ontology Library Service ONKI. In: Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvönen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Paslaru Bontas Simperl (editors), *The Semantic Web: Research and Applications*, 6th European Semantic Web Conference, ESWC 2009, Heraklion, Crete, Greece, May 31-June 4, 2009, Proceedings, volume 5554 of *Lecture Notes in Computer Science*, pages 781–795. Springer. ISBN 978-3-642-02120-6.
- [77] Tom D. Wilson. 1982. Current awareness services and their value in local government. *Journal of Librarianship and Information Science* 14, no. 4, pages 279–288.
- [78] Tom D. Wilson. 1994. Information needs and uses: fifty years of progress. In: Brian Campbell Vickery (editor), *Fifty years of information progress: a Journal of Documentation review*, pages 15–51. Aslib, London.
- [79] Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti A. Hearst. 2003. Faceted metadata for image search and browsing. In: *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–408. ACM, New York, NY, USA. ISBN 1-58113-630-7.
- [80] Junliang Zhang and Gary Marchionini. 2005. Evaluation and evolution of a browse and search interface: Relation Browser++. In: *dg.o2005: Proceedings of the 2005 national conference on Digital government research*, pages 179–188. Digital Government Research Center.
- [81] Lei Zhang, Yong Yu, Jian Zhou, Chenxi Lin, and Yin Yang. 2005. An enhanced model for searching in semantic portals. In: Allan Ellis and Tatsuya Hagino

(editors), Proceedings of the 14th international conference on World Wide Web, WWW 2005, Chiba, Japan, May 10-14, 2005, pages 453–462. ACM. ISBN 1-59593-046-9.



I

Publication I

Eetu Mäkelä, Eero Hyvönen, Samppa Saarela and Kim Viljanen. 2004. OntoViews – A Tool for Creating Semantic Web Portals. In: Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen (editors), *The Semantic Web - ISWC 2004: Third International Semantic Web Conference, Hiroshima, Japan, November 7-11, 2004. Proceedings*, volume 3298 of *Lecture Notes in Computer Science*, pages 797–811. Springer. ISBN 3-540-23798-4.

© 2004 Springer Verlag Berlin Heidelberg

ONTOVIEWS

– A Tool for Creating Semantic Web Portals

Eetu Mäkelä, Eero Hyvönen, Samppa Saarela, and Kim Viljanen

Helsinki Institute for Information Technology (HIIT), University of Helsinki
P.O. Box 26, 00014 UNIV. OF HELSINKI, FINLAND
{Firstname.Lastname}@cs.Helsinki.FI
<http://www.cs.helsinki.fi/group/seco/>

Abstract. This paper presents a semantic web portal tool ONTOVIEWS for publishing RDF content on the web. ONTOVIEWS provides the portal designer with a content-based search engine server, Ontogator, and a link recommendation system server, Ontodella. The user interface is created by combining these servers with the Apache Cocoon framework. From the end-user's viewpoint, the key idea of ONTOVIEWS is to combine the multi-facet search paradigm, developed within the information retrieval research community, with semantic web RDFS ontologies, and extend the search service with a semantic browsing facility based on ontological reasoning. ONTOVIEWS is presented from the viewpoints of the end-user, architecture, and implementation. The implementation described is modular, easily modified and extended, and provides a good practical basis for creating semantic portals on the web. As a proof of concept, application of ONTOVIEWS to a deployed semantic web portal is discussed.

Keywords: Semantic web, information retrieval, multi-facet search, view-based search, recommendation system.

1 Introduction

Much of the semantic web content will be published using semantic portals¹ [1]. Such portals typically provide the end-user with two basic services: 1) a search engine based on the semantics of the content [2] and 2) dynamic linking between pages based on the semantic relations in the underlying knowledge base [3].

This paper presents ONTOVIEWS — a tool for creating semantic portals — that facilitates these two services combined with user interface and Web Services functionality inside the Apache Cocoon framework². The search service of ONTOVIEWS is based on the idea of combining multi-facet search [4,5] with RDFS ontologies. The dynamic linking service is based on logical rules that define associations of interest between RDF(S) resources. Such associations are rendered on the user interface as links with labels that explain the nature the semantic linkage to the end-user.

These ideas were initially developed and tested as a stand-alone Java application [6] and as a Prolog-based HTML generator [7]. ONTOVIEWS combines and extends these

¹ See, e.g., <http://www.ontoweb.org/>.

² <http://cocoon.apache.org/>

two systems by separating the search and link recommendation services into independent servers (Ontogator and Ontodella) to be used on the Semantic Web, and by providing a clear logic-based interface by which ontology and annotation schema specific RDF(S) structures can be hidden from the servers.

In the following, ONTOVIEWS is first presented from the viewpoint of the end-user. We present an application developed using the tool: MUSEUMFINLAND³ [8]. This system is a deployed semantic portal for publishing heterogeneous museum collections on the Semantic Web. After this the architecture and the implementation of the framework are discussed mostly from the portal designer's viewpoint. In conclusion, benefits and limitations of the system are summarized, related work is discussed, and directions for further research are outlined.

2 ONTOVIEWS from the End-User's Perspective

ONTOVIEWS provides the end-user with a semantic view-based search engine and a recommendation system. In MUSEUMFINLAND, these services are provided to the end-user via two different user interfaces: one for desktop computers and one for mobile devices. In below the desktop computer web interface is first presented.

2.1 A Multi-facet Search Engine

The search engine of ONTOVIEWS is based on the multi-facet search paradigm [4,5]. Here the concepts used for indexing are called *categories* and are organized systematically into a set of hierarchical, orthogonal taxonomies. The taxonomies are called subject *facets* or *views*. In multi-facet search the views are exposed to the end-user in order to provide her/him with the right query vocabulary and for presenting the repository contents and search results along different views.

In MUSEUMFINLAND, the content consists of collections of cultural artifacts and historical sites in RDF format consolidated from several heterogeneous Finnish museum databases [9]. The RDF content is annotated using a set of seven ontologies. From the seven ontologies, nine view-facets are created. The ontologies underlying the application consist of some 10,000 RDF(S) classes and individuals, half of which are in use in the current version on the web. There are some 7,600 categories in the views and 4,500 metadata records of collection artifacts and old cultural sites in Finland.

Figure 1 shows the search interface of MUSEUMFINLAND. The nine facet hierarchies, such as Artifact (“Esinetyypit”) and Material (“Materiaalit”), are shown (in Finnish) on the left column. For each facet hierarchy, the next level of sub-categories is shown as links. A query is formulated by selecting a category by clicking on its name. When the user selects a category c in a facet f , the system constrains the search by leaving in the result set only such objects that are annotated in facet f with some sub-category of c or c itself. The result set is shown on the right grouped by the sub-categories of the last selection. In this case the user has selected “Tools”, whose sub-categories include Textile making tools (“tekstiilityövälineet”) and Tools of folk medicine (“kansanlääkinnän työvälineet”).

³ <http://museosuomi.cs.helsinki.fi/>

The screenshot shows the MuseoSuomi website interface. At the top, there is a search bar with the text "Suolahaku:" and a "Hae" button. Below the search bar, there are several facets (filters) on the left side, each with a list of items and their counts. The facets include:

- Eisintyyppi (koko luokittelu) (työvälineet)**: 193 items
- Materiaali (koko luokittelu) (työvälineet)**: 241 items
- Valmistaja (koko luokittelu) (työvälineet)**: 38 items
- Valmistuspaikka (koko luokittelu) (työvälineet)**: 84 items
- Valmistusaika (koko luokittelu) (työvälineet)**: 89 items
- Käyttäjät (koko luokittelu) (työvälineet)**: 54 items
- Käyttöpaikka (koko luokittelu) (työvälineet)**: 71 items
- Käyttötilanne (koko luokittelu) (työvälineet)**: 179 items
- Kokoelma (koko luokittelu) (työvälineet)**: 193 items

The main content area displays a grid of tool images with their respective IDs and category names. The tools shown include:

- kehräpuku, kuosaak (NBA SU4527 50)**
- kehruslaista, kehräpuku, kuosaak (NBA SU5069 26)**
- runkolapa (ECM 100 1)**
- smelldie, värinainlappi, värinnyöyry (NBA SU2449 7)**
- suoraraista puunemilöntäntäruuta (ECM 2711 1)**
- nappikoukku napituskoukku (ECM 3594 264)**
- kiekkamälldi, komsiolaha (NBA SU4922 32)**
- palohepat palohosat (ECM 614 1)**
- luonnilasta (NBA SU4135 166)**

Fig. 1. MUSEUMFINLAND search interface after selecting the category link Tools (“työvälineet”).

Hits in different categories are separated by horizontal bars and can be paged through independently in each category.

When answering the query, the result sets resulting from the selection of each category seen on the screen are recomputed, and a number (n) is shown to the user after the category name. It tells that if the category is selected next, then there will be n hits in the result set. For example, in figure 1, the number 193 in the Collection facet (“Kokoelma”) on the bottom tells that there are 193 tools in the collections of the National Museum (“Kansallismuseon kokoelmat”). A selection leading to an empty result set ($n = 0$) is removed from its facet (or alternatively disabled and shown grayed out, depending on the user’s preference). In this way, the user is hindered from making a selection leading to an empty result set, and is guided toward selections that are likely to constrain the search appropriately. The query can be relaxed by making a new selection on a higher level in the facet or by dismissing the facet totally from the query.

Above, the category selection was made among the direct sub-categories listed in the facets. An alternative way is to click on the link Whole facet (“koko luokittelu”) on a facet. The system then shows all possible selections in the whole facet hierarchy with hit counts. For example, in figure 2 the user selected in the situation of figure 1 the link Whole facet of the facet Time of Creation (“Valmistusaika”). The system shows how the

Uusi haku | Takaisin hakusivulle | Ohjeet | Näytä kaikki kategoriat | Tietoa ohjelmasta | MuseoSuomi-palautte

Hakuehdot

Kategoria: Esmetyyppi > työvälineet (ryhmittele kohteet) (poista)

Kategoria: Valmistusaika

- [aikakaudet](#) (90)
 - [eshistoriallinen aika](#) (1)
 - [kivkausi](#) (1)
 - [historiallinen aika](#) (89)
 - [uusikausi](#) (89)
 - [sodat](#) (19)
 - [I maailmansota](#) (11)
 - [II maailmansota](#) (9)
- [vuosisadat](#) (89)
 - [1700-luku](#) (3)
 - [1750-1799](#) (1)
 - [1800-luku](#) (20)
 - [1800-1809](#) (3)
 - [1810-1819](#) (2)
 - [1820-1829](#) (3)
 - [1830-1839](#) (4)
 - [1840-1849](#) (2)
 - [1850-1859](#) (5)
 - [1860-1869](#) (5)
 - [1870-1879](#) (6)
 - [1880-1889](#) (11)
 - [1890-1899](#) (12)
 - [1900-luku](#) (76)
 - [1900-1909](#) (15)
 - [1910-1919](#) (14)
 - [1920-1929](#) (12)
 - [1930-1939](#) (16)
 - [1940-1949](#) (8)
 - [1950-1959](#) (30)
 - [1960-1969](#) (15)
 - [1970-1979](#) (5)
 - [1980-1989](#) (3)
 - [1990-1999](#) (1)

Fig. 2. The Time facet hierarchy classifying the result set of tools in figure 1.

tools in the current result set are classified according to the selected facet. This gives the user a good overview of the distribution of items over a desired dimension. With the option of graying out categories with no hits, it is also immediately obvious where the collections are lacking artifacts.

When the user is capable of expressing her information need straightforward in terms of keywords, then a Google-like keyword search interface is usually preferred. ONTO-VIEWS seamlessly integrates this functionality in the following way: First, the search keywords from the search form are matched against category names in the facets. A new dynamic facet is created in the user interface, containing all facet categories matching the keyword shown with the corresponding facet name. Second, a result set of object hits is shown. This result set contains all objects contained in any of the categories matched in addition to all objects whose metadata directly contains the keyword, grouped by the categories found. This way, the keyword search also solves the search problem of finding relevant categories in facets that may contain thousands of categories.


Sanahaku: Hae tarkenna hakua

Hakusana: esp (poista)
 Valmistuspaikka > ... > [Espanja](#) (4),
 Valmistuspaikka > ... > [Espoo](#) (140),
 Käyttöpaikka > ... > [Espoo](#) (512),
 Käyttäjä > ... > [Espoon kaupunginmuseo](#) (1),
 Kokoelma > ... > [Espoon kaupunginmuseon kokoelmat](#) (1190),
 Valmistaja > ... > [tekninen virasto Espoon kaupunki](#) (1),
 Käyttäjä > ... > [tekninen virasto Espoon kaupunki](#) (1),
 Valmistaja > ... > [Espoon partiotuki](#) (1)


Esinetyyppi (koko luokitella) (ryhmittele kohteet)
[taideteokset](#) (3), [asetit ja ampumatarvikkeet](#) (1),
[asiat ja taloustarvikkeet](#) (90), [henkilökohtaiset esineet](#) (59),
[ulkaisen tilan esineet](#) (16),
[kulkuneuvot ja kuljetusvälineet](#) (24),
[laitteet ja koneet osineen](#) (25), [luokittelemattomat esineet](#) (62),
[maatalous- ja karjanhoitovälineet](#) (3),
[puhtaanapitoon käytettävät esineet](#) (13),
[pukineet ja muut tekstiilit](#) (352), [sällytetyt osineet](#) (242),
[ulkokalusteet ja pihatarvikkeet](#) (4),
[ns 18 maoryhmanurheiluvälineet ja pelivälineet ja leikkikalut](#) (17),
[valaisuihin käytettävät esineet](#) (26),
[yhteisölliset esineet](#) (9), [käsityöt](#) (3),
[leikkikalut](#) (146), [pynnälvälineet](#) (8),
[sisustus](#) (141), [näytty kaikki 22](#)

Materiaali (koko luokitella) (ryhmittele kohteet)
[materiaalit](#) (1089)

Hakuehdot
Hakusana: esp (ryhmittele kohteet) (poista)
 Kohteet ryhmiteltyinä hakusanan *esp* mukaisesti
 (näytä ilman ryhmitelyä)
 Valmistuspaikka > [Espanja](#), kohteet 1-4/4 (ryhmittele kohteet)




pullo: appelsiinihivistepullo (ECM
3675 10)




Jalkineet, naisen sandaalit (LKM
LHM LHM ES 94108 377)

Valmistuspaikka > [Espoo](#), kohteet 1-4/140 (ryhmittele kohteet)



kukkaro tupakkakukkaro (ECM
3594 290)



ritsaritsa (ECM 3343 2)

Fig. 3. Using the keyword search for finding categories.

A sample keyword search is shown in figure 3. Here, a search for “esp” has matched, for example, the categories Spain (“Espanja” in Finnish) and Espoo in the facet Location of Creation and the category Espoo City Museum (“Espoon kaupunginmuseo”) in the facet User (“Käyttäjät”). The categories found can be used to constrain the multi-facet search as normal, with the distinction that selections from the dynamic facet replace selections in their corresponding facets and dismiss the dynamic facet.

2.2 The Item View with Semantic Links

At any point during multi-facet search the user can select any hit found by clicking on its image. The corresponding data object is then shown as a web page, such as the one in figure 4. The example depicts a special part, distaff (“rukinlapa” in Finnish) used in a spinning wheel. The page contains the following information and links:

1. On top, there are links to directly navigate in the groups and results of the current query.
2. The image(s) of the object is (are) depicted on the left.
3. The metadata of the object is shown in the middle on top.
4. All facet categories that the object is annotated with are listed in the middle bottom as hierarchical link paths. A new search can be started by selecting any category there.
5. A set of semantic links on the right provided by a semantic recommendation system.

The semantic links on the right reveal to the end-user a most interesting aspect of the collection items: the implicit semantic relations that relate collection data with their context and each other. The links provide a *semantic browsing* facility to the end-user.

The screenshot shows the MuseoSuomi website interface. At the top, there are logos for the Finnish Institute for Cultural Heritage Technology and the University of Helsinki. The main header reads 'MuseoSuomi - Suomen museot semanttisessa webissä'. Below this, there are navigation options for 'Uusi laulu', 'Tähtäin-haku', 'Objekt', 'Tietoa objekteista', and 'MuseoSuomi-palvelu'. There are also filters for 'Elokuu (180)*', 'Ruhalle (1)', and 'Joulukuun (0)'. The main content area is titled 'ruukinlaipa' and features an image of a wooden distaff. To the right of the image, there is a detailed metadata section including 'Vahvistuspaikka: Suomi', 'Vahvistusajka: 1793', 'Käyttöpäikka: Suomi', 'Aiasaana: KEEHUU, KORISTEVEISTO, PUUMERKKI, VUOSILUKU', 'Museokokoelma: MuseoKokoelma', 'Vastuualue: Espoon kaupungin museo', 'Aiasamateriaali: Espoon kaupungin museo raa'nto', 'Esimen numero: ECM1001', and 'ID: 1001'. Below the metadata, there are several facet categories with recommendation links: 'Esimen nimi', 'Vahvistuspaikka', 'Vahvistusajka', 'Käyttöpäikka', 'Käyttökohde', and 'Kokoonlaatu'. Each facet category has a list of items with their respective IDs and titles. On the right side of the page, there are two sections: 'Sama käyttöpäikka' and 'Esimeneseen liittyvään paikkaan liittyii nimimäärittäjä'. The 'Sama käyttöpäikka' section lists various locations like 'Espoo' and 'Suomi'. The 'Esimeneseen liittyvään paikkaan liittyii nimimäärittäjä' section lists various events and activities like 'Espoo', 'Suomi', and 'Suomen'. The 'Kokoonlaatu' section lists various materials like 'Kokoonlaatu' and 'Kokoonlaatu'.

Fig. 4. Web page depicting a collection object, its metadata, facet categories, and semantic recommendation links to other collection object pages.

For example, in figure 4 there are links to objects used at the same location (categorized according to the name of the common location), to objects related to similar events (e.g., objects used in spinning, and objects related to concepts of time, because the distaff in question has a year carved onto it), to objects manufactured at the same time, and so on. Since a decoratively carved distaff used to be a typical wedding gift in Finland, it is also possible to recommend links to other objects related to the wedding event, such as wedding rings. In ONTOVIEWS, such associations can be exposed to the end-user as link groups whose titles and link names explain to the user the reason for the recommendation.

2.3 The Mobile User Interface

Using ONTOVIEWS the same content and services can easily be rendered to the end-users in different ways. To demonstrate this, we created another user interface for MUSEUM-FINLAND to be used by WAP 2.0 (XHTML/MP) compatible devices. ONTOVIEWS is a particularly promising tool for designing a mobile user interface due to the following reasons. Firstly, empty results can be eliminated, which is a nice feature in an environment where data transfer latencies and costs are still often high. Secondly, the elimination of infeasible choices makes it possible to use the small screen size more efficiently for displaying relevant information. Thirdly, the semantic browsing functionality is a simple and effective navigation method in a mobile environment.

The mobile interface repeats all functionality of the PC interface, but in a layout more suitable to the limited screen space of mobile devices. In addition, to better facilitate

finding interesting starting points for browsing, some mobile-specific search shortcuts were created. The search results are shown first up front noting the current search parameters for easy reference and dismissal, as seen in figure 5. Below this, the actual search facets are shown. In the mobile user interface selectable sub-categories are not shown as explicit links as in the PC interface, but as drop-down lists that replace the whole view when selected. This minimizes screen space usage while browsing the facets, but maximizes usability when selecting sub-categories from them. In-page links are provided for quick navigation between search results and the search form.

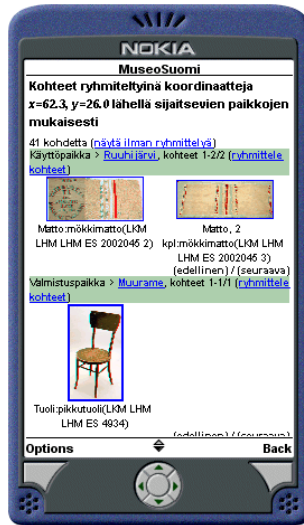


Fig. 5. Results of a mobile geolocation search initiated somewhere near Ruuhijärvi, Finland.

The item page (corresponding to figure 4) is organized in a similar fashion, showing first the item name, images, metadata, annotations, semantic recommendations, and finally navigation in the search result. There are also in-page links for jumping quickly between the different parts of the page.

The mobile user interface also provides two distinct services aimed specifically for mobile use. Firstly, the interface supports search by the geolocation of the mobile device in the same manner as in the concept-based ONTOVIEWS keyword search. Any entries in the Location ontology near the current location of the mobile user are shown in a dynamic facet as well as all data objects made *or* used in any of these locations. In addition, any objects directly annotated with geo-coordinates near the mobile user are shown grouped as normal. This feature gives the user a one-click path to items of likely immediate interest. Secondly, because navigation and search with mobile devices is tedious, any search state can be “bookmarked”, sent by email to a desired address, and inspected later in more detail by using the more convenient PC interface.

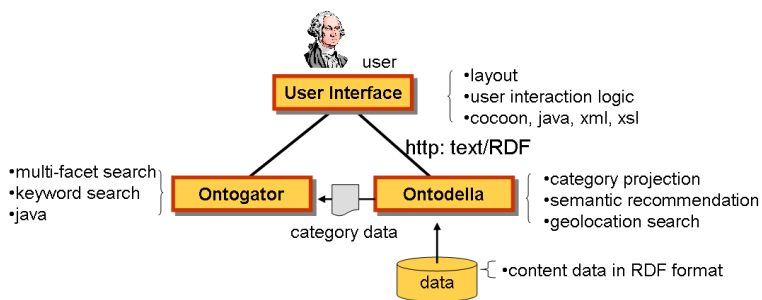


Fig. 6. The components of ONTOVIEWS.

3 Architecture and Implementation

ONTOVIEWS consists of the three major components shown in figure 6. The logic server of ONTOVIEWS, Ontodella, provides the system with reasoning services, such as category generation and semantic recommendations. It is based on the HTTP server version of SWI-Prolog⁴. It has also been extended to provide simple point-of-interest search based on geo-coordinates. It is queried via a HTTP connection.

The multi-facet search engine of ONTOVIEWS, Ontogator, is a generic view-based RDF search engine. It defines and implements an RDF-based query interface that is used to separate view-based search logic from the user interface. The interface is defined as an OWL ontology⁵. It can be used to query for category hierarchies and items grouped by and/or constrained by these. Both categories and items can also be queried using keywords. Given a set of category and/or keyword-based constraints, Ontogator filters categories that would lead to an empty result set. Alternatively, filtering the categories, for example by graying out, can be left for the user interface. This is possible since Ontogator (optionally) tags every category with a number of hits. There are also a number of other general options (e.g. accepted language) and restrictions (e.g. max items/categories returned) that can be used, for example, to page the results by categories and/or items. Ontogator replies to queries in RDF/XML that has a fixed structure. Since the search results are used in building the user interface, every resource is tagged with an `rdfs:label`.

The third component in figure 6, User Interface, binds the services of Ontogator and Ontodella together, and is responsible for the user interfaces and interaction. This component is built on top of the Apache Cocoon framework⁶. Cocoon is a framework based wholly on XML and the concept of pipelines constructed from different types of components, as illustrated in figure 7. A pipeline always begins with a generator, that generates an XML-document. Then follow zero or more transformers that take an XML-document as input and output a document of their own. The pipeline always ends

⁴ <http://swi-prolog.org/>

⁵ <http://www.cs.helsinki.fi/group/seco/ns/2004/03/ontogator#>

⁶ <http://cocoon.apache.org/>

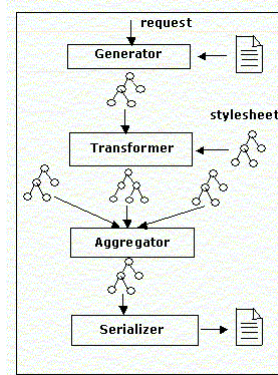


Fig. 7. The components of a Cocoon pipeline.

in a serializer that serializes its input into the final result, such as an HTML-page, a PDF-file, or an image. It is also possible for the output of partial pipelines to be combined via aggregation into a single XML-document for further processing. Execution of these pipelines can be tied to different criteria, e.g. to a combination of the request URI and requesting user-agent.

In ONTOVIEWS, all of the intermediate components produce not only XML, but valid RDF/XML. Figure 8 depicts two pipelines of the ONTOVIEWS system. The pipe lines look alike, but result in quite different pages, namely in the search result page seen in figure 1 (and another similar page used for depicting results of the keyword search), and in the item page seen in figure 4. This is due to the modular nature of the pipelines, which makes it possible to split a problem into small units and reuse components.

Every pipeline that is tied to user interaction web requests begins with a user state generator that generates an RDF/XML representation of the user’s current state. While browsing, the state is encoded wholly in the request URL, which allows for easy bookmarking and also furthers the possibilities of using multiple servers. This user state is then combined with system state information in the form of facet identifiers and query hit counts, and possible user geolocation based information. This information is then transformed into appropriate queries for the Ontogator and Ontodella servers depending on the pipeline.

In the Search Page pipeline on the left, an Ontogator query returning grouped hits and categories is created. In the Item Page pipeline on the right, Ontogator is queried for the properties and annotations of a specific item and its place in the result set, while Ontodella is queried for the semantic links relating to that item. The Ontogator search engine is encapsulated in a Cocoon transformer, while the Ontodella transformer is actually a generic Web Services transformer that creates a HTTP-query from its input, executes it, and creates SAX events from the HTTP-response. The RDF/XML responses from the search engines are then given to user interface transformers depending on the pipeline and the device that originated the request. These transform the results into appropriate XHTML or to any other format, which is then run through an internationalization transformer for language support and serialized. Most of the transformations into queries

and XHTML are implemented with simple XSLT-stylesheets. In this way, changes to layout are very simple to implement, as is the creation of new interfaces for different media. The mobile interface to MUSEUMFINLAND discussed earlier was created in this way quite quickly.

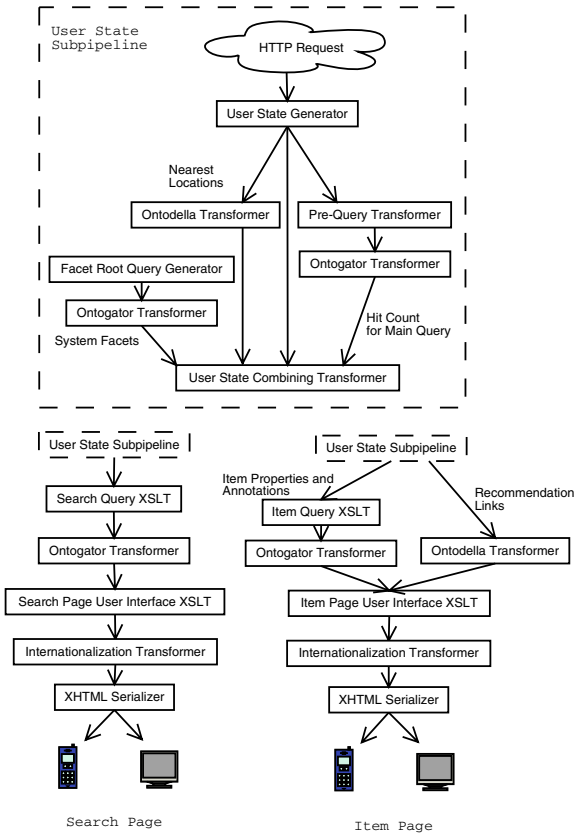


Fig. 8. Two Cocoon pipelines used in ONTOVIEWS.

All of the transformer components can also be made available for use in other web applications as Web Services, by creating a pipeline that generates XML from an HTTP-query and returns its output as XML. In this way, other web applications could make use of the actual RDF-data contained in the system, querying the Ontogator and Ontodella servers directly for content data in RDF/XML-format. It also provides a way of distributing the processing in the system to multiple servers. For example, ONTOVIEWS instances running Ontogator could be installed on multiple servers, and a main ONTOVIEWS handling user interaction could distribute queries among these servers in a round-robin fashion to balance load.

4 Adapting ONTOVIEWS to New Data

ONTOVIEWS is capable of supporting any RDF-based input data (e.g., OWL). To adapt the system to new data, the following steps must be taken: First, create rules describing how categories are generated and items connected to them for the view based search. Second, create rules describing how links are generated for the recommendations. Third, possibly change the layout templates.

In the following, we describe how the logic rules needed are defined in our system using Prolog. The layout templates are straightforward XSLT and will not be discussed.

4.1 Category View Generation

A view is a hierarchical index-like decomposition of category resources where each category is associated with a set of subcategories and data items. A view is defined in Ontodella, the logic server of ONTOVIEWS, by specifying a view predicate called `ontodella_view` with the following information: 1) the root resource URI, 2) a binary subcategory relation predicate, and 3) a binary relation predicate that maps the hierarchy categories with the items used as leaves in the view. In addition, each view must have a label.

An example⁷ of a view predicate is given below:

```
ontodella_view(
  'http://www.cs.helsinki.fi/seco/ns/2004/03/places#earth',
  place_sub_category,
  place_of_use_leaf_item,
  [fi:'K"{a}ytt"{o}paikka', en:'Place of Use'] % the labels
).
```

Here the URI on the second line is the root resource, `place_sub_category` is the name of the subcategory relation predicate and `place_of_use_leaf_item` is the leaf predicate. The label list contains the labels for each supported language. In our case, we support both Finnish (fi) and English (en).

The binary subcategory predicate can be based, e.g., on a containment property in the following way:

```
place_sub_category( ParentCategory, SubCategory ) :-
  SubCategoryProperty =
  'http://www.cs.helsinki.fi/seco/ns/2004/03/places#isContainedBy',
  rdf( SubCategory, SubCategoryProperty, ParentCategory ).
```

The leaf predicate describes when a given resource item is a member of the given category. For example, `place_of_use_leaf_item` in our example above can be described as follows:

```
place_of_use_leaf_item( ResourceURI, CategoryURI ) :-
  Relation = 'http://www.cs.helsinki.fi/seco/ns/2004/03/artifacts#usedIn',
  rdf( ResourceURI, Relation, CategoryURI ).
```

⁷ The syntax is slightly simplified due to presentation reasons. We use SWI-Prolog (<http://www.swi-prolog.org>) as the inference engine and SWI-Prolog syntax in the examples.

Based on these rules, the categories can be generated by iterating through the predicate `ontodella_view`, and by recursively creating the category hierarchies using the subcategory rules starting from the given root category. At every category, all relevant leaf resources are attached to the category based on the leaf rules. When the categories have been generated, they can be navigated using the Ontogator module presented earlier.

4.2 Recommendation Link Generation

Link generation is based on rules that describe when two resources should be linked. Each link rule can be arbitrary complex and is defined by a domain specialist. A linking rule is described by a predicate of the form $p(\textit{SubjectURI}, \textit{TargetURI}, \textit{Explanation})$ that should succeed with the two resources *SubjectURI* and *TargetURI* are to be linked. The variable *Explanation* is then bound to an explanatory label (string) for the link. In the following, one of the more complex rules — linking items related to a common event — is presented as an example:

```
related_by_event( Subject, Target, Explanation ) :-
  ItemTypeProperty =
    'http://www.cs.helsinki.fi/seco/ns/2004/03/artifacts#item_type',
  ItemTypeToEventRelatingProperty =
    'http://www.cs.helsinki.fi/seco/ns/2004/03/mapping#related_to_event',

  % check that both URIs correspond in fact to artifacts
  isArtifact(Subject),
  isArtifact(Target),
  % and are not the same
  Subject \= Target,

  % find all the item types the subject item belongs to
  rdf(Subject, ItemTypeProperty, SubjectItemType),
  rdfs_transitive_subClassOf(SubjectItemType, SubClassOfSubjectItemType),

  % find all the events any of those item types are related to
  rdf(SubClassOfSubjectItemType, ItemTypeToEventRelatingProperty, Event),
  % and events they include or are part of
  (
    rdfs_transitive_subClassOf(Event, SubOrSuperClassOfEvent),
    DescResource=TransitiveSubOrSuperClassOfEvent;
    % or
    rdfs_transitive_subClassOf(SubOrSuperClassOfEvent, Event),
    DescResource=Event;
  ),

  % find all item types related to those events
  rdf(TargetItemType, ItemTypeToEventRelatingProperty, SubOrSuperClassOfEvent),
  % and all their superclasses
  rdfs_transitive_subClassOf(SuperClassOfTargetItemType, TargetItemType),

  % don't make uninteresting links between items of the same type
  SuperClassOfTargetItemType \= SubjectItemType,
  not(rdfs_transitive_subClassOf(SuperClassOfTargetItemType, SubjectItemType)),
  not(rdfs_transitive_subClassOf(SubjectItemType, SuperClassOfTargetItemType)),

  % finally, find all items related to the linked item types
  rdf(Target, ItemTypeProperty, SuperClassOfTargetItemType),

  list_labels([DescResource], RelLabel),
  Explanation=[commonResources(DescResource), label(fi:RelLabel)].
```

The rule goes over several ontologies, first discovering the object types of the objects, then traversing the object type ontology, relating the object types to events, and finally traversing the event ontology looking for common resources. Additional checks are made to ensure that the found target is an artifact and that the subject and target are not the same resources. Finally, information about the relation is collected, such as the URI and the label of the common resource, and the result is returned as the link label.

The links for a specific subject are generated when the ONTOVIEWS main module makes an HTTP query to the Ontodella. As a result, the Ontodella returns an RDF/XML message containing the link information (the target URI and the relation description). These links are then shown to the user using the User Interface (cf. figure 6).

5 Discussion

5.1 Benefits and Limitations

The example application MUSEUMFINLAND shows that the multi-facet search paradigm combined with ontologies is feasible as a basis for search on the Semantic Web. The paradigm is especially useful when the user is not just searching for a particular piece of information, but is interested in getting a broader view of the contents of the repository, and in browsing of the contents in the large. The addition of keyword-based searching complements multi-facet searching nicely, better addressing the search needs of people with a clear idea of what they want and with means of expressing it. Such a search can be integrated seamlessly into the user interaction logic and visualization of the user interface. It can be used to also solve the search problem of finding appropriate concepts in the ontologies to be used as a basis in multi-facet search. The semantic recommendation system provides further browsing possibilities by linking the items semantically with each other by relations that cannot be expressed with the hierarchical categorizations used in the multi-facet search.

The Cocoon-based implementation of the ONTOVIEWS is eminently portable, extendable, modifiable, and modular when compared to our previous test implementations. This flexibility is a direct result of designing the application around the Cocoon concepts of transformers and pipelines, in contrast to servlets and layout XSLT. The generality and flexibility of ONTOVIEWS has been verified in creating the mobile device interface for MUSEUMFINLAND. Furthermore, we have used ONTOVIEWS in the creation of a semantic yellow page portal [10], and (using a later version of the tool) a test portal based on the material of the Open Directory Project (ODP)⁸. These demonstrations are based on ontologies and content different from MUSEUMFINLAND. With the ODP material, the system was tested to scale up to 2.3 million data items and 275,000 view categories with search times of less than 5 seconds on an ordinary PC server.

The use of XSLT in most of the user interface and query transformations makes it easy to modify the interface appearance and to add new functionality. However, it has also led to some quite complicated XSLT templates in the more involved areas of user interaction logic, e.g., when (sub-)paging and navigating in the search result pages. In using XSLT with RDF/XML there is also the problem that the same RDF triple can

⁸ <http://www.dmoz.org/>

be represented in XML in different ways but an XSLT template can be only tied to a specific representation. In our current system, this problem can be avoided because the RDF/XML serialization formats used by each of the subcomponents of the system are known, but in a general web service environment, this could cause complications. However, the core search engine components of ONTOVIEWS would be unaffected even in this case because they handle their input with true RDF semantics.

When applying ONTOVIEWS, the category and linking rules are described by a domain specialist. She selects what should be shown in the views and what relations between data items should be presented to the user as semantical links. With the help of the rules, any RDF data can be used as input for ONTOVIEWS. The prize of the flexibility is that somebody must create the rules, which can be a difficult task if the input data is not directly suitable for generating the wanted projections and links.

5.2 Related Work

Much of the web user interface and user interaction logic in ONTOVIEWS and MUSEUM-FINLAND pertaining to multi-facet search is based on Flamenco [5]. In ONTOVIEWS, however, several extensions to this baseline have been added, such as the tree view of categories (figure 2), the seamless integration of concept-based keyword and geolocation search, extended navigation in the result set, and semantic browsing. The easy addition of these capabilities was made possible by basing the system on RDF. We feel it would have been much more difficult to implement the system by using the traditional relational database model. Our approach also permits ONTOVIEWS to load and publish any existing RDF data, once the rules defining the facet projections and semantic links are defined.

5.3 Future Work

ONTOVIEWS is a research prototype. The test with the material of the Open Directory Project has proved it quite scalable with regards to the amount of data, but the response-time of the system currently scales linearly with respect to the number of simultaneous queries. Further optimization work would be needed to make the system truly able to handle large amounts of concurrent users. The benefits observed applying the XML-based Cocoon pipeline concept to RDF handling has lead us to the question, whether it would be possible to create a true “Semantic Cocoon” based on RDF semantics and an RDF-transformation language.

References

1. A. Maedche, S. Staab, N. Stojanovic, R. Struder, and Y. Sure, “Semantic portal — the SEAL approach,” Institute AIFB, University of Karlsruhe, Germany, Tech. Rep., 2001.
2. S. Decker, M. Erdmann, D. Fensel, and R. Studer, “Ontobroker: Ontology based access to distributed and semi-structured unformation,” in *DS-8*, 1999, pp. 351–369, cite-seer.nj.nec.com/article/decker98ontobroker.html.
3. C. Goble, S. Bechhofer, L. Carr, D. D. Roure, and W. Hall, “Conceptual open hypermedia = the semantic web?” in *Proceedings of the WWW2001, Semantic Web Workshop, Hongkong*, 2001.

4. A. S. Pollitt, “The key role of classification and indexing in view-based searching,” University of Huddersfield, UK, Tech. Rep., 1998, <http://www.ifla.org/IV/ifla63/63polst.pdf>.
5. M. Hearst, A. Elliott, J. English, R. Sinha, K. Swearingen, and K.-P. Lee, “Finding the flow in web site search,” *CACM*, vol. 45, no. 9, pp. 42–49, 2002.
6. E. Hyvönen, S. Saarela, and K. Viljanen, “Application of ontology-based techniques to view-based semantic search and browsing,” in *Proceedings of the First European Semantic Web Symposium, May 10-12, Heraklion, Greece*. Springer-Verlag, Berlin, 2004.
7. E. Hyvönen, M. Holi, and K. Viljanen, “Designing and creating a web site based on RDF content,” in *Proceedings of WWW2004 Workshop on Application Design, Development, and Implementation Issues in the Semantic Web, New York, USA*. CEUR Workshop Proceedings, Vol-105, 2004, <http://ceur-ws.org>.
8. E. Hyvönen, M. Junnila, S. Kettula, E. Mäkelä, S. Saarela, M. Salminen, A. Syreeni, A. Valo, and K. Viljanen, “Finnish Museums on the Semantic Web. User’s perspective on MuseumFinland,” in *Proceedings of Museums and the Web 2004 (MW2004), Selected Papers, Arlington, Virginia, USA, 2004*, <http://www.archimuse.com/mw2004/papers/hyvonen/hyvonen.html>.
9. E. Hyvönen, M. Salminen, S. Kettula, and M. Junnila, “A content creation process for the Semantic Web,” in *Proceeding of OntoLex 2004: Ontologies and Lexical Resources in Distributed Environments, May 29, Lisbon, Portugal, 2004*.
10. M. Laukkanen, K. Viljanen, M. Apiola, P. Lindgren, and E. Hyvönen, “Towards ontology-based yellow page services,” in *Proceedings of WWW2004 Workshop on Application Design, Development, and Implementation Issues in the Semantic Web, New York, USA*. CEUR Workshop Proceedings, Vol-105, 2004, <http://ceur-ws.org>.



II

Publication II

Eero Hyvönen, Eetu Mäkelä, Mirva Salminen, Arttu Valo, Kim Viljanen, Samppa Saarela, Miikka Junnila, and Suvi Kettula. 2005. MuseumFinland – Finnish museums on the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web* 3, no. 2-3, pages 224 – 241. Selected Papers from the International Semantic Web Conference, 2004 - ISWC, 2004.

© 2005 Elsevier B.V.



MUSEUMFINLAND—Finnish museums on the semantic web

Eero Hyvönen*, Eetu Mäkelä, Mirva Salminen, Arttu Valo, Kim Viljanen,
Samppa Saarela, Miikka Junnila, Suvi Kettula

*Helsinki Institute for Information Technology (HIIT), University of Helsinki, and Helsinki University of Technology,
P.O. Box 5500, 02015 TTK Helsinki, Finland*

Received 24 May 2005; accepted 25 May 2005

Abstract

This article presents the semantic portal MUSEUMFINLAND for publishing heterogeneous museum collections on the Semantic Web. It is shown how museums with their semantically rich and interrelated collection content can create a large, consolidated semantic collection portal together on the web. By sharing a set of ontologies, it is possible to make collections semantically interoperable, and provide the museum visitors with intelligent content-based search and browsing services to the global collection base. The architecture underlying MUSEUMFINLAND separates generic search and browsing services from the underlying application dependent schemas and metadata by a layer of logical rules. As a result, the portal creation framework and software developed has been applied successfully to other domains as well. MUSEUMFINLAND got the Semantic Web Challenge Award (second prize) in 2004.

© 2005 Elsevier B.V. All rights reserved.

Keywords: Semantic web; Information retrieval; Multi-facet search; View-based search; Ontology; Recommendation system

1. Why museums on the semantic web?

A special characteristic of cultural collection databases is that they contain semantically rich information. Collection items have a history and are related in many ways to our environment, to the society, and to other collection items. For example, a chair may be made of oak and leather, may be of a certain

style, was designed by a famous designer, was manufactured by a certain company during a time period, was used in a certain building together with other pieces of furniture, and so on. Other collection items, locations, time periods, designers, companies, etc. can be related to the chair through their properties and implicitly constitute a complicated semantic network of associations. This semantic network is not limited to a single collection but spans over other related collections in other museums. The network of semantic associations can be extended to contents of other types in other organizations, as well.

* Corresponding author. Tel.: +358 9 4513362;
fax: +358 9 4513356.

E-mail address: eero.hyvonen@tkk.fi (E. Hyvönen).

URL: <http://www.cs.helsinki.fi/group/seco/>.

Much of the semantic web content will be published using semantic portals [24]. Such portals typically provide the end-user with two basic services: (1) a search engine based on the semantics of the content [2] and (2) dynamic linking between pages based on the semantic relations in the underlying knowledge base [6]. Semantic web technology¹ enables new possibilities when publishing museum collections on the web [15]:

- Collection interoperability in content: web languages, standards, and ontologies make it possible to make heterogeneous museum collections of different kind mutually interoperable. This enables, e.g., the creation of large inter-museum exhibitions.
- Intelligent applications: more versatile, user-friendly, and useful applications based on the semantics of the collections can be created.

To realize these ideas in practice, we have developed a semantic web portal called “MUSEUMFINLAND—Finnish Museums on the Semantic Web”.² This system contains an inter-museum exhibition of over 4000 cultural artifacts, such as textiles, pieces of furniture, tools, etc. Also metadata concerning some 260 historical sites in Finland were incorporated in the system. The goals for developing the system were the following:

- Global view to distributed collections: it is possible to use the heterogeneous distributed collections of the museums participating in the system as if the collections were in a single uniform repository.
- Content-based information retrieval: the system supports intelligent information retrieval based on ontological concepts, not on simple keyword string matching as is customary with current search engines.
- Semantically linked contents: a most interesting aspect of the collection items to the end-user are the implicit semantic relations that relate collection data with their context and to each other. In MUSEUMFINLAND, such associations are exposed dynamically to the end-user by defining them in terms of logical predicate rules that make use of the underlying ontologies and collection metadata.

- Easy local content publication: the portal should provide the museums with a cost-effective publication channel.

Museum databases are usually situated at different locations and use different database systems and schemas. This creates a severe obstacle to information retrieval. To address the problem, the web can be used for creating a single interface and access point through which a search query can be sent to distributed local databases and the results combined into a global hit list. This “multi-search” approach is widely applied and there are many cultural collection systems on the web based on it, such as the portals Australian Museums Online³ and Artefacts Canada.⁴

A problem of multi-search is that by processing the query independently at each *local database*, the *global dependencies*, associations between objects in different collections are difficult to find. Since exposing semantic associations between collections items is one of our main goals, MUSEUMFINLAND cannot be based on the multi-search paradigm. Instead, the local collections are first consolidated into a global repository, and the search queries are answered based on it. Mutually shared conceptual models, ontologies, are used for enriching the content and for making the collections interoperable. To show the associations to the end-user, the collection items are represented as web pages interlinked with each other through the semantic associations. The MUSEUMFINLAND home page is the single entry point through which the end-user enters the global semantic WWW space. A challenge in this approach is that a separate content creation process is needed for consolidating the global repository based on local databases.

This paper presents MUSEUMFINLAND from different viewpoints [15,13,19,18,25]. The ontologies underlying the system are first discussed. After this we explain how content from the museum databases can be imported into the global RDF(S)⁵ [21,1] repository conforming to the shared ontologies. Next the semantic search and browsing services of MUSEUMFINLAND are explained from the end-user’s viewpoint, and adaptation of the system to new data is discussed. Then we get

¹ <http://www.w3.org/2001/SW/>.

² <http://museosuomi.fi/>.

³ <http://www.amonline.net.au/>.

⁴ <http://www.chin.gc.ca/>.

⁵ <http://www.w3.org/RDF/>.

down to the implementation and describe the general architecture underlying the system and its components. The paper concludes by discussing the lessons learned as well as related and future work.

2. Ontologies

MUSEUMFINLAND is based on seven domain ontologies:

- (1) The Artifacts ontology (3227 classes) is a subclass taxonomy of tangible collection objects, such as pottery, cloths, weapons, etc. All artifacts in the system belong to some class in this ontology.
- (2) The Materials ontology (364 classes) is a subclass taxonomy of the artifact materials, such as steel, silk, tree, etc.
- (3) The Actors ontology (26 classes, 1715 instances) defines classes of agents, such as persons, companies, etc., and individuals as instances of these classes.
- (4) The Situations ontology (992 classes) is a taxonomy that includes intangible happenings, situations, events, and processes that take place in the society, such as farming, feasts, sports, war, etc.
- (5) The Locations ontology (33 classes, 864 instances) represents areas and places on the Earth. It contains classes such as Continent, Country, County, City, Farm, etc. The main content in the ontology is its individual location instances (e.g., Helsinki or Finland) and their mutual meronymy relations (e.g., Helsinki is a part of Finland).
- (6) The Times ontology (57 classes) is a meronymy of various predefined historical periods. First, there are categories representing special eras of interest such as the Middle Ages and the time of the World War II. Second, there is a linear breakdown hierarchy of centuries and decennia. The properties of time concepts are a human readable label of period and the beginning and end year of the time interval.
- (7) The Collections ontology (22 classes, 24 instances) is a taxonomy that classifies the collections included in the portal under the museums hosting them. The properties of the taxonomy indicate the name and the hosting museum of the collection.

In Finland, the most notable and widely used thesaurus for cultural content in Finnish is MASA [23]

maintained by the National Board of Antiquities.⁶ MASA consists of some 6000 terms and employs the usual thesaurus relations [5], such as narrower term (NT), broader term (BT), and related term (RT). In our work, MASA thesaurus was transformed into an RDF Schema ontology called MAO by first creating an RDF Schema structure from the MASA database. This initial ontology was then enhanced and edited as a Protégé-2000⁷ project by hand. In this way, three domain ontologies, Artifacts, Materials, and Situations emerged as sub-ontologies of MAO. These ontologies were later on extended based on collection item data from the collections of the National Museum,⁸ Espoo City Museum,⁹ and Lahti City Museum.¹⁰

The Locations ontology was created by first defining classes like Continent, Country, City, Farm, etc. An initial set of a couple hundred individual countries and cities was generated automatically from official data sources, and the ontology was populated further based on actual collection data. In the same vein, the small class structure of the Actors ontology (classes Person, Woman, Company, etc.) was first created by hand and populated later by instance data. Details of the ontology creation process of MUSEUMFINLAND can be found in [19,18].

A major goal of MUSEUMFINLAND is to provide the end-user with semantic association links relating collection contents with each other. Such associations are based on cultural and common sense knowledge about the society and its functions. They tell, for example, how, in what context, and for what purpose different artifacts have been used. Much of this kind of knowledge falls outside of traditional taxonomic ontological knowledge and is not explicit in the metadata descriptions either. We therefore decided to enrich the knowledge base of MUSEUMFINLAND with addition cultural and common sense knowledge. Such knowledge serves two purposes:

- From the end-user's viewpoint, it enables semantic link generation and semantic browsing. This

⁶ <http://www.nba.fi/>.

⁷ <http://protege.stanford.edu/>.

⁸ <http://www.nba.fi/en/nmf/>.

⁹ <http://www.espoo.fi/museo/>.

¹⁰ <http://www.lahti.fi/museot/>.

feature will be discussed in detail in the coming sections.

- From the cataloger’s viewpoint, it makes the cataloging process simpler because many additional annotations can be automatically created. For example, if we know that the artifact is a doctor’s hat then there is no need to tell that it is related to academic ceremonies, because this inference can be drawn by a simple rule.

Additional knowledge was incorporated into the system in two ways: (1) by explicit associations and (2) by more complex logic rules using them (in addition to ontological knowledge and metadata).

A few simple explicit association types of form *X isRelatedTo Y* were identified. Firstly, we envisioned that the events taking place in the society, i.e., classes in the Situations ontology, are of central importance for creating useful semantic linkage. Therefore additional association triples of form (*artifact, is-related-to-event, situation*) were created. These relations were defined by a museum curator with the user-friendly N3-notation.¹¹ For example:

```
masa:spade mapping:is-related-to-event
  masa:forestry.
masa:Christmas_tree mapping:is-related-to-
  event masa:Christmas.
```

Secondly, artifacts are related to each other, which can be represented by the triple (*artifact1, is-related-to-artifact, artifact2*). For example, sailing ships are related to sails, screw drivers to screws, etc. Thirdly, there are association between artifacts and materials. Altogether, 301 different associations between ontology classes were created in this way.

Based on the ontologies, associations, annotation schema, and the metadata from the databases, a set of more complex labeled associations between resources were defined in terms of predicate logic rules. These rules (to be discussed in more detail later) exploit, e.g., the fact that the associations are inherited along the `rdfs:subClassOf` hierarchy, make use of the relations defined in MASA, and use the various metadata annotation properties of the collection artifacts.

3. Content creation process

The collection item (meta)data of MUSEUMFINLAND came from four databases. The databases were situated in different locations (Espoo, Helsinki, and Lahti) and used four different database schemas and cataloging systems (Escoll, Antikvaria, Musketi, and MS Access) that were based on three different database systems (Ingress, MS Server, MS Access). A part of the MUSEUMFINLAND project was to create a content creation process for transforming local heterogeneous databases into a global, syntactically and semantically interoperable knowledge base in RDF format, which conforms to the set of seven global museum ontologies. The process was designed to meet two requirements: first, new museum collections need to be imported into the MUSEUMFINLAND portal as easily as possible and with as little manual work and technical expertise as possible. Second, the museums should have maximal local freedom in annotations and need to commit to only necessary restrictions and complications imposed by the portal and the other content providers. For example, two museums may use different terms for the same thing. The system should be able to accept the different terms as far as the terms are consistently used and their local meanings with respect to the global reference ontologies are provided.

Fig. 1 depicts the annotation process [19,18] that consists of three major parts. First syntactic homogenization is obtained by transforming the relational database records into a shared XML language, cf. the DB2XML arrow on the left. The result is a set of *XML cards*. Second, terminology definitions in RDF, called *term cards*, are created based on the XML data, cf. the lower XML2RDF arrow. The transformation is performed with the help of a tool called Terminator. The term cards map XML level literals onto URIs in the museum ontologies. Third, semantic interoperability is obtained by transforming the XML cards—with the help of term cards—into RDF form that conforms to the global museum ontologies, cf. the upper XML2RDF arrow on the right. The result is a set of *RDF cards*. This transformation is performed by a tool called Annomobile.

The first step in combining the heterogeneous databases is to gain syntactic interoperability by transforming database contents into a shared XML for-

¹¹ <http://www.w3.org/DesignIssues/Notation3.html>.

mat. Based on the schema, each collection item has an XML description of its own, the XML card. For example, the XML card representing a calendar is presented below¹²:

```
<artifactCard created='`2003-7-29
10:43:16''>
  <artifactId> ECM:22461:1 </artifactId>
  <artifactType> Christmas calendar,
    Finland's Scouters Assoc.
</artifactType>
  <museum> Espoo City Museum </museum>
  <material> cardboard </material>
  <keywords>
    <keyword> Christmas </keyword>
    <keyword> calendar </keyword>
    <keyword> scouts </keyword>
  </keywords>
  <placeOfUsage> Tapiola, Espoo
</placeOfUsage>
  <creator> Ulla Vaajakoski </creator>
  ...
  <photo> photos/image3451.jpg </photo>
</artifactCard>
```

```
<rdf:RDF
  xmlns:rdf='`http://www.w3.org/1999/02/22-rdf-syntax-ns#'
  xmlns:card='`http://www.fms.fi/RDFCard#'>
  <card:RDFCard
    rdf:about='`http://www.fms.fi/rdfCard#card11023''>
    card:artifactId='`16851''
    card:artifactType-www='`calendar''
    card:artifactType='`http://www.fms.fi/artifacts#calendar''
    card:museum-www='`Espoo City Museum''
    card:museum='`http://www.fms.fi/agents#EspooCityMuseum''
    card:material-www='`cardboard''
    card:material='`http://www.fms.fi/materials#cardboard''
    ...
  </card:RDFCard>
  ...
</rdf:RDF>
```

An XML card presents the main features of a collection object by sub-elements. The values of the features, such as the string “Espoo City Museum” in the sub-element

<museum>, are read from the underlying database tables.

Semantic interoperability is obtained by transforming XML cards into RDF cards. The process is based on a terminology represented as a set of term cards. A term card essentially associates a literal term with an URI in an ontology. Based on such mappings, ontological literal data values on the XML level can be replaced with URI references to ontological concepts and individuals on the RDF level. Initial sets of term cards were first created automatically based on the MASA thesaurus and the ontologies of MUSEUMFINLAND, and were later populated by additional new terms used in the collection databases. Each museum can in principle use a terminology of its own as long a term card mapping to ontological URIs is provided.

For example, the XML card presented earlier would translate into the RDF card below:

The elements of the XML cards fall in two categories: *literal features* and *ontological features*. Literal features are to be represented only as literal values on the RDF level, too. They are, for example, used in the user interface. Ontological feature values need to be linked to not only literal values but to ontological resources (URIs), too. For example, in the above RDF card the feature `artifactId` is literal and is not connected with the ontology resources. In contrast, the ontological feature `material` is represented with the literal property `material-www` and the ontolog-

¹² The example is translated and slightly simplified from the original version in Finnish.

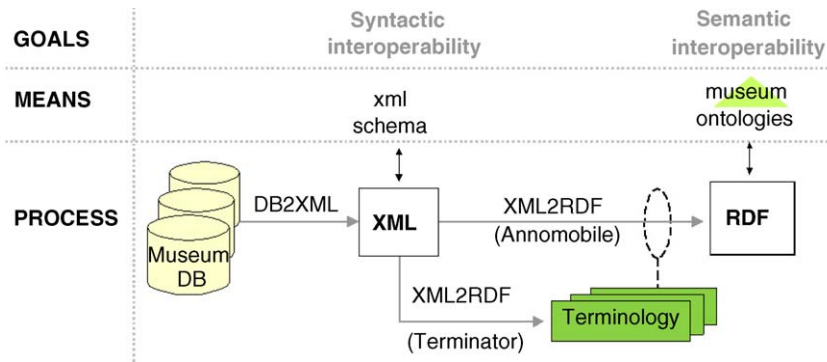


Fig. 1. The content creation process in MUSEUMFINLAND.

ical property material that has an RDF resource (URI) as its value. This URI connects the card resource with the material ontology and through it with other resources.

When mapping ontological feature values to URIs in domain ontologies, two major problem situations occur related to (1) unknown values and (2) homonyms. In case of unknown values, there are no applicable term card candidates in the terminology. The solution to this is to map the feature value either to a more general concept or to a resource considered unknown. For example, if one knows that an artifact was created in some city in Lapland, one can create an unknown instance of the class City, tell that it is a part of Lapland, and annotate the place-of-manufacture feature with this instance.

The problem of homonymous terms occurs when there are homonyms within the range of ontologies used for annotating the ontological feature at hand. For example, the Finnish literal term “kilvet” as a value of the artifact type feature, can mean either a signboard or a coat of arms, and these interpretations cannot be disambiguated by using the range restrictions of property values. The solution employed in Annomobile is to fill the RDF card with all potential choices, inform the human editor about the problem, and ask her to remove the false interpretations on the RDF card manually. Our first experiments indicate, that at least in Finnish not much manual disambiguation work is needed, since homonymy typically occurs between terms referring to different domain ontologies. However, the problem still remains in some cases and is likely to be more severe in languages like English having more homonyms.

4. End-user’s perspective

MUSEUMFINLAND provides the end-user with two services:

- A *semantic view-based search engine* that is based on the underlying knowledge base consisting of ontologies and instance data.
- A *semantic linking system* by which the user can find out semantic associations within the portal content, and use the associations for browsing.

The search engine of MUSEUMFINLAND is based on the multi-facet search paradigm [28,9]. Here the concepts used for indexing are called *categories* and are organized systematically into a set of hierarchical, orthogonal taxonomies. The taxonomies are called subject *facets* or *views*. In multi-facet search the views are exposed to the end-user in order to provide her with the right query vocabulary and for presenting the repository contents and search results along different views.

Each category is related to a set of search objects that we will call its *projection*. The *extension* E of a category is the union of its projection P and the extensions of its sub-categories S_i : $E = P \cup S_1 \cup S_2 \cup \dots \cup S_n$. A search query in multi-facet view-based search is formulated by selecting categories of interest from the different facets, typically one selection from a facet. The answer to the query is simply the intersection of the extensions E_i of the selected categories: $A = \cap \{E_i\}$. For example, by selecting the category “Chairs” from the Artifact facet, and “Helsinki” from the place of manufacturing facet, the user can express the query for retrieving all

chairs (of any subtype) manufactured in Helsinki (or in any of its suburbs and other locations within Helsinki).

MUSEUMFINLAND classifies the collection items along nine views organized in four groups. The Artifact Views describe the physical aspects of the collection item (artifact type and materials). The Creation Views tell who manufactured or created the artifact, as well as the location and time of the creation. The Usage Views indicate the user of the artifact, place of usage, and situations in which the artifact is used. Finally, the Collection View classifies the museums and collections participating in the portal.

Facets can be used for helping the user in information retrieval in many ways. First, the facet hierarchies give the user an overview of what kind of information there is in the repository. Second, the hierarchies can guide the user in formulating the query in terms of appropriate concepts. Third, the hierarchies do not suffer from the problems of homonymous query terms. Fourth, the facets can be used as a navigational aid when browsing the database content [9]. Fifth, the number of hits in every category that can be selected next can be computed *beforehand* and be shown to the user [28].

Fig. 2 shows the search interface of MUSEUMFINLAND. The nine facets are shown on the left (in Finnish), such as artifact type (“Esinetyyppi”) and material (“Materiaali”). For each facet, the next level of sub-categories is shown as a set of links. A query is formulated by selecting a category by clicking on its name. When the user selects a category *c* in a facet *f*, the system constrains the search by leaving in the result set only such objects that are annotated in facet *f* with some sub-category of *c* or *c* itself. Here the user has selected the sub-category tools (“työvälineet”) from the artifact type facet (“Esinetyyppi”), and the result set is seen on the right grouped by the sub-categories of tools, such as textile making tools (“tekstiilityövälineet”) and tools of folk medicine (“kansanlääkinnän työvälineet”). Hits in different categories are separated by horizontal bars and can be viewed page by page independently in each category. The number of hits shown in each sub-category is determined from the number of sub-categories in the result set in order to maximize useful information on the limited screen space. In this case, all sub-categories do not fit on the screen, and only a single line of hits is shown for each sub-category.

The user can refine the query further by selecting another category on the left. For example, assume that

the user selects category Farming and cattle tending (“maatalous ja karjanhoito”) in the view Situation of usage (“Käyttötilanne”) in Fig. 2. Three things happen when answering the query. First, the result set on the right is refined to the intersection of previous selections; here the result is tools used in farming and cattle tending. Second, the selected view is changed to expose the sub-categories of the selected category. Third, the size *n* of the result set resulting from the selection of any category link seen on the screen is recomputed proactively, and a number (*n*) is shown to the user after each category name. This number tells that if the category is selected next, then there will be *n* hits in the result set. For example, in Fig. 2, the number 193 in the Collection facet (“Kokoelma”) on the bottom tells that there are 193 tools in the collections of the National Museum (“Kansallismuseon kokoelmat”). A selection leading to an empty result set (*n* = 0) is removed from the facet (or alternatively disabled and shown grayed out, depending on the user’s preference). In this way, the user is hindered from making a selection leading to an empty result set, and is guided toward selections that are likely to constrain the search appropriately. The query can be relaxed by making a new selection on a higher level in the facet or by dismissing the facet totally from the query.

Above, the category selection was made among the direct sub-categories of the facets. An alternative way is to click on the link Whole facet (“koko luokittelu”) on any facet. The system then shows all possible selections in the whole facet hierarchy with hit counts. This gives the user a good overview of the distribution of items over a desired dimension. By grayed out categories with no hits, it is also easy to see in what categories the collections are lacking artifacts. This may be a useful piece of information for, e.g., the collection manager.

View-based search is not a panacea for information retrieval. Google-like keyword search interface is usually preferred [4] if the user is capable of expressing her information need in terms of accurate keywords. MUSEUMFINLAND seamlessly integrates this functionality with view-based search in the following way: first, the search keywords are matched against category names in the facets in addition to text fields in the metadata. A new dynamic facet is created in the user interface. This facet contains all facet categories whose name (or other property values) matches the keyword. Intuitively these facet categories tell the different inter-

Address: <http://museosuomi.cs.helsinki.fi/?l=fil&m=0&n=%2500%2516&g=c%2500%2516> Go

MuseoSuomi
- Suomen museot semanttisessa webissä -

HELSINKI INSTITUTE FOR INFORMATION TECHNOLOGY UNIVERSITY OF HELSINKI

Uusi haku | Ohjeet | Näytä kaikki kategoriat | Tietoa ohjelmasta | MuseoSuomi-palautte

Sanahaku: Hee tarkenna hakua

Esinetyyppi [kaikki](#) > [työvälineet](#) (koko luokittelu)

[tekstiilikasityövälineet](#) (219),
[kansanlaakinnan työvälineet](#) (1),
[luokittelemattomat työvälineet](#) (36),
[maataloustyövälineet](#) (7), [metallityövälineet](#) (1),
[pilkkomis ja hienontamisvälineet](#) (4),
[kirjoitusvälineet](#) (9), [metsätyövälineet](#) (4),
[työkälut](#) (22)

Materiaali (koko luokittelu) (ryhmittele kohteet)

[materiaalit](#) (241)

Valmistaja (koko luokittelu) (ryhmittele kohteet)

[henkilöt](#) (9), [tuotemerkit](#) (2),
[yritykset](#) (38)

Valmistuspaikka (koko luokittelu) (ryhmittele kohteet)

[Afrikka](#) (2), [Etelä-Amerikka](#) (1),
[Eurooppa](#) (84)

Valmistusaika (koko luokittelu) (ryhmittele kohteet)

[aikakaudet](#) (90), [vuosisadat](#) (89)

Käyttäjä (koko luokittelu) (ryhmittele kohteet)

[henkilöt](#) (54), [laitokset](#) (1),
[yritykset](#) (3)

Käyttöpaikka (koko luokittelu) (ryhmittele kohteet)

[Eurooppa](#) (71)

Käyttötilanne (koko luokittelu) (ryhmittele kohteet)

[harrastus- ja kansalaistoiminta](#) (4),
[kohteelle tehtävät toimenpiteet](#) (17),
[maatalous ja karjanhoito](#) (2),
[ruoan- ja puomanvalmistus](#) (3),
[toimijoiden yleiset prosessit](#) (2),
[elinkeinot](#) (9), [valmistustekniikat](#) (179)

Kokoelma (koko luokittelu) (ryhmittele kohteet)

[Espoon kaupunginmuseon kokoelmat](#) (54),
[Kansallismuseon kokoelmat](#) (193),
[Lahden kaupunginmuseon kokoelmat](#) (50)

Hakuehdot

Kategoria: Esinetyyppi > [työvälineet](#) (ryhmittele kohteet) (poista)

Kohteet ryhmiteltyinä kategorian [työvälineet](#) mukaisesti
(näytä ilman ryhmittelyä)

[tekstiilikasityövälineet](#), kohteet 1-4/219 (ryhmittele kohteet)

kehräpuu, kuosali (NBA SU4527 50)	kehruulauta, kehräpuu, kuezzel, kuosali (NBA SU5069 26)	rukinlapa (ECM 100 1)	sneldde, värttänlumppio, värttänpyörä (NBA SU2449 7)
(edellinen) / (seuraava)			

[kansanlaakinnan työvälineet](#), kohteet 1-1/1 (ryhmittele kohteet)

suonirauta suonensisäntärauta (ECM 2711 1)			
(edellinen) / (seuraava)			

[luokittelemattomat työvälineet](#), kohteet 1-4/36 (ryhmittele kohteet)

nappikoukku, napituskoukku (ECM 3594 264)	kietkamläbdzi, komsiöhina (NBA SU4922 32)	palohosat, palohosat (ECM 614 1)	luontilasta (NBA SU4135 166)

Fig. 2. MUSEUMFINLAND search interface after selecting the category link tools (“työvälineet”).

pretations of the keyword, and by selecting one of them next the right choice can be made. This approach also solves the search problem of finding relevant categories in facets that contain thousands of categories. Second, a result set of object hits is shown. This result set contains all objects contained in any of the categories matched in addition to all objects whose metadata directly contains

the keyword. The hits are grouped by the categories found.

At any point during multi-facet search the user can select any hit found by clicking on its image. The corresponding data object is then shown as a web page, such as the one in Fig. 3. The example depicts a special part, distaff (“rukinlapa” in Finnish) used in a spinning

The screenshot shows the MuseoSuomi website interface. At the top, there are logos for Helsinki Institute for Information Technology and the University of Helsinki. The main header reads "MuseoSuomi - Suomen museot semanttisessa webissä". Below the header, there are navigation links for "Uusi haku", "Tarkain hakutuloksille", "Ohjeet", "Tietoa ohjelmasta", and "MuseoSuomi-palautte".

The main content area is titled "rukinlapa" and features a photograph of a wooden distaff. To the right of the image, the following metadata is displayed:

- Valmistuspaikka: Suomi
- Valmistusaika: 1793
- Käyttöpaikka: Suomi, Bemböle, Espoo, Suomi, Vanhakartano, Espoo, Suomi
- Asiasana: KEHRUU, KORISTEVEISTO, PUUMERKKI, VUOSILUKU
- Museokokoelma: Museokokoelma
- Vastuumuseo: Espoon kaupunginmuseo
- Asiasanasto: Espoon kaupunginmuseon sanasto
- Esineen numero: ECM:100.1
- ID: 1001

Below the metadata, there are several facet categories with hierarchical link paths:

- Esinetyyppi:**
 - työvälineet (290) > tekstihäkätyövälineet (219) > kehrun ja langanvalmistuksen työvälineet (63) > kehruvälineet (59) > kuontalopihmet (3) > **rukinlapat** (1)
- Valmistuspaikka:**
 - Eurooppa (2541) > **Suomi** (2239)
- Valmistusaika:**
 - aikakaudet (3024) > historiallinen aika (3023) > **uusi aika** (3013)
 - vuosikadat (3012) > **1700-luku** (123)
- Käyttöpaikka:**
 - Eurooppa (2232) > **Suomi** (2227)
 - Eurooppa (2232) > Suomi (2227) > Etelä-Suomen läänit (1999) > Uusimaa-Nyland (670) > Espoo (512)
 - Eurooppa (2232) > Suomi (2227) > Etelä-Suomen läänit (1999) > Uusimaa-Nyland (670) > Espoo (512) > **Bemböle** (14)
- Käyttötilanne:**
 - valmistustekniikat (1587) > tekniikan työ (39) > **veisto** (32) > **koristeveisto** (8)
 - valmistustekniikat (1587) > tekstiilityo (586) > kumutyö (74) > **kehruu** (64)
- Kokoelma:**
 - Espoon kaupunginmuseon kokoelmat (1190) > **Museokokoelma** (1129)

On the right side of the page, there are sections for semantic recommendations:

- Sama käyttöpaikka:**
 - Bemböle:
 - lämsävuolin
 - opetusväline.peli
 - opetusväline.peli
 - opetusväline.peli
 - opetusväline.peli
 - Espoo:
 - kurakirja kuvakirja kangasta
 - lennokilapen biyhöhänen lenkki
 - neuletakki naisen neuletakki
 - hartiavaate naisen pitsinen hartiavaate
 - puvin väkosa, jalkunaisen puvin väkosa
 - Suomi:
 - ruokailina ruokailina, damasti
 - Kartalina Kartalina, etupistokirjontaa
 - pöytälinna pöytälinna, kirjottu
 - pöytälinna risti-ristokirjontainen pöytälinna
 - Kartalina bahtilina, kirjottu
- Esineeseen liittyvään paikkaan liittyviä muinaismuistoja:**
 - Espoo:
 - Röykkiöt
 - Puolustusvarustukset
 - Röykkiöt
 - Röykkiöt
- Samaan aiheeseen liittyviä esineitä:**
 - ajan_kasitteet:
 - hevosloimi
 - arjku vaatearjku
 - takki vanupeite
 - veisto: pienosveistokset
 - pesukarttu kunkka
 - kehruu:
 - jalkkara kehruuajakkara
 - rullatuokirullateline
 - puola lankapuola
 - puola lankapuola
 - loukku pellavaloukku

Fig. 3. Web page depicting a collection object, its metadata, facet categories, and semantic recommendation links to other collection object pages.

wheel. The page contains the following information and links:

- (1) On top, there are links to directly navigate in the groups and results of the current query.
- (2) The image(s) of the object is (are) depicted on the left.
- (3) The metadata of the object is shown in the middle on top.
- (4) All facet categories that the object is annotated with are listed in the middle bottom as hierarchical link paths. A new search can be started by selecting any category there.
- (5) A set of semantic links to related artifacts is shown on the right.

The semantic links on the right reveal to the end-user a most interesting aspect of the collection items:

the implicit semantic relations that relate collection data with their context and each other. The links provide a *semantic browsing* facility to the end-user. For example, in Fig. 3 there are links to objects used at the same location (categorized according to the name of the common location), to objects related to similar events (e.g., objects used in spinning, and objects related to concepts of time, because the distaff in question has a date carved onto it), to objects manufactured at the same time, and so on. Since a decoratively carved distaff used to be a typical wedding gift in Finland, it is also possible to recommend links to other objects related to the wedding events, such as wedding rings. These associations are exposed to the end-user as link groups whose titles and link names explain to the user the reason for the link recommendation.

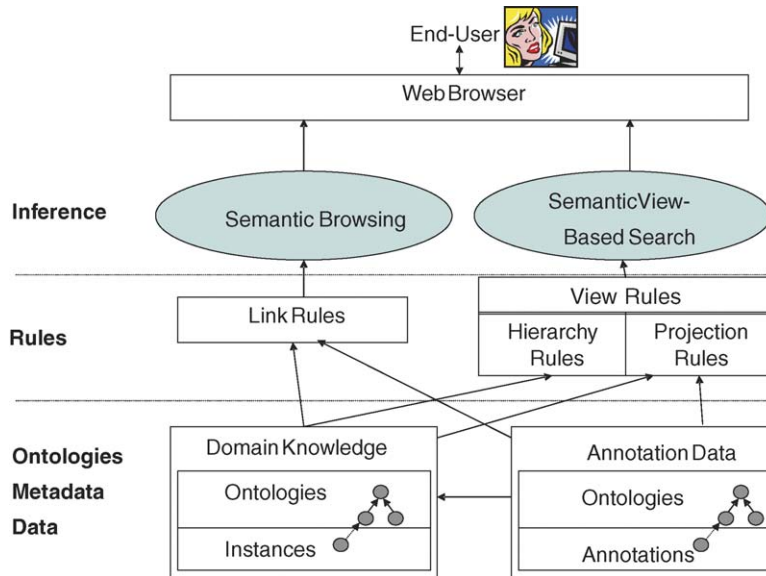


Fig. 4. Architecture of MUSEUMFINLAND on the server side.

5. Adapting services for new content

Fig. 4 depicts the relation between contents and services in MUSEUMFINLAND on the server side. The system is used by a web browser that provides the semantic view-based search and semantic browsing services to the end-user. The services are based on two forms of content: (1) domain knowledge consists of the ontologies that define the domain concepts and the individuals; (2) annotation data describes the metadata of the data resources represented as RDF cards.

A technical innovation of MUSEUMFINLAND is to introduce an intermediate mapping layer of logical rules between the content and semantic services: Link Rules for the browsing service and view rules for the search engine. By using the rules the generic service

In the following, the idea of View Rules and Link Rules is described in more detail by using examples. We use SWI-Prolog¹³ as the inference engine and SWI-Prolog syntax in the examples.¹⁴

A view is a hierarchical index-like decomposition of category resources where each category is associated with a set of sub-categories and a set of directly related data items. A view is defined in terms of ontologies by specifying a view rule predicate called *ontodella_view*. It contains the following information: (1) the root resource URI; (2) a *hierarchy rule* defined by a binary sub-category relation predicate; (3) a binary *projection rule* predicate that maps search objects onto the view categories; (4) a label for the view. An example of a view rule predicate is given below:

```
ontodella_view(
  'http://www.cs.helsinki.fi/seco/ns/2004/03/places#earth',
  place_sub_category, place_of_use_leaf_item,
  [fi:'Käyttöpaikka', en:'Place of Usage'] % the labels
).
```

engines can be separated from domain- and annotation-specific details and be adapted to contents of different kind by changing the rules only. The rules are defined declaratively in terms of Prolog predicates operating on RDF triples as in [12].

¹³ <http://www.swi-prolog.org>.

¹⁴ The syntax used in the examples is translated from Finnish and is slightly simplified for better readability.

Here the URI on the second line is the root resource, `place_sub_category` is the name of the hierarchy sub-category predicate and `place_of_use_leaf_item` is the projection rule predicate. The label list contains the labels for each supported language, here in Finnish (fi) and in English (en).

The root URI defines the resource in a domain ontology that will become the root of the view hierarchy tree, while the hierarchy rule specifies how to construct the facet hierarchies from the domain ontologies. Hierarchy rules are needed in order to make the classifications shown to the user independent from the design choices of the underlying domain ontologies. The view-based search engine itself does not know about the ontologies, it deals with tree-like category hierarchies.

We have used two hierarchy rules to extract a facet from the RDF(S)-based domain knowledge. Firstly, the `rdfs:subClassOf` relation can be used in facets such as `Artifact type`, and the projection rules map RDF cards of corresponding artifacts to these categories. Second, places constitute a part of meronymy. Creating views along this dimension is a natural choice for the location facets in the user interface. For example, in the above view rule, the binary sub-category predicate `place_sub_category` can be defined by the containment property `isContainedBy` in the following way:

```
place_sub_category(ParentCategory, SubCategory):-
  SubCategoryProperty = 'http://www.cs.helsinki.fi/seco/ns/2004/03/places#isContainedBy',
  rdf(SubCategory, SubCategoryProperty, ParentCategory).
```

A projection rule tells when an RDF card instance is a member of a category. For example, the rule `place_of_use_leaf_item` in our example above could be defined as follows:

```
place_of_use_leaf_item(ResourceURI, CategoryURI):-
  Relation = 'http://www.cs.helsinki.fi/seco/ns/2004/03/artifacts#usedIn',
  rdf(ResourceURI, Relation, CategoryURI).
```

Based on hierarchy and projection rules, the view categories can be generated by iterating through the predicate `ontodella_view`, and by recursively creating the category hierarchies using the sub-category rules starting from the given root category. At every category, all relevant resources are attached to the category based on the projection rules.

Hierarchy rules tell how the views are projected logically. A separate question is how these hierarchies should be shown to the user. Firstly, the ordering of the sub-resources may be relevant. For example, the sub-happenings of an event should be presented in the order in which they take place and persons be listed in alphabetical order. The ordering of the sub-nodes can be specified by a configurable property; the sub-categories are sorted based on the values of this property. Second, one may need a way to filter unnecessary resources away from the user interface. For example, the ontology is typically created partly before the actual annotation work and may have more classes and details than were actually needed. Then empty categories should be pruned out. A hierarchy may also have intermediate classes that are useful for knowledge representation purposes but are not very natural categories to the user. Such categories should be present internally in the search hierarchies but should not be shown to the user. Third, the names for categories need to be specified. For example, the label for a person category should be constructed from the last and first names represented by distinct property values.

Link Rules (cf. Fig. 4) are used for creating semantic recommendation links, such as those in Fig. 3. Such links can be created in various ways [29]. In our work, we employed the idea of *rule-based recommendations*

where the domain specialist explicitly describes the notion of “interesting related resource” with generic logic rules. The system then applies the rules to the underlying knowledge base in order

to find interesting resources related to the selected one.

This method has several strengths. Firstly, the rule can be associated with a label, such as “Other artifacts used in event *x*”, that can be used as the explanation for the recommendations found. It is possible to deduce the explanation label as a side effect of apply-

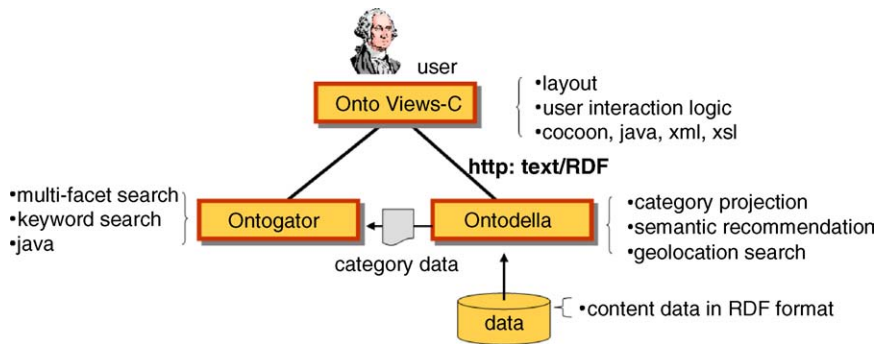


Fig. 5. The components of ONTOVIEWS.

ing the rule. Secondly, since semantic linking rules are described by the domain specialist, the rules and explanations are explicitly defined and are not based on heuristic measures, which could be difficult to understand and motivate. The specialist knows the domain and may promote the most important relations between the resources. However, this could also be a weakness if the user's goals and the specialists thoughts about what is important do not match, and the user is not interested in the recommendations. Thirdly, the rule-based recommendations do not exclude the possibility of using other recommendation methods but provides an infrastructure for applying any rules. For example, the recommendation rules could perhaps be learned or tuned by observing the users actions.

The rule-based semantic linking system of MUSEUMFINLAND is described in more detail in the next section.

6. Architecture and implementation

MUSEUMFINLAND has been implemented by using a tool called ONTOVIEWS¹⁵ [25]. This tool was developed during the project and has later been applied to creation of other semantic portals as well [22,25]. ONTOVIEWS consists of the three major components shown in Fig. 5:

(1) The logic server ONTODELLA provides the system with reasoning services, such as category view pro-

jection and dynamic semantic link recommendations.

- (2) The search engine ONTOGATOR is a generic view-based RDF search engine, responsible for the multi-facet search functionality of the system.
- (3) The third component ONTOVIEWS-C binds the services of ONTOGATOR and ONTODELLA together, and provides the user interfaces.

ONTODELLA is a logic engine for defining and executing the view and link rules of Fig. 4. ONTODELLA is a multi-threaded web server which provides remote access to execute the rules in the framework. The web server and the rule execution framework are written using SWI Prolog¹⁶ and its readily available HTTP libraries. For the mobile user interface, ONTODELLA has been extended to provide simple point-of-interest search based on geo-coordinates available from the mobile phone.

ONTODELLA provides services for (1) view creation, (2) semantic link generation, and (3) geolocation search. View creation is done by a separate process before starting MUSEUMFINLAND due to the long time required to execute the hierarchy and projection rules, and due to the size of the view trees. Linking services and geolocation search are run dynamically on request. In below, these services are explained in more detail.

View creation service provides necessary hooks for executing the hierarchy and projection predicates. The view creation algorithm traverses the ontologies by

¹⁵ The software is available at <http://www.cs.helsinki.fi/group/seco/museums/dist/> in open source.

¹⁶ <http://www.swi-prolog.org>.

using the given predicates dynamically in a depth-first search. The resulting view structure is serialized in RDF/XML according to a model derived from the Annotea Bookmark Schema¹⁷. This structure is used by ONTOGATOR as the basis for the view-based search.

The dynamic *semantic link service* of ONTODELLA is based on linking rules. In response to a semantic linking service request with a given URI, the framework calls for all defined semantic link rules. Each link rule can be arbitrarily complex and is defined by a domain

specialist. A linking rule is described by a predicate of the form:

predicate(SubjectURI, TargetURI, Explanation)

that succeeds when the two resources *SubjectURI* and *TargetURI* are to be linked. The variable *Explanation* is then bound to an explanatory label (string) for the link.

In below, one of the more complex rules—linking items related to a common event—is presented as an example:

```
related_by_event(Subject, Target, Explanation):-
ItemModelProperty =
    'http://www.cs.helsinki.fi/seco/ns/2004/03/artifacts#item.type',
ItemToEventRelatingProperty =
    'http://www.cs.helsinki.fi/seco/ns/2004/03/mapping#related_to_event',
% check that both URIs correspond in fact to artifacts
isArtifact(Subject), isArtifact(Target),
% and are not the same
Subject \= Target,

% find all the item types the subject item belongs to
rdf(Subject, ItemModelProperty, SubjectItemType),
rdfs_transitive_subClassOf(SubjectItemType, SubClassOfSubjectItemType),

% find all the events any of those item types are related to
rdf(SubClassOfSubjectItemType, ItemToEventRelatingProperty, Event),
% and events they include or are part of
(
    rdfs_transitive_subClassOf(Event, SubOrSuperClassOfEvent),
    DescResource=TransitiveSubOrSuperClassOfEvent;
    % or
    rdfs_transitive_subClassOf(SubOrSuperClassOfEvent, Event),
    DescResource=Event;
),

% find all item types related to those events
rdf(TargetItemType, ItemToEventRelatingProperty, SubOrSuperClassOfEvent),
% and all their superclasses
rdfs_transitive_subClassOf(SuperClassOfTargetItemType, TargetItemType),

% don't make uninteresting links between items of the same type
SuperClassOfTargetItemType \= SubjectItemType,
not(rdfs_transitive_subClassOf(SuperClassOfTargetItemType,
SubjectItemType)), not(rdfs_transitive_subClassOf(SubjectItemType,
SuperClassOfTargetItemType)),
```

¹⁷ <http://www.w3.org/2003/07/Annotea/BookmarkSchema-20030707>.

```
% finally, find all items related to the linked item types
rdf (Target, ItemTypeProperty, SuperClassOfTargetItemType),

list_labels([DescResource], RelLabel),
Explanation=[commonResources(DescResource), label(fi:RelLabel)].
```

The rule goes over several ontologies, first discovering the object types of the objects then traversing the object type ontology, relating the object types to events, and finally traversing the event ontology looking for common resources. Additional checks are made to ensure that the found target is an artifact and that the subject and target are not the same resources. Finally, information about the relation is collected, such as the URI and the label of the common resource, and the result is returned as the link label.

Each rule returns as a result a (possibly empty) set of associated URIs with explanatory labels. The results are grouped according to the rule which generated them and according to the resource that caused the linking. For example, in a rule providing links to collection items manufactured at the same place, the URI of the shared place can be returned as the link causing resource.

ONTODELLA returns the results in XML form that is transformed into HTML by the component ONTOVIEWS-C. In the user interface, the result groups form classified collections of links that can be presented under classification titles subtitled by link causing resources. For example, in the lower right corner of Fig. 3 there is the title objects related to the same theme (“Samaan aiheeseen liittyviä esineitä”) and under it two subtitles corresponding to two link causing resources: concepts of time (“ajan käsitteet”) and spinning (“kehruu”). Under the latter subtitle, the first link “jakkara:kehruujakkara” (spinning chair) points to the web page of a chair used in spinning.

The third ONTODELLA service, geolocation search, gets as input a set of coordinates. In response, the service returns a fixed length ordered list of the location resources nearest to the coordinates, and a corresponding list of bookmarks annotated with the coordinates. This service is used by the mobile telephone interface of MUSEUMFINLAND.

ONTOGATOR defines and implements an RDF-based query interface that is used to separate view-based search logic from the user interface. The interface is

defined as an OWL¹⁸ ontology,¹⁹ and is based on selectors that can be used to query for both view category hierarchies and the projection resources of their categories based on various criteria, such as category, keyword, and geolocation-based constraints. The query is represented in XML/RDF form.

The search result of ONTOGATOR is expressed as an RDF-tree that conforms to a fixed order XML-structure. This allows us to use XML tools such as XSLT to process the results more easily. Since the search results are used in building user interfaces, every resource is tagged with an `rdfs:label`.

Fig. 6 illustrates what happens in an ONTOGATOR search. The query on the left calls for bookmarks, i.e., resources that are searched for, that (1) belong to a sub-category *S* of a view category hierarchy *2* and (2) contain a given keyword. The results on the right are grouped according to an independent additional view hierarchy with the root category *G*. Grouping is based on the next sublevel of *G* as in Fig. 2. Those bookmarks found that do not belong in the grouping hierarchy are returned in the ungrouped category *U*. In the user interface, the results can be shown in groups 1.1, 1.2, and *U*. The RDF query interface allows many options to filter, group, cut, annotate, and otherwise modify the results.

ONTOVIEWS-C is the user interface, interaction, and control component of ONTOVIEWS (cf. Fig. 5). It is built on top of the Apache Cocoon framework.²⁰ Cocoon is a framework based wholly on XML and the concept of pipelines constructed from different types of components. A pipeline always begins with a generator that generates an XML-document. Then follow zero or more transformers that take an XML-document as input and output a document of their own. The pipeline always ends in a serializer that serializes its input into the final result, such as an HTML-page, a PDF-file, or an image. It is also possible for the output of partial

¹⁸ <http://www.w3.org/OWL/>.

¹⁹ <http://www.cs.helsinki.fi/group/seco/ns/2004/03/ontogator#>.

²⁰ <http://cocoon.apache.org/>.

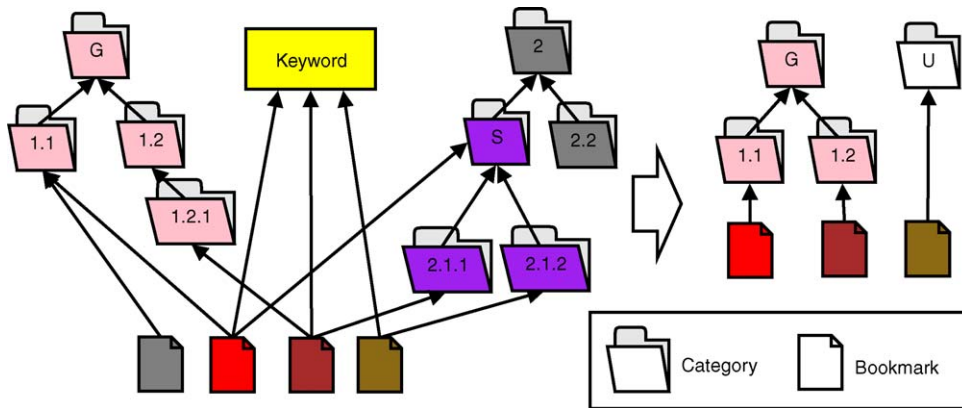


Fig. 6. A keyword plus category selector search with results grouped into an independent, partially cut hierarchy.

pipelines to be combined via aggregation into a single XML-document for further processing. Execution of these pipelines can be tied to different criteria, e.g., to a combination of the request URI and requesting user-agent.

7. Discussion

MUSEUMFINLAND demonstrates the power of semantic web technologies to solving interoperability problems of heterogeneous museum collections when publishing them on the web. The power of the application comes from the use of ontologies and logic:

- Exact definitions: by using ontologies, the museums can define the concepts used in cataloging in a precise, machine understandable way.
- Terminological interoperability: the terms used in different institutions can be made mutually interoperable by mapping them onto common shared ontologies. The ontologies are not used as a norm for telling the museums what terms to use, but rather to make it possible to tolerate terminological variance as far as the terminology mapping from the local term conventions to the global ontology is provided.
- Ontology sharing: ontologies provide means for making exact references to the external world. For example, in MUSEUMFINLAND, the location ontology (villages, cities, countries, etc.) and the actor ontology (persons, companies, etc.) is shared by the museums in order to make the right and interoperable

references. For example, two persons who happen to have the same name should be disambiguated by different URIs, and a person whose name can be written in many ways, should be identified by a single URI to which the alternative terms refer.

- Automatic content enrichment: ontological class and individual definitions, cultural and common sense rules, view projection rules, semantic linking rules, and consolidated metadata enrich collection data semantically.
- Intelligent services: ontologies can be used as a basis for intelligent services to the end-user. In MUSEUMFINLAND, the view-based multi-facet search engine is based on the underlying ontological structures and the semantic link recommendation system reveals to the end-user the underlying semantical context of the collection items and their mutual relations.

A semi-automatic content creation process [19,18] was developed for the museums for transforming their databases into RDF conforming to the shared ontologies. A problem encountered here was that the original museum collection metadata was not systematically annotated, which resulted in manual work when populating the term ontology. The homonymy problem encountered when mapping literal data values to ontology resources was another major problem, but resulted in less manual work than terminology creation. The semi-automatic annotation tools Terminator and Annomobile proved out to be decent programs for the purposes of the project. The annotation process

could be fully automated if the collection cataloging systems were enhanced with datafields for storing URIs in addition to literal descriptions.

A technical innovation of MUSEUMFINLAND is to combine benefits of the multi-facet view-based search paradigm [28,9] with semantic web ontology techniques and reasoning. Logic rules were used for separating the semantic search and link generation services from the underlying domain specific ontologies and (meta)data. In this way, we could separate the generic parts of the system into the tool ONTOVIEWS [25] that has been applied to other application domains as well. The prize of the adaptability is that somebody has to create the view and link rules in Prolog, which can be a difficult task if the input data is not directly suitable for generating the needed projections and links.

When using ONTODELLA, the rules for creating category trees and projections were fairly easy to formulate and verify. The idea of semantic link rules appeared to be a good concept if you know exactly what kind of link rules you want and the data enables the reasoning of those links. We set out to create “intriguing” semantic links for the end-user. However, subjectivity of intriguingness made it difficult (1) to choose what semantic link rules to create, (2) to evaluate the “intrigueness” of the rule, and (3) to order the resulting links based on their relevance.

The use of the Cocoon-based implementation of the ONTOVIEWS appeared to be a good solution compared to our previous test implementations [20,14,17], since it is eminently portable, extendable, modifiable, and modular. This flexibility is a direct result of designing the application around the Cocoon concepts of transformers and pipelines, in contrast to servlets and layout XSLT. We have used ONTOVIEWS in the creation of a semantic yellow page portal [22], and (using a later version of the tool) a test portal based on the material of the Open Directory Project (ODP)²¹. These demonstrations are based on ontologies and content different from MUSEUMFINLAND. With the ODP material, the ONTOGATOR and ONTOVIEWS-C subparts of the system were tested to scale up to 2.3 million data items and 275,000 view categories with search times of less than 5 s on an ordinary PC server.

The use of XSLT in most of the user interface and query transformations makes it easy to modify the inter-

face appearance and to add new functionality. However, it has also led to some quite complicated XSLT templates in the more involved areas of user interaction logic, e.g., when (sub-)paging and navigating in the search result pages. In using XSLT with RDF/XML there is also the problem that the same RDF triple can be represented in XML in different ways but an XSLT template can be only tied to a specific representation. In our current system, this problem can be avoided because the RDF/XML serialization formats used by each of the subcomponents of the system are known, but in a general web service environment, this could cause complications. However, the core search engine components of ONTOVIEWS would be unaffected even in this case because they handle their input with true RDF semantics.

Lots of research has been done in annotating web pages or documents using manual or semi-automatic techniques and natural language processing, c.f. for example CREAM and Ont-O-Mat by [7] and the SHOE Knowledge Annotator [10]. Stojanovic et al. [31] present an approach that resembles ours in trying to create a mapping between a database and an ontology, but they have not tackled the questions of integrating many databases or using global and local terminology to make the mapping inside a domain. Also Handschuh et al. [8] address the problems of mapping databases to ontologies, but their way of doing the mapping is very different from ours, trying to get the data dynamically out of the database and involving the database owner. In [30] a related concepts–terms–data model has been used to define different elements used for creating an ontology out of a thesaurus.

The idea of linking collection items with semantic associations was inspired by Topic Maps [26]. However, in our case the links are not given by a map but are determined by logical inference using the underlying RDF(S) ontology and RDF metadata. Another application of this idea to generating semantically linked static HTML sites from RDF(S) repositories is presented in [12]. Logic and dynamic link creation on the semantic web has been discussed, e.g., in the work on Open Hypermedia [6,3], and in the Promoottori system [17]. In the HyperMuseum [32], collection items are also semantically linked with each other. Here linking is based on shared words in the metadata and their linguistic relations, such as synonymy and antonymy. In contrast, our system is not based on words but on onto-

²¹ <http://www.dmoz.org/>.

logical references in the underlying RDF(S) knowledge base and the links can be defined freely in terms of logical rules. The idea of annotating cultural artifacts with ontologies has been explored, e.g., in [11]. Other ontology-related approaches used for indexing cultural content include Iconclass²² [33] and the Art and Architecture Thesaurus²³ [27].

Much of the web user interface and user interaction logic of MUSEUMFINLAND is based on Flamenco's multi-facet search [9]. In ONTOVIEWS, however, several extensions to this baseline have been added, such as the whole facet view of categories, the seamless integration of concept-based keyword and geolocation search, extended navigation in the result set, and semantic browsing. The easy addition of these capabilities was made possible by basing the system on RDF.

We are investigating how new kinds of cultural RDF material, conforming to different ontologies, can be imported into MUSEUMFINLAND. In the next version of the system called "CultureSampo", more versatile annotation schemas will be used based on events and processes that take place in the society. CultureSampo will contain, e.g., photographs, paintings, folk lore, videos, external web pages, and documents in addition to the artifacts and historical sites present in the current version of MUSEUMFINLAND.

Acknowledgments

Our work was funded mainly by the National Technology Agency Tekes, Nokia Corp., TietoEnator Corp., the Espoo City Museum, the Foundation of the Helsinki University Museum, the National Board of Antiquities, and the Antikvaria Group consisting of some 20 Finnish museums.

References

- [1] D. Brickley, R. V. Guha, Resource description framework (RDF) schema specification 1.0, W3C candidate recommendation 2000-03-27, 2000. <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>.
- [2] Stefan Decker, Michael Erdmann, Dieter Fensel, Rudi Studer, Ontobroker: ontology based access to distributed and semi-structured unformation, in: DS-8, 1999, pp. 351–369. <http://citeseer.nj.nec.com/article/decker98ontobroker.html>.
- [3] P. Dolong, N. Henze, W. Nejdl, Logic-based open hypermedia for the semantic web, in: Proceedings of the International Workshop on Hypermedia and the Semantic Web, Hypertext 2003 Conference, Nottingham, UK, 2003.
- [4] J. English, M. Hearst, R. Sinha, K. Swearingen, K.-P. Lee, Flexible search and navigation using faceted metadata. Technical report, University of Berkeley, School of Information Management and Systems, 2003, submitted for publication.
- [5] D.J. Foskett, Thesaurus. Encyclopaedia of Library and Information Science, vol. 30, Marcel Dekker, New York, 1980, pp. 416–462.
- [6] C. Goble, S. Bechhofer, L. Carr, D. De Roure, W. Hall, Conceptual open hypermedia = the semantic web? in: Proceedings of the WWW2001, Semantic Web Workshop, Hongkong, 2001.
- [7] S. Handschuh, S. Staab, F. Ciravegna, S-CREAM: Semi-automatic CREATION of Metadata, in: Proceedings of the EKAW2002, LNCS, 2002, pp. 358–372.
- [8] S. Handschuh, S. Staab, R. Volz, On deep annotation, in: Proceedings of the International World Wide Web Conference, 2003, pp. 431–438.
- [9] M. Hearst, A. Elliott, J. English, R. Sinha, K. Swearingen, K.-P. Lee, Finding the flow in web site search, CACM 45 (9) (2002) 42–49.
- [10] J. Hefflin, J. Hendler, S. Luke, SHOE: a knowledge representation language for internet applications. Technical report, Department of Computer Science, University of Maryland, College Park, 1999.
- [11] L. Hollink, A.Th. Schreiber, J. Wielemaker, B.J. Wielinga, Semantic annotations of image collections, in: Proceedings of the KCAP'03, Florida, 2003.
- [12] E. Hyvönen, M. Holi, K. Viljanen, Designing and creating a web site based on RDF content, in: Proceedings of the WWW2004 Workshop on Application Design, Development, and Implementation Issues in the Semantic Web, New York, USA, CEUR Workshop Proceedings, vol. 105, 2004, <http://ceur-ws.org>.
- [13] E. Hyvönen, M. Junnila, S. Kettula, E. Mäkelä, S. Saarela, M. Salminen, A. Syreeni, A. Valo, K. Viljanen, Finnish museums on the semantic web. User's perspective on MUSEUMFINLAND, in: Proceedings of Museums and the Web 2004 (MW2004), Selected Papers, Arlington, Virginia, USA, 2004, <http://www.archimuse.com/mw2004/papers/hyvonen/hyvonen.html>.
- [14] E. Hyvönen, M. Junnila, S. Kettula, S. Saarela, M. Salminen, A. Syreeni, A. Valo, K. Viljanen, Publishing collections in the Finnish museums on the semantic web portal—first results, in: Proceedings of the XML Finland 2003 conference, Kuopio, Finland, 2003, <http://www.cs.helsinki.fi/u/eahyvone/publications/xmlfin-land2003/FMSOverview.pdf>.
- [15] E. Hyvönen, S. Kettula, V. Raatikka, S. Saarela, K. Viljanen, Semantic interoperability on the web. Case Finnish Museums Online, in: E. Hyvonen, M. Klemettinen (Eds.), Proceedings of the XML Finland 2002 Conference [16], 41–53, <http://www.hiit.fi/publications/>.

²² <http://www.inconclass.nl/>.

²³ http://www.getty.edu/research/conducting_research/vocabularies/aat/.

- [16] Towards the semantic web and web services, in: E. Hyvönen, M. Klemettinen (Eds.), Proceedings of the XML Finland 2002 conference, Helsinki, Finland, number 2002–03 in HIIT Publications, Helsinki Institute for Information Technology (HIIT), Helsinki, Finland, 2002, <http://www.hiit.fi/publications/>.
- [17] E. Hyvönen, S. Saarela, K. Viljanen, Application of ontology-based techniques to view-based semantic search and browsing, in: *The Semantic Web: Research and Applications*. First European Semantic Web Symposium, ESWS 2004, Heraklion, Greece, Springer-Verlag, Berlin, 2004, pp. 92–106.
- [18] E. Hyvönen, M. Salminen, M. Junnila, Annotation of heterogeneous database content for the semantic web, in: Proceedings of the 4th International Workshop on Knowledge Markup and Semantic Annotation (SemAnnot 2004), Hiroshima, Japan, November, 2004.
- [19] E. Hyvönen, M. Salminen, S. Kettula, M. Junnila, A content creation process for the Semantic Web, in: Proceedings of the OntoLex 2004: Ontologies and Lexical Resources in Distributed Environments, Lisbon, Portugal, 29 May, 2004.
- [20] E. Hyvönen, A. Styrman, S. Saarela, Ontology-based image retrieval, in: E. Hyvönen, M. Klemettinen (Eds.), Proceedings of the XML Finland 2002 Conference [16], 15–27, <http://www.hiit.fi/publications/>.
- [21] O. Lassila, R. Swick (Eds.), Resource description framework (RDF): model and syntax specification. Technical report, W3C, 1999. W3C Recommendation 1999-02-22, <http://www.w3.org/TR/REC-rdf-syntax/>.
- [22] M. Laukkanen, K. Viljanen, M. Apiola, P. Lindgren, E. Hyvönen, Towards ontology-based yellow page services, in: Proceedings of the WWW2004 Workshop on Application Design, Development, and Implementation Issues in the Semantic Web, New York, USA, CEUR Workshop Proceedings, vol. 105, 2004, <http://ceur-ws.org>.
- [23] R.L. Leskinen (Ed.), *Museoalan asiasanasto*, Museovirasto, Helsinki, Finland, 1997.
- [24] A. Maedche, S. Staab, N. Stojanovic, R. Studer, Y. Sure, Semantic portal—the SEAL approach. Technical report, Institute AIFB, University of Karlsruhe, Germany, 2001.
- [25] E. Mäkelä, E. Hyvönen, S. Saarela, K. Viljanen, Ontoviews—a tool for creating semantic web portals, in: Proceedings of the 3rd International Semantic Web Conference (ISWC 2004), Hiroshima, Japan, Springer-Verlag, Berlin, November 2004, pp. 797–811.
- [26] Steve Pepper. The TAO of Topic Maps, in: Proceedings of the XML Europe 2000, Paris, France, 2000. <http://www.ontopia.net/topicmaps/materials/rdf.html>.
- [27] T. Peterson, Introduction to the Art and Architecture Thesaurus, 1994. <http://shiva.pub.getty.edu>.
- [28] A.S. Pollitt, 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>.
- [29] J. Ben Schafer, Joseph A. Konstan, John Riedl, E-commerce recommendation applications, *Data Min. Knowl. Disc.* 5 (1/2) (2001) 115–153.
- [30] D. Soergel, B. Lauser, A. Liang, F. Fisseha, J. Keizer, S. Katz, Reengineering thesauri for new applications: the AGROVOC example, *J. Digital Inform.* (4) (2004).
- [31] L. Stojanovic, N. Stojanovic, R. Volz, Migrating data-intensive web sites into the semantic web, in: Proceedings of the ACM Symposium on Applied Computing SAC-02, Madrid, 2002, pp. 1100–1107.
- [32] Peter Stuer, Robert Meersman, Steven De Bruyne, The Hyper-Museum theme generator system: ontology-based internet support for active use of digital museum data for teaching and presentations, in: D. Bearman, J. Trant (Eds.), *Museums and the Web 2001: Selected Papers*. Archives and Museum Informatics, 2001. <http://www.archimuse.com/mw2001/papers/stuer/stuer.html>.
- [33] J. van den Berg, Subject retrieval in pictorial information systems, in: Proceedings of the 18th International Congress of Historical Sciences, Montreal, Canada, 1995, pp. 21–29, <http://www.iconclass.nl/texts/history05.html>.



III

Publication III

Eetu Mäkelä, Eero Hyvönen, and Teemu Sidoroff. 2005. View-Based User Interfaces for Information Retrieval on the Semantic Web. In: Abraham Bernstein, Ion Androutsopoulos, Duane Degler, and Brian McBride (editors), End User Semantic Web Interaction Workshop, volume 172 of *CEUR Workshop Proceedings*. CEUR-WS.org.

© 2005 by the authors

View-Based User Interfaces for Information Retrieval on the Semantic Web

Eetu Mäkelä, Eero Hyvönen and Teemu Sidoroff

Helsinki Institute for Information Technology (HIIT),
Helsinki University of Technology, Media Technology, and University of Helsinki
FirstName.LastName@tkk.fi, <http://www.cs.helsinki.fi/group/seco/>

Abstract. This paper¹ argues for using the multi-facet search paradigm as a basis in information retrieval on the Semantic Web. To support the argument, two user interfaces for extant semantic web portals based on the concept of view-hierarchies are presented. The interfaces described reveal and contrast how the view-based paradigm can be applied to support, in turn, both browsing and searching strategies in information retrieval in applications using different domain and annotation ontologies. New semantics-based user interface elements complementing the basic paradigm are also discussed.

1 Introduction

Hierarchies provide a natural and useful way of categorizing information. They are ubiquitously used in libraries, catalogs, web portals, etc. to structure repository contents of various kinds. The categorized items are not restricted to a *single* classification, but can be annotated into *multiple* categories at the same time. The various classifications can then be used both to constrain searches as well as to organize search results. The idea of multi-facet classification was proposed already in the 1930's by S. R. Ranganathan. However, most directory services on the web, such as Yahoo! and the Open Directory Project² are still mostly based of the single-facet classification paradigm. In the 1980's and 1990's the idea of multi-facet search was employed by the information retrieval research community [1] and found its way to the web [2] and was integrated with the idea of ontologies, reasoning, and the semantic web [3, 4].

On the semantic web, ontologies are used to describe information items. Ontologies typically contain hierarchical structures, most often defined with explicit relations, such as 'part-of' and 'subclass-of'. The most obvious explicit way to form a view hierarchy from an ontology is to project (transform) classes and the `rdfs:subClass` relations between them into a tree or an acyclic graph. The same ontology can be projected into several view hierarchies. For example, in [4], the geographical part-of ontology is projected into the views Place of usage and Place of manufacture providing two distinct views into a repository of museum collection objects. In general, ontological hierarchies provide a rich base for the projection of usable view hierarchies to be used in user interaction.

¹ This work was mainly supported by the National Technology Agency Tekes.

² <http://dmow.org>

This paper argues for using the multi-facet search paradigm, with appropriate user interfaces, as a basis in information retrieval on the Semantic Web. To support the argument, two complete user interfaces for extant semantic web portals based on the concept of view-hierarchies are presented. The systems were created with the ONTOVIEWS[5] toolkit. A major design principle underlying this tool was to separate the semantics of the view-based search paradigm and user interface from the underlying domain ontologies and annotation schemas by a layer of logic rules [3]. This idea holds the promise that the same view-based search and browsing engine can be easily adapted to different applications. A major goal of our work has been to test feasibility of this idea in practice by applying the tool to domains and interfaces of different kinds.

In the following, a general idea of view-hierarchy based querying and result set formation is presented. Then, the two different user interfaces are presented, primarily supporting the browsing and searching information retrieval strategies [6], respectively, though acknowledging that in actual use the two strategies most often intertwine [7]. In conclusion, related work is discussed.

2 View-Based Information Retrieval

Figure 1 shows a conceptual overview and an example of view-based querying and result formation in ONTOVIEWS. The left of the picture shows the organization of data in the application. The data consists of a number of hierarchical category trees, as well as data items that are annotated according to these categories. There may also be other indexing information related to the data items, such as keywords.

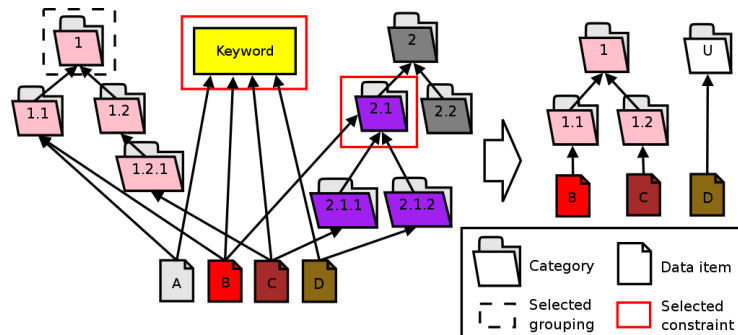


Fig. 1. View Hierarchy Based Querying and Result Organisation

Query constraining works as follows: When the user selects a category c in a facet f , the system constrains the search by leaving in the result set only such objects that are annotated in facet f with some sub-category of c or c itself. When an additional selection of category c_2 from another facet g is made the result set of the selection is the intersection of the items in the selected categories, i.e., $c \cap c_2$. Other constraints can

also be integrated in a similar manner. In figure 1, for example, the user has selected a combination constraint of a keyword and the category 2.1. The result set contains data items *B*, *C* and *D*, but not *A*.

After the result set is calculated, it can be organized again according to the view hierarchies for visualization. In the example, the category tree beginning from 1 has been selected for result grouping, even though it was not used as a search constraint. The grouping hierarchy is cut on the first sub-level. This configuration results in the grouping depicted on the right of figure 1. Item *C* is bumped up to its parent category, and item *D*, which has not been annotated anywhere within the grouping hierarchy, is shown within the dynamically created “ungrouped” category *U*.

3 A View-Based User Interface for Browsing

When a user’s information need is not articulated, either because formulating that need is difficult or because the exact goal is not well defined even in the user’s mind, collection browsing is a useful way of mapping the content of available data sources. The MUSEUMFINLAND³ [4] portal presents a collection of museum items that are described by a set of 7 RDF(S) ontologies, from which 9 different view hierarchies are projected. Being a virtual exhibition, the portal focuses very much on the browsing aspect of information retrieval.

The user interface of MUSEUMFINLAND owes much to the user interface studies conducted on the Flamenco system [8]. Figure 2 shows the main search interface of MUSEUMFINLAND. The nine facets are shown on the left (in Finnish), such as Artifact type (“Esinetyyppi”) and Material (“Materiaali”). For each facet, the next level of sub-categories is shown as a set of links. A category is added to the constraints by clicking on its name. Only categories whose selection will not lead to an empty set are presented for selection. Also, each facet are accompanied with the number of items in the result set that would result if the user was to select that category as a query constraint. The idea is to indicate *proactively* the user the query result set sizes the selectable categories. Currently effective constraints are show on the top right of the view (’Hakuehdot’), from where they can also be dismissed.

The current result set of items matching the constraints is seen on the right grouped by the direct sub-categories of the last selection. Hits in different categories are separated by horizontal bars and can be viewed page by page independently in each category. The system also supports grouping along arbitrary views by clicking on the ’(ryhmittele kohteet)’ (group targets) link next to each facet. Items in the result set that do not belong in any of the groups are gathered in an “Other hits” group.

Because at each step only the choices in the next level of sub-categories are shown, the query gets refined iteratively, with each step providing a manageable set of choices to choose from. In addition, at each level a result set of possibly interesting items is shown. This interaction strategy is especially suited for cases in which the user does not explicitly know what he is searching for, as it quickly gives the user an impression for what is contained in the portal collection. For example, looking at the main page

³ The portal is operational at <http://www.museosuomi.fi/> and includes an English online tutorial.

Käsittehaku: Hae tarkenna hakua

Hakusana: nokia (poista)

Valmistaja > ... > [Nokia](#) (14),
 Valmistuspaikka > ... > [Nokia](#) (32),
 Käyttöpaikka > ... > [Nokia](#) (4),
 Valmistaja > ... > [Oy Nokia Ab](#) (1),
 Valmistaja > ... > [Nokia-Mobira](#) (1),
 Valmistaja > ... > [Nokian Jalkinetehtas Oy](#) (7),
 Valmistaja > ... > [Nokian Kutomo ja Väriäys Oy](#) (2)

Esinetyyppi (koko luokittelu) (ryhmittele kohteet)

[kulkuneuvot ja kuljetusvälineet osineen](#) (1),
[koneet ja laitteet](#) (4),
[pukineet ja tekstiilit](#) (30),
[säilyttimet](#) (2), [leikkikalut](#) (2),
[sisustus](#) (1)

Materiaali (koko luokittelu) (ryhmittele kohteet)

[materiaalit](#) (37)

Valmistaja (koko luokittelu) (ryhmittele kohteet)

[yritykset](#) (34)

Hakuehdot

Hakusana: nokia (ryhmittele kohteet) (poista)

Kohteet ryhmiteltyinä hakusanan *nokia* mu
 (näytä ilman ryhmittelyä)

Valmistaja > [Nokia](#), kohteet 1-4/14 (ryhmittele kohteet)

	
leluauto:henkilöauto (ECM 3594 47)	leluauto:lelukilpa-auto (ECM 3598 1)

Valmistuspaikka > [Nokia](#), kohteet 1-4/32 (ryhmittele kohteet)

Fig. 2. The search interface of MUSEUMFINLAND.

of MUSEUMFINLAND, the user, not really looking for anything particular, may decide that he'll start with looking at items used in Europe. In the results, he then sees several chairs he likes, and decides to constrain his search to furnishing items used in Europe, and so on.

The user interface of MUSEUMFINLAND also provides an alternate view to the material and the facets of the application. Clicking the link Whole facet ("koko luokittelu") on any facet brings up a tree-view of the whole facet, along with the number of items in each category according to current constraints. This gives the user an overview of the distribution of items over a desired dimension. By graying out categories with no hits, it is also easy to see in what categories the collections are lacking artifacts. This may be a useful piece of information for, e.g., the collection manager.

To facilitate quick searching when the user knows what he is looking for MUSEUMFINLAND includes semantic keyword-searching functionality. This functionality is seamlessly integrated with view-based search in the following way: First, the search keywords are matched against category names in the facets in addition to text fields in the meta-data. A new dynamic facet is created in the user interface. This facet contains all facet categories whose name (or other property values) matches the keyword. Intuitively these facet categories tell the different interpretations of the keyword, and by selecting one of them a semantically disambiguated choice can be made. This also solves the search problem of finding relevant categories in facets that contain thousands of categories. A result set of object hits is also shown. This result set contains all objects contained in any of the categories matched in addition to all objects whose meta-data directly contains the keyword. The hits are grouped by the categories found. The view in figure 2 includes a keyword search facet for the word "nokia". Matched are, for example, the categories Nokia (the telephone company), Nokia (the place) and Nokia-Mobira (an earlier incarnation of the telephone company).

4 A View-Based User Interface for Searching

Searching, most often currently realized through keyword searching, provides the user with a fast way to reach their goal, provided that he knows what he is looking for, and additionally knows how to describe it in the terms that the search engine requires. Yellow pages directories is a domain where one can often expect users to know what they are looking for. There are no guarantees however, that the user can formulate their queries accordingly. In the Intelligent Web Services (IWebS) project⁴, the yellow pages service portal Veturi was created to address this search problem. The portal contains some 220,000 real-world services from both the public and private sectors, annotated semantically with a SUMO [9]-based service ontology.

The user interface of Veturi is based on on-the-fly semantic autocompletion of keywords into categories, made possible by the use of AJAX⁵ techniques. This user interaction pattern tightly integrates keyword searching with the specificity, semantic disambiguation and context visualisation capabilities of the view facets, as described in the following.

Figure 3 depicts the search interface of the Veturi portal. The five view-facets used in the portal (Consumer, Producer, Target, Process, and Location of the Service provided) are located at the top, initially marked only by their name and an empty keyword box. Typing search terms in the boxes immediately opens the corresponding facet to show matching categories. The results view below the facets also dynamically updates to show relevant hits, defined by the current search constraints in other facets and a union of all the categories in the current facet matched by the keyword. If there is a need for more specificity or an alternate selection, a single category can be selected from the facet. After such a selection, the facet again closes, showing only the newly selected constraint, with the results view updating accordingly.

The user is guided in formulating his query by focusing the views on clearly identifiable distinct variables of the service. For users more familiar with the portal and its service description model, a globally effective keyword search box is provided in the upper left corner of the interface for quick, undifferentiated searches. Because in the service model used the contents of the views seldom overlap, most queries can be adequately and precisely replied to simply by typing the service need in plain text in the global keyword box, e.g. “car repair helsinki”, with possible disambiguations done through selections in the facets.

The example search depicted in figure 3 shows a user trying to find out where he can buy rye bread in Helsinki. He has already selected Helsinki as the locale for the services he requires. Now, he is in the process of describing the actual service. In the view ‘Mitä?’ (service target), the user has typed in the word ‘ruis’ (rye). While the annotation ontology used does not contain different grains, the concept ‘Viljatuotteet ja Leipä (KR)’ (grain products and bread) contains a textual reference to rye, resulting in a category match. In this way, existing textual material can be used to augment incomplete

⁴ <http://www.cs.helsinki.fi/group/iwebs/>

⁵ Asynchronous JavaScript and the XMLHttpRequest -object, which allow for making HTTP calls to the server in the background while viewing a page. See e.g. <http://en.wikipedia.org/wiki/AJAX>.

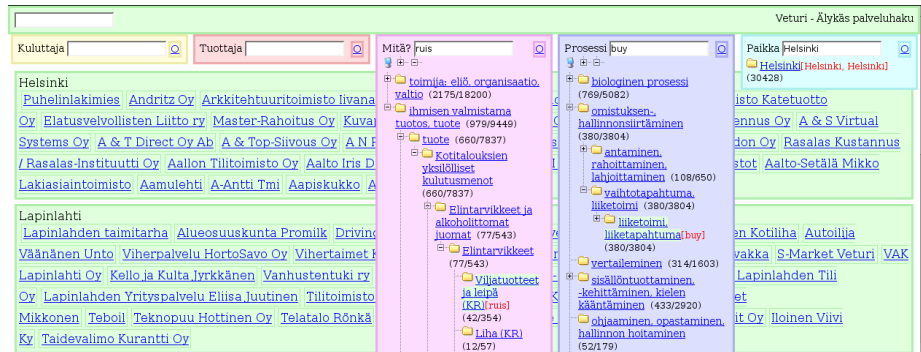


Fig. 3. The Veturi user interface.

ontologies to at least return some hits for concepts that have not yet been added to the ontology.

In the figure, the matched categories are shown directly in their hierarchical contexts. This allows for quick evaluation of the relevance of the hits, as well as reveals close misses, where for example the keyword matches a sub-category of a more appropriate one, as in when the common-language word 'vitamin' has been given when actually the whole category of dietary supplements was meant. As a side effect of viewing the trees, the user is also guided on the content of the collection and how it is indexed in the system. The trees can also be opened and navigated freely without using keywords for an alternate form of navigation and familiarization with the indexing concepts and facets.

The search query entered in the view 'Prosessi' (Process) divulges an additional feature of the portal: multilanguage support. Typing in the word 'buy' matches the appropriate 'liiketoimi, liiketapahtuma' (business transaction), even though the word for 'buy' in Finnish would be 'ostaa'. The implementation also supports the T9-type ambiguous numerical queries [10, 11] common in mobile phone text input environments. These extensions were implemented to demonstrate how the core semantic autocompletion interface can easily be combined with other advances in predictive text autocompletion, because the ontological navigation happens separately after string matching, similarly to what is described in [12].

5 Complementing User Interface Elements

In addition to the basic view-based interfaces depicted above, ONTOVIEWS portals can be enhanced with several features that can aid in presenting the content. All of these features deal with what happens after an interesting data item is located, providing additional semantics-based browsing options from the viewpoint of that item.

The data items that make up the content of ONTOVIEWS portals can be joined to create *compound items*. These are collections of items that are then displayed on a single web page in a suitable manner. By using compound items it is possible to collect

together complexly interrelated information from different sources and present it in a clear way as a single unit. The compound items can be created manually by the portal maintainers, or through logic rules operating on the ontology level.

The compound item can be visualized in different ways, depending on the nature of the collection. In the e-government SW-Suomi.fi -portal [13], a simple list was deemed sufficient. Here, for example, a compound item of traveling abroad contains information about acquiring a passport, possible vaccinations needed and so forth.

Another possible visualization format are graphs. This has been experimented with in prototypes of CultureSampo, the successor to MUSEUMFINLAND, where the idea is to use process descriptions as the central glue joining together a wide variety of cultural content. For example, finding a process of farming that interests him, a user can go through a graph visualization of it, at each step seeing all the material relating to that particular step of the process, e.g. videos on sowing wheat, museum implements used in the clearing the field or a poem about plowing.

The ONTOVIEWS system also supports direct horizontal navigation between the items via semantic links so users can browse through the portal content without having to return back to the category level.

These semantic links are generated automatically by a set of domain specific category rules. The rules act as shortcuts between two items in the knowledge base. A rule fires when the two items are connected through an RDF path defined in the rule and creates a direct hyper-link between the items. A labeling system is used to provide context for the link, containing both the meaning of the rule and the details of the actual link, e.g. 'Common theme: Christmas' for two related museum items, or 'Nearby Taxi Companies: Espoo' related to the details-page of a car repair shop.

6 Discussion and Related Work

We argued for using multi-facet search as a basis for searching and browsing on the semantic web, and presented different user interfaces to support the argument. The interfaces respond to different use case requirements, but are sufficiently similar to maintain familiarity. In both cases, the facet hierarchies give the user an overview of what kind of information there is in the data repository in terms of hierarchical vocabularies, and guide the user in formulating the query in terms of appropriate concepts. According to user tests [8, 14], the multiple views, as well as the semantic firmness and specificity of the categories are desirable qualities when doing more complex searches.

Many ideas for the interfaces presented are similar to [1], [8] (MUSEUMFINLAND) and [14] (Veturi). ONTOVIEWS however, is based on taking advantage of the semantics of the data, which allows us to easily extend the interfaces. For example, the views are projected from ontological hierarchies while a semantic linking facility based on the ontologies and logic recommendation rules is provided, as well as semantic auto-completion. Multi-facet search has more recently been exploited, e.g., in Longwell, in the RDF Browser by the Simile project⁶, and in the SWED [15] directory portal. First commercial implementations for multi-facet search exist.

⁶ <http://www.simile.mit.edu/>

References

1. 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>.
2. Hearst, M., Elliott, A., English, J., Sinha, R., Swearingen, K., Lee, K.P.: Finding the flow in web site search. *CACM* **45** (2002) 42–49
3. Hyvönen, E., Saarela, S., Viljanen, K.: Application of ontology-based techniques to view-based semantic search and browsing. In: *Proceedings of the First European Semantic Web Symposium, May 10-12, Heraklion, Greece, Springer-Verlag, Berlin* (2004)
4. Hyvönen, E., Junnila, M., Kettula, S., Mäkelä, E., Saarela, S., Salminen, M., Syreeni, A., Valo, A., Viljanen, K.: Finnish Museums on the Semantic Web. User's perspective on MuseumFinland. In: *Proceedings of Museums and the Web 2004 (MW2004), Selected Papers, Arlington, Virginia, USA. (2004)* <http://www.archimuse.com/mw2004/papers/hyvonen/hyvonen.html>.
5. Mäkelä, E., Hyvönen, E., Saarela, S., Viljanen, K.: OntoViews - A Tool for Creating Semantic Web Portals. In: *Proceedings of the Third International Semantic Web Conference (ISWC2004), Hiroshima, Japan, Springer Verlag* (2004)
6. Sellen, A., Murphy, R., Shaw, K.L.: How Knowledge Workers Use the Web. In: *Proceedings of the SIGCHI conference on Human factors in computing systems, CHI Letters 4(1), ACM* (2002)
7. Teevan, J., Alvarado, C., Ackerman, M.S., Karger, D.R.: The Perfect Search Engine is not Enough: A Study of Orienteering Behavior in Directed Search. In: *Proceedings of the 2004 conference on Human factors in computing systems, ACM Press* (2004)
8. Lee, K.P., Swearingen, K., Li, K., Hearst, M.: Faceted metadata for image search and browsing. In: *Proceedings of CHI 2003, April 5-10, Fort Lauderdale, USA, Association for Computing Machinery (ACM), USA* (2003)
9. Niles, I., Pease, A.: Towards a standard upper ontology. In: *FOIS '01: Proceedings of the international conference on Formal Ontology in Information Systems, New York, NY, USA, ACM Press* (2001) 2–9
10. Dunlop, M.D., Crossan, A.: Predictive text entry methods for mobile phones. *Personal Technologies* **4** (2000)
11. Hasselgren, J., Montnemery, E., Nugues, P., Svensson, M.: Hms: A predictive text entry method using bigrams. In: *Proceedings of the Workshop on Language Modeling for Text Entry Methods, 10th Conference of the European Chapter of the Association of Computational Linguistics, Budapest, Hungary, Association for Computational Linguistics* (2003) 43–49
12. Legrand, S., Tyrväinen, P., Saarikoski, H.: Bridging the word disambiguation gap with the help of owl and semantic web ontologies. In: *Proceedings of the Workshop on Ontologies and Information Extraction, Europlan 2003. (2003)* 29–35
13. Sidoroff, T.: Semantic web portals (in Finnish, Semanttiset portaalit). Master's thesis, University of Helsinki, Dept. of Computer Science (2005) <http://www.cs.helsinki.fi/group/seco/publications>.
14. Zhang, J., Marchionini, G.: Evaluation and evolution of a browse and search interface: relation browser. In: *dg.o2005: Proceedings of the 2005 national conference on Digital government research, Digital Government Research Center* (2005) 179–188
15. Reynolds, D., Shabajee, P., Cayzer, S.: Semantic Information Portals. In: *Proceedings of the 13th International World Wide Web Conference on Alternate track papers & posters, ACM Press* (2004)



IV

Publication IV

Eero Hyvönen and Eetu Mäkelä. 2006. Semantic Autocompletion. In: Riichiro Mizoguchi, Zhongzhi Shi, and Fausto Giunchiglia (editors), *The Semantic Web - ASWC 2006, First Asian Semantic Web Conference, Beijing, China, September 3-7, 2006, Proceedings*, volume 4185 of *Lecture Notes in Computer Science*, pages 739–751. Springer. ISBN 3-540-38329-8.

© 2006 Springer Verlag Berlin Heidelberg

Semantic Autocompletion

Eero Hyvönen and Eetu Mäkelä

Semantic Computing Research Group (SeCo)
Helsinki University of Technology (TKK), Laboratory of Media Technology
University of Helsinki, Department of Computer Science
FirstName.LastName@tkk.fi
<http://www.seco.tkk.fi/>

Abstract. This paper generalizes the idea of traditional syntactic text autocompletion onto the semantic level. The idea is to autocomplete typed text into ontological categories instead of words in a vocabulary. The idea has been implemented and its application for semantic indexing and content-based information retrieval in multi-facet search is proposed. Four operational semantic portals on the web using the implementation are presented as application cases.

1 Introduction

The idea of *autocompletion*¹ is to predict what the user is typing in, and to complete the work automatically. The benefits of this simple idea are manyfold: First, the computer helps the user in memorizing the right vocabulary used. Second, typing errors in the input can be minimized. Third, autocompletion speeds up the interaction. A side effect of the idea is that it encourages the usage of long descriptive names and commands that are more understandable to the users. An idea related to autocompletion is *autoreplace*, where the idea is to use predefined abbreviations in typing and the system automatically replaces these with full-blown strings.

In order to make the prediction right and as early as possible, the underlying vocabulary must be known, be limited, and the words in the lexicon should differ from each other in terms of the leading characters. These conditions hold in many applications, such as operating system shells, email programs, browsers, etc.

Autocompletion is used, e.g., in Microsoft's Intellisense feature of the Visual Studio, where the idea is applied to source code editing. Here a pop-up menu is used to show the programmer possible auto-completed forms. This is useful when it is difficult to remember or type in, e.g., the names of the methods of a particular class at hand. A widely used application of autocompletion is the predictive text entry system in mobile phones [1,2] commonly known as T9, where only a limited number of keys are available instead of the full QWERTY keyboard. By associating each key with a set of letters (e.g. '1' with a, b, and c)

¹ See e.g. <http://en.wikipedia.org/wiki/Autocompletion>

and by completing single keypresses automatically based on a dictionary, input typing can be speeded up significantly e.g. in text messaging.

Autocompletion can be done *by request* or *on-the-fly*. In Linux/Unix and DOS operating systems, for example, the command line is completed—or possible continuations are shown—after a hit on the TAB-key. The on-the-fly-approach is used e.g. in browsers and email-systems: the text typed in is completed into matching URLs or email addresses that have been used before, or are stored in an address book. A nice recent application of autocompletion on-the-fly on the web is the beta version of Google Suggest² that completes input text into feasible search keywords.

Traditional autocompletion is based on matching input strings with a list of usable words in a vocabulary. This paper generalizes this approach onto the semantic level. The idea is to complete user written text not only into similar words, but into matching ontological concepts whose labels may not be related to the input on the literal level. For example, the typed input 'preside...' could be autocompleted into 'George W. Bush' since George W. Bush is an instance of the class president. It is also possible to complete the input text into the different homonymous meanings (concepts) of the input, and into the different semantic roles in which the concepts are used. This possibility provides the end-user not only with a semantic matching service but can be used to disambiguate the meanings and thematic roles in which the concepts are used. To continue the example above, input 'preside...' could be autocompleted into 'George W. Bush (as an author)' or 'George W. Bush (as a document subject)'. By providing the autocompleted choices to the end-user, the right interpretation can be disambiguated and, for example, search be performed with the right meaning.

In the following, this idea to be called *semantic autocompletion* is first discussed as a means for semantic information retrieval, and some of its different forms are identified. After this, implementation of the idea in the OntoViews framework [3] is presented, and application in three semantic portals for concept-based information retrieval and in semantic indexing is exemplified.

2 From Syntactic to Semantic Completion

We consider the idea of semantic autocompletion in information retrieval, especially, in multi-facet search [4,5,6,7]. Multi-facet search is a generalized form of the traditional single-facet search paradigm. Examples of single-facet systems include Yahoo!, Open Directory Project³, and many traditional web portals. In multi-facet search, content is organized and retrieved using multiple hierarchical structures at the same time, instead of just one like in single-facet search.

2.1 Autocompletion in Multi-facet Search

In multi-facet systems the data has been indexed using keywords from a set of hierarchical orthogonal facet categories. For example, in [5] the facet categories

² <http://www.google.com/webhp?complete=1&hl=en>

³ <http://dmoz.org/>

of the Art and Architecture Thesaurus AAT⁴ are used as subject terms. The location facet divides the earth into continents (Africa, Antarctica, Asia, ...), each continent consists of countries, and each country is divided further into counties, cities, etc. The material facet is a classification hierarchy of materials used or depicted in the collection items. The search objects are classified along facets based on the keywords used in annotating the collection items. The user selects categories from different facets and the search result is the intersection of the items belonging to the selected categories. By selecting a supercategory, all hits related to its subcategories (recursively) are returned, too. Let mapping $m : S \rightarrow C$ map each search item $s \in S$, where S is the set of search times, to the set of facet categories C . Then the hit set H corresponding to selected search categories c_1, \dots, c_n is $H = \{s | c_i \in m(s), i = 1, \dots, n\}$.

In traditional multi-facet search, the keywords are strings as usual in keyword search. In [6] multi-facet search is extended with semantic web ontology techniques and reasoning. The idea is to replace keywords with ontological resources in indexing and then determine the mapping m between search categories and search items using logical mapping rules. In this way, multi-facet search can be generalized onto a semantic level where the mapping between facets and search items can be based on semantic relations and not only on simple keyword match. For example, in [8] the category 'Nokia' as a company in an actor facet is mapped onto different search items than 'Nokia' as a city in Finland in a location facet.

Semantic autocompletion in multi-facet search can be defined as a function $f : text \rightarrow \langle C, H \rangle$ that maps an input string $t \in text$ onto a set of search categories of the facets C and the corresponding search item hits H in the data set. The hits are based on the different semantic meanings of the input. For example, if the user types in the word 'bank', this could be completed into categories 'river bank' and 'bank (financial)' and the result set includes an union of both geographical and organizational hits.

The input may consist of several partly written keywords that correspond to category selections. For example, 'Finl presid' could mean that the user searches information about categories 'Finland' and 'president', e.g., about the presidents of Finland. The categories C and hits H matching the input should, in the user's view, match in meaning with the intended meanings of *text*. For example, input 'Scandin...' may match the category 'Nordic countries'. Notice that here 'Scandinavia' and 'Nordic countries' do not share substrings as required in traditional autocompletion. In our case, autocompletion is occurring on the semantic level in the user's mind, and is implemented using the underlying ontological structures.

Autocompleting an input string into facet categories can be based on several principles. In below, some forms of autocompletion are discussed.

2.2 Autocompletion Based on Equivalence Relations

This form of autocompletion deals with the problems of lexical variants, synonymy, polysemy, and homonymy. Lexical variants and synonyms are alternative terms

⁴ http://www.getty.edu/research/conducting_research/vocabularies/aat/about.html

that correspond to the same ontological concept. For example, 'NYC', 'New York City', and 'Big Apple' refer to the same city. Semantic autocompletion can provide a service, where typing in any of the terms is completed into the same concept, denoted by its preferred term, here 'New York City'.

This kind of autocompletion can be enabled to some extent by listing alternative and preferred labels for concepts. If the input matches any of these, the corresponding concept is selected, and the preferred label is shown to the user. However, in morphologically rich languages, such as a Finnish, listing all morphological variants as explicit alternatives may not be feasible, and dynamic morphological analysis may be needed as a part of autocompletion before ontological matching. For example, the genitive plural form of the Finnish word 'yö' (night) is 'öiden', a literal quite different from the nominative form.

In polysemy, a single term has different but related meanings (e.g., 'arrow head' and 'human head'); in homonymy the meanings are totally different (e.g., 'river bank' and 'blood bank'). In both cases, the meaning cannot be disambiguated based on the user's shorthand input ('head' or 'bank'). The same happens when the user's partial input can be completed in different ways (e.g., 'New' \mapsto 'New York' or 'New' \mapsto 'New Year'). In these cases the autocompletion function can provide the user with a list of possible choices from which to disambiguate.

One problem in determining the equivalence between input text and categories is how to deal with phrasal concept labels, such as 'broadband integrated services digital network'. Here, the categories can be matched against all permutations, and only the combinations leading to actual hits returned, so that for example the search can return the two-category combination 'Integrated Services + Digital Network (11 hits)' as a reasonable autocompletion, while the two-category combination 'Broadband Integration + Digital Services (0 hits)' is left out. In such complex multi-word labels, words may also appear in morphologically conjugated forms, which makes pattern matching more difficult, again possibly requiring morphological analysis as a pre-step. On the user interface level, one must also remember that particularly for compound words the matching part may not necessarily begin the input string, so that the prefix matching is not sufficient, but the whole string needs to be scanned for matches.

In multilingual autocompletion the keywords can be expressed in different languages and be matched on the same concept. This facilitates multilingual search even when the actual data is available or has been indexed in one only language. For example, 'bank (financial)' \mapsto 'pankki (Finnish)'.

A benefit of semantic autocompletion is that the ontological environment of the matched categories can be visualized in addition to the actual matches. By showing the category hierarchy leading to the matched concept, the user can easily understand the meaning of the different completions. Furthermore, she can complete the text into the superclass or related concepts. For example, 'bank' \mapsto 'financial institution > bank', where '>' indicates the subclass relation in the hierarchy.

2.3 Indirect Semantic Autocompletion

Semantic completion can be extended beyond equality to other semantic relations. The input string can be matched with not only the corresponding equivalent category, but with other related categories, too. For example, assume that you are looking for information about countries. By typing in 'EU' or 'US' semantic autocompletion could complete the text into a choice list of member countries of EU or states of the US, respectively, saving the effort of memorizing their names. Here the *isPartOf*-relation to *is* is used for completing the text into neighboring ontological resources. However, in principle any arbitrarily complex relation could be used here, as long as its interpretation is intuitive and of use to the end-user.

2.4 Semantic Role Completion

An application of semantic autocompletion is *semantic role completion*. Here we not only match the input text with categories but also take into account the roles in which the categories are used. For example, the same city may be related with a museum collection artifact either as the place of manufacture or as the place of usage in the metadata. Depending on the choice, different result sets are obtained (unless all relevant items are both manufactured and used in the same place). Semantic autocompletion can provide the user with the possible choices to disambiguate.

2.5 Semantic Autocompletion Search

Semantic autocompletion can be combined seamlessly with semantic search. By completing the input string not only in related categories but also into the actual hits in the underlying data set, the user can actually see the hit list to narrow down as she types in text.

3 Application of Semantic Autocompletion

In the following we show by examples from various case studies, how the different forms of semantic autocompletion can be realized in practise in semantic information retrieval and indexing.

3.1 Semantic Category Search: Case MuseumFinland

Autocompletion can be used to disambiguate meanings in queries. This is useful especially if the content searched for has been annotated using correspondingly disambiguated concepts. An example of such a system is the semantic portal MUSEUMFINLAND⁵ [7]. We have incorporated a version of semantic autocompletion into this application.

⁵ <http://www.museosuomi.fi>

MUSEUMFINLAND integrates semantic autocompletion with multi-facet search. The search keywords are matched not only against the actual textual item descriptions, but also the labels and descriptions of the ontological categories by which they are annotated and organized into the view facets. As a result of semantic autocompletion, a new “dynamic facet” is created in the user interface. This facet contains all categories whose name or other configurable property value, such as alternative labels, match the keyword. Intuitively, the dynamic facet categories tell 1) the different interpretations of the keyword and 2) their roles with respect to the search items (here museum collection artifacts) in the metadata.

The result of a sample keyword search is shown in figure 1. Here, a search for input “nokia” has matched, for example, the following view categories:

- ‘Nokia’ as the telephone company and a manufacturer in the view Manufacturer (‘Valmistaja’ in the screenshot),
- ‘Nokia’ as a town in the view Place of Manufacture (‘Valmistuspaikka’),
- ‘Nokia’ as a town in the view Place of Usage (‘Käyttöpaikka’), and
- ‘Nokia-Mobira’, a predecessor of the telephone company, in the view Manufacturer.

By default, search is done by using the union of all possible interpretations. Search results are shown and classified according the possible choices on the right in the figure. However, the categories found can be used to constrain the multi-facet search further, with the distinction that selections from the dynamic facet replace selections in their corresponding facets and dismiss the dynamic facet. The right interpretation is selected by clicking on the corresponding link in the dynamic facet.

The screenshot displays a search interface with the following elements:

- Search Bar:** "Käsithaku:" followed by a text input field containing "nokia", a "Hae" button, and a "tarkenna haku" checkbox.
- Hakusana:** A yellow bar showing "Hakusana: nokia (poista)".
- Facets:** A list of search facets with counts:
 - Valmistaja > ... > [Nokia](#) (14),
 - Valmistuspaikka > ... > [Nokia](#) (32),
 - Käyttöpaikka > ... > [Nokia](#) (4),
 - Valmistaja > ... > [Oy Nokia Ab](#) (1),
 - Valmistaja > ... > [Nokia-Mobira](#) (1),
 - Valmistaja > ... > [Nokian Jalkinetehdas Oy](#) (7),
 - Valmistaja > ... > [Nokian Kutomo ja Värjäys Oy](#) (2)
- Esinetyppi:** A yellow bar with "Esinetyppi (koko luokittelu) (ryhmittele kohteet)" and a list of categories:
 - [kulkuneuvot ja kuljetusvälineet osineen](#) (1),
 - [koneet ja laitteet](#) (4),
 - [pukineet ja tekstiilit](#) (30),
 - [säilyttimet](#) (2),
 - [leikkikalut](#) (2),
 - [sisustus](#) (1)
- Materiaali:** A yellow bar with "Materiaali (koko luokittelu) (ryhmittele kohteet)" and [materiaalit](#) (37).
- Valmistaja:** A yellow bar with "Valmistaja (koko luokittelu) (ryhmittele kohteet)" and [yritykset](#) (34).
- Hakuehdot:** A yellow bar with "Hakuehdot" and "Hakusana: nokia (ryhmittele kohteet) (poista)".
- Results:** A section titled "Kohteet ryhmiteltyinä hakusanan nokia mu (näytä ilman ryhmittelyä)" showing a list of items:
 - Valmistaja > [Nokia](#), kohteet 1-4/14 (ryhmittele kohteet)
 Below this are two images of toy cars:
 -  leluauto:henkilöauto (ECM 3594 47)
 -  leluauto:lelukilpa-auto (ECM 3598 1)
- Bottom Facet:** A yellow bar with "Valmistuspaikka > [Nokia](#), kohteet 1-4/32 (ryhmittele kohteet)".

Fig. 1. Using the keyword search for finding categories

In MUSEUMFINLAND, semantic autocompletion can be seen as search over a set of RDF(S) categories that correspond to classes in the underlying ontologies. At the same time, also hit lists of museum collection items are generated. This idea expanding queries over hierarchies has been applied also, e.g., in the Open Directory Project search engine. However, in our case the 9 category views have been projected, using a set of logical rules, from a set of 7 underlying ontologies in the system knowledge base. Matching is not straight-forward because of the projection, but indirect and more flexible. For example, in the search results of figure 1, the category 'Nokia' appears twice as a place (town). This is because the category can appear in the content of the portal in two different roles. Simply choosing e.g. the category 'Nokia (the place)' would not disambiguate the meaning sufficiently, since the same resource has the role of place of manufacture (Valmistuspaikka>...>Nokia) or place of usage (Käyttöpaikka>...>Nokia), or both, in the metadata of the museum artifacts. In the case of MUSEUMFINLAND, these roles can be disambiguated automatically by semantic autocompletion: the user can choose from a list of given options the correct role meaning of the keyword 'nokia' indicated by the subcategory path leading to it.

3.2 Semantic Autocompletion on the Fly: Case Orava

In MUSEUMFINLAND autocompletion is done on request, i.e., after pushing the search button. We have also created an on-the-fly version of the idea and applied it to another semantic portal Orava⁶[9]. This portal provides the user with semantic search and browsing facilities similar to MUSEUMFINLAND but to a database of some 2200 video and audio clips⁷ and learning object metadata (LOM)⁸ related to them.

Figure 2 depicts the home page of the portal with the on-the-fly semantic autocompletion in action in the upper right corner. The user has typed in the characters 'mat', aiming perhaps at the word 'matkailu' (travel). The autocompletion function dynamically and automatically updates the category trees below as selectable links. It shows all facet categories matching the typed characters used in the multi-facet search. The facets, such as 'Oppiaine' (learning subject) and 'Teema' (theme), and their uppermost levels of subcategories are seen on the left hand side column.

Continuing by typing the letter 'k' would eliminate the category 'matematiikka' (mathematics) as no longer matching, updating the trees accordingly. Alternately, at any point the user can select a link in the dynamic facet, and the system retrieves all material related to the selected category or any of its subcategories. The presentation of the retrieved categories as trees gives the user the context necessary to make informed selections, as well as makes it possible to make a broader search by selecting some supercategory of the ones matched.

⁶ <http://www.museosuomi.fi/orava/>

⁷ The material is from the Klaffi portal (<http://www.yle.fi/klaffi/>) of the Finnish Broadcasting Company YLE.

⁸ <http://ltsc.ieee.org/wg12/>

The screenshot shows the Orava website interface. On the left, there is a navigation menu with categories like 'Oppiaine', 'Teema', 'Kohderyhmä', and 'Vaativuus'. The main content area is titled 'Orava Opetusvideoiden haku- ja suosittelukone'. It contains a search bar with the text 'mat' and a 'Hae' button. Below the search bar, there is a list of results for 'mat'. The results are organized into a tree structure. The 'Oppiaine' category is expanded to show sub-categories: 'Matematiikka (38)', 'Kielitiede (147)', 'Kielioppi (10)', 'Liikunta (14)', 'Luonnontieteet (163)', 'Matematiikka (38)', 'Musiiikki (15)', 'Terveystieto (20)', 'Uskonto (67)', and 'Yhteiskuntatieteet (106)'. The 'Teema' category is expanded to show 'Koulutus (183)', 'Luonto (123)', 'Työ (139)', 'Yhteiskunta (421)', and 'Yksilö (244)'. The 'Kohderyhmä' category is expanded to show 'Alkuiset (1382)', 'Ala-aste (351)', 'Ei-koulun (26)', 'Lukio tai muu toinen aste (1130)', and 'Yläaste (356)'. The 'Vaativuus' category is expanded to show 'Helppo (896)' and 'Keskivaikea (988)'. Below the search bar, there is a section titled 'mat' with a list of results: 'Oppiaine (38)', 'Ohjelmasarja (35)', 'Teema (23)', and 'Kohteita (23)'. The 'Oppiaine' category is expanded to show 'Matematiikka (38)'. The 'Ohjelmasarja' category is expanded to show 'Kieliohjelmat (35)', 'Kreikkaa matkailijoille (14)', and 'Portugalia matkailijoille (14)'. The 'Teema' category is expanded to show 'Yksilö (23)', 'Vanaja-alku (23)', and 'Matkailu (23)'. The 'Kohteita' category is expanded to show 'Etälukio: matematiikka, tilastotiede, keskiarvo ja keskitieto', 'Etälukio: matematiikka, tilastotiede, normaaliarvo', and 'Etälukio: matematiikka, tilastotiede, hypoteesi'.

Fig. 2. Semantic autocompletion on-the-fly in Orava

Below the dynamic autocompleted category tree, a dynamic hit list that consists of the union of all video and audio clips matching 'mat' is also shown for the direct selection of a particular item. As in MUSEUMFINLAND, autocompletion is here extended to actually searching the contents, but this time on-the-fly.

3.3 Semantic Autocompletion Facet by Facet: Case Veturi

In the semantic yellow page portal Veturi [10], created in the Intelligent Web Services (IWebS) project⁹, the integration between view hierarchy based search and on-the-fly semantic autocompletion is taken even further. For this portal, on-the-fly semantic autocompletion was chosen as the central user interface element. The portal makes ample use of otherwise invisible metadata to match typed-in keywords to categories, as will be shown below.

Figure 3 depicts the search interface of the Veturi portal. The five view-facets used in the portal are Consumer ('Kuluttaja'), Producer ('Tuottaja'), Target ('Mitä?'), Process ('Prosessi'), and Location of the Service ('Paikka'). The views are located on the top horizontally, initially marked only by their name and an empty keyword field. Typing search terms in the fields immediately opens the corresponding facet to show matching categories available for selection. After such a selection, the facet closes again, showing only what was selected, while the results view below the facets dynamically updates to show relevant hits. For quick searches, a globally effective keyword search box is provided in the upper left corner of the interface. In this box it is possible to write a sequence of (possible partial) keywords, e.g. 'buy marmelade', that are completed one after another against the views.

The example search depicted in figure 3 shows the user trying to find out where he can buy rye bread in Helsinki. He has already selected Helsinki as the

⁹ <http://www.seco.tkk.fi/projects/iwebs/>

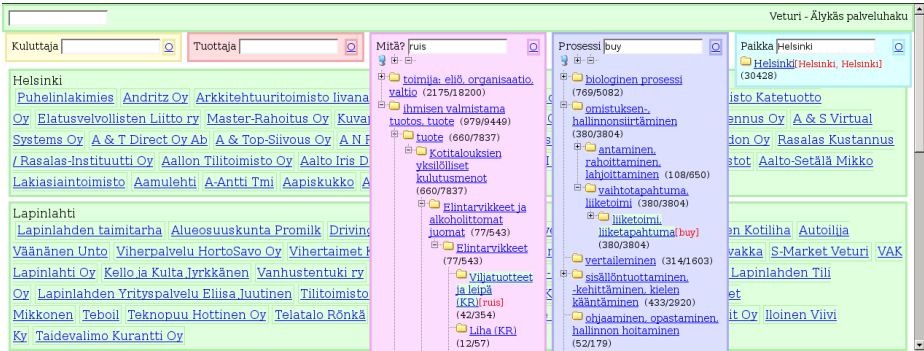


Fig. 3. Semantic autocompletion on-the-fly in Veturi

locale for the services he requires. Now, he is in the process of describing the actual service.

In the view Target view ('Mitä?'), the user has typed in the word 'rye' ('ruis'). While the annotation ontology used does not contain different grains, the concept 'grain products and bread' ('Viljatuotteet ja Leipä (KR)') contains a textual reference to rye, resulting in a category match. In this way, existing textual material can be used to augment incomplete ontologies to at least return some hits for concepts that have not yet been added into the ontology. Showing such hits in their ontological context allows for easy spotting of irrelevant hits and close misses, where for example the keyword matches a subcategory of a more appropriate one.

The search query entered in the view Process ('Prosessi') divulges another feature of semantic autocompletion: multilanguage support. Typing in the word 'buy' matches the appropriate business transaction, even though the word for 'buy' in Finnish would be 'ostaa'.

3.4 Semantic Indexing: Case ONKI Ontology Server

ONKI [11] is a part of the "Finnish National Ontologies on the Semantic We" (FinnONTO)¹⁰ framework project. Its goal is to support the development and use of nationally shared ontologies in order to enhance semantic interoperability on the Finnish semantic web. A central part of FinnONTO research deals with providing ontology services through public web services. For a content indexer, the ONKI ontology server¹¹ provides a web-based browser for finding desired concepts. Semantic autocompletion has been implemented as a part of a demonstrational ONKI service.

The interface is analogous to the one in the Orava portal. In figure 4, the user has typed in the regular expression '*housu' (trouser), where '*' matches any sequence of characters, and ONKI browser has completed the input into several

¹⁰ <http://www.seco.tkk.fi/projects/finnonto/>

¹¹ <http://www.seco.tkk.fi/applications/onki/>

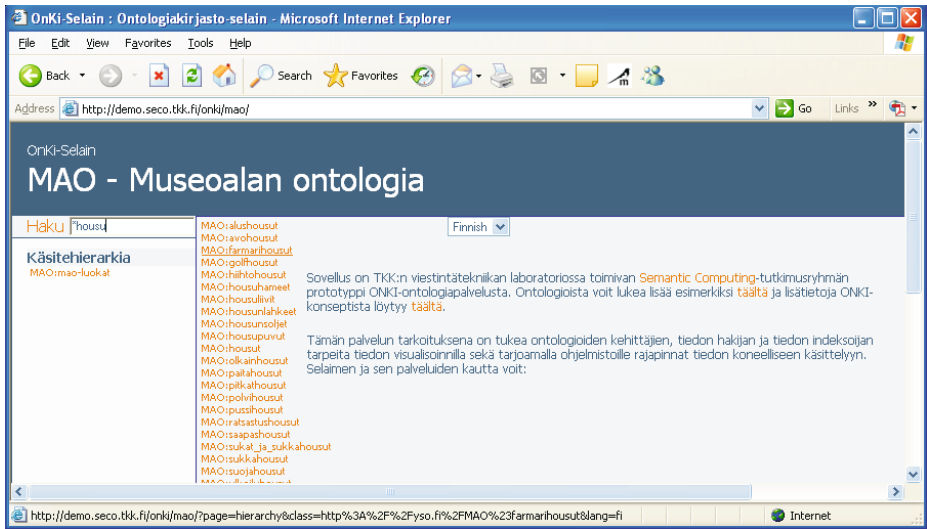


Fig. 4. Semantic autocomplete in the ontology server ONKI

concept categories of different types of trousers defined in the underlying cultural ontology MAO of the MUSEUMFINLAND portal. After selecting a concept by clicking on, the semantic neighborhood of the concept can be browsed further, if needed. Using ONKI, data of the selected concept such as label and the corresponding URI can read into an external application via a web service interface. ONKI can in this way be used as a service for accurate semantic indexing.

4 Implementation

The portals discussed are based on the semantic portal tool OntoViews [3], and share the same implementation of semantic autocomplete. In the implementation, the user interface component is a shallow HTML/JavaScript wrapper, whose only responsibility is to forward typed keypresses to the server. In MUSEUMFINLAND the user interface elements are static HTML, but all the newer on-the-fly implementations make use of Ajax (Asynchronous JavaScript and XML) and the XMLHttpRequest-object¹² technologies to make HTTP queries to the server in the background while viewing a page. Depending on the complexity of the user interface, the returned content is either simple HTML to be added to the page, or JavaScript code to be executed in the context of the page.

In OntoViews, all the actual keyword matching is done on the server by Ontogator [12], the view-based search engine of OntoViews. This gives the benefit of tight integration with the main multi-facet search facilities of the engine. The search is accomplished as follows:

¹² See e.g. <http://en.wikipedia.org/wiki/AJAX>

Firstly, the complex ontological mapping, navigation and processing associated with semantic autocompletion is accomplished as a precalculation, alongside the view projection for the multi-facet search. For each category to be projected, a set of logic rules expressed in Prolog is consulted that dictate which labels of which ontological entities are to be associated with that category. By using such rules, the ontology manipulation involved is abstracted into chunks that are quite general, as well as easy to understand, combine and implement. For example, the Veturi system includes the following rules:

```
annotation(Category,Value):- rdf(Category,'rdfs:comment',Value).
```

```
annotation(Category,Value) :-
    sumoclass(Category), rdfs_subclassof(Category,SubCategory),
    not_projected(SubCategory), annotation(SubCategory,Value).
```

The first rule states that for all classes, also their `rdfs:comment` should be indexed for keyword search. The second rule then states that for each class to be projected, any annotations of subclasses *not* projected will be added. In Veturi, these two rules result in adding to the quite abstract descriptions Suggested Upper Merged Ontology (SUMO) classes used, more concrete descriptions from the mid level ontology MILO that provides example subclasses for the SUMO concepts.

At runtime, the system does only very limited processing, mostly just character manipulation of the query string, such as expanding T9-type ambiguous numerical queries [1,2] to their possible extensions. Done this way, semantic autocompletion can easily be combined with other advances in predictive text autocompletion, because the ontological navigation happens completely separately from any string matching, similarly to the approach described in [13].

5 Discussion

This paper introduced the idea of semantic autocompletion as a natural extension to traditional autocompletion based on string matching. The idea is to use semantic structures for completing user text input into semantically relevant choices based on the underlying ontologies and content. Several forms of semantic autocompletion were proposed using equivalence relations, indirect semantic relations, semantic roles, and the idea extends seamlessly into semantic search. Semantic autocompletion uses not only string matching but also logical reasoning based on the underlying ontological structures. From the end-users viewpoint the matching occurs on the semantic level. The input text and completed choice labels may be quite different, but their relation to the query can still be understood and useful.

Our implementations and practical application of the idea to multi-facet search in semantic portals suggest that semantic autocompletion should be of practical value on the semantic web. Comprehensive user testing of the approach has not been done yet. However, the intuition obtained in implementing and expanding the view-based user interfaces to support semantic autocompletion point

to good results. Combining keyword searching to the visualization capabilities of the facet hierarchies gives the user a quick path into the system, and gives at the same time an overview of what kind of information there is in the vocabulary. This guides the user in formulating the query in terms of appropriate concepts. Furthermore, showing hits inside the hierarchies solves the problems of homonymous query terms: the right meaning can be disambiguated by the view context.

Dealing with large and deep hierarchies is a major bottleneck of the multi-facet search paradigm. According to user tests [14], keyword search is usually preferred over multi-facet search if the user is capable of expressing her information need terms of accurate keywords. Semantic autocompletion makes it easier to the end-user to deal the wealth of categories used in facets. The value of semantic autocompletion here comes from the integration of the benefits of the keyword-based and multi-facet search paradigms.

Acknowledgements

Samppa Saarela was responsible for creating the server-side search engine used in the OntoViews framework, and Teppo Käsälä integrated autocompletion on-the-fly with Orava. Semantic autocompletion for the ONKI ontology server was implemented by Ville Komulainen.

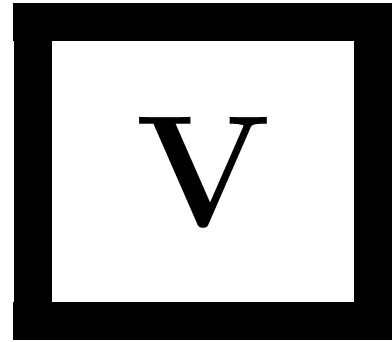
Our work is a part the “National Semantic Web Ontology Project in Finland (FinnONTO)¹³”, funded by the National Funding Agency for Technology and Innovation (Tekes) and a consortium of some 30 public organizations and companies.

References

1. Dunlop, M.D., Crossan, A.: Predictive text entry methods for mobile phones. *Personal Technologies* **4** (2000)
2. Hasselgren, J., Montnemery, E., Nugues, P., Svensson, M.: Hms: A predictive text entry method using bigrams. In: *Proceedings of the Workshop on Language Modeling for Text Entry Methods*, 10th Conference of the European Chapter of the Association of Computational Linguistics, Budapest, Hungary, Association for Computational Linguistics (2003) 43–49
3. Mäkelä, E., Hyvönen, E., Saarela, S., Viljanen, K.: Ontoviews—a tool for creating semantic web portals. In: *Proceedings of the 3rd International Semantic Web Conference (ISWC 2004)*, Hiroshima, Japan, Springer-Verlag, Berlin (2004) 797–811
4. Pollitt, A.S.: The key role of classification and indexing in view-based searching. Technical report, University of Huddersfield, UK (1998) <http://www.ifa.org/IV/ifa63/63polst.pdf>
5. Hearst, M., Elliott, A., English, J., Sinha, R., Swearingen, K., Lee, K.P.: Finding the flow in web site search. *CACM* **45** (2002) 42–49

¹³ <http://www.seco.tkk.fi/projects/finnonto/>

6. Hyvönen, E., Saarela, S., Viljanen, K.: Application of ontology-based techniques to view-based semantic search and browsing. In: *The semantic web: research and applications*. First European Semantic Web Symposium, ESWS 2004, Heraklion, Greece, Springer-Verlag, Berlin (2004) 92–106.
7. Hyvönen, E., Mäkelä, E., Salminen, M., Valo, A., Viljanen, K., Saarela, S., Junnila, M., Kettula, S.: *MuseumFinland—Finnish Museums on the Semantic Web*. *Journal of Web Semantics* **3** (2005)
8. Hyvönen, E., Junnila, M., Kettula, S., Mäkelä, E., Saarela, S., Salminen, M., Syreeni, A., Valo, A., Viljanen, K.: *Finnish Museums on the Semantic Web. User's perspective on MuseumFinland*. In: *Proceedings of Museums and the Web 2004 (MW2004)*, Selected Papers, Arlington, Virginia, USA. (2004) <http://www.archimuse.com/mw2004/papers/hyvonen/hyvonen.html>.
9. Kännsälä, T., Hyvönen, E.: *A semantic view-based portal utilizing Learning Object Metadata*. Paper, submitted, <http://www.seco.hut.fi/publications/2006/kansala-hyvonen-2006-semantic-portal-lom.pdf> (2006)
10. Mäkelä, E., Viljanen, K., Lindgren, P., Laukkanen, M., Hyvönen, E.: *Semantic yellow page service discovery: The veturi portal*. In: *Proceedings of the 4rd International Semantic Web Conference (ISWC 2005)*, Poster papers, Galway, Ireland. (2005)
11. Valo, A., Hyvönen, E., Komulainen, V.: *A collaborative ontology development and service framework ONKI*. In: *Proceedings of Int. Conf. on Dublin Core and Metadata Application (DC-2005)*, Madrid. (2005)
12. Mäkelä, E., Hyvönen, E., Saarela, S.: *Ontogator—a semantic view-based search engine service for web applications*. Paper, submitted, <http://www.seco.hut.fi/publications/2006/makela-hyvonen-saarela-ontogator-2006.pdf> (2006)
13. Legrand, S., Tyrväinen, P., Saarikoski, H.: *Bridging the word disambiguation gap with the help of OWL and semantic web ontologies*. In: *Proceedings of the Workshop on Ontologies and Information Extraction, Europlan 2003*. (2003) 29–35
14. English, J., Hearst, M., Sinha, R., Swearingen, K., Lee, K.P.: *Flexible search and navigation using faceted metadata*. Technical report, University of Berkeley, School of Information Management and Systems (2003)



Publication V

Eetu Mäkelä, Eero Hyvönen, and Samppa Saarela. 2006. Ontogator - A Semantic View-Based Search Engine Service for Web Applications. In: Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold, and Lora Aroyo (editors), *The Semantic Web - ISWC 2006*, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings, volume 4273 of *Lecture Notes in Computer Science*, pages 847–860. Springer. ISBN 3-540-49029-9.

© 2006 Springer Verlag Berlin Heidelberg

Ontogator — A Semantic View-Based Search Engine Service for Web Applications

Eetu Mäkelä¹, Eero Hyvönen¹, and Samppa Saarela^{1,2}

¹ Semantic Computing Research Group (SeCo),
Helsinki University of Technology (TKK), Laboratory of Media Technology
University of Helsinki, Department of Computer Science

`firstname.lastname@tkk.fi`

`http://www.seco.tkk.fi/`

² Mysema Ltd

`samppa.saarela@mysema.com`

Abstract. View-based search provides a promising paradigm for formulating complex semantic queries and representing results on the Semantic Web. A challenge for the application of the paradigm is the complexity of providing view-based search services through application programming interfaces (API) and web services. This paper presents a solution on how semantic view-based search can be provided efficiently through an API or as web service to external applications. The approach has been implemented as the open source tool Ontogator, that has been applied successfully in several practical semantic portals on the web.

Keywords: semantic view-based search, view projection, Semantic Web middleware.

1 Interfacing Search Services

The Semantic Web enables querying data based on various combinations of semantic relationships. Because of the RDF data model, these queries are usually drafted as possibly complex sets of semantic relation patterns. An example would be “Find all toys manufactured in Europe in the 19th century, used by someone born in the 20th century”. Here “toys”, “Europe”, “the 18th century”, “someone” and “the 19th century” are ontological class restrictions on nodes and “manufactured in”, “used by” and “time of birth” are the required connecting arcs in the pattern. While such queries are easy to formalize and query as graph patterns, they remain problematic because they are not easy for users to formulate. Therefore, much of the research in complex semantic queries has been on user interfaces [1,2] for creating complex query patterns as intuitively as possible.

View-based search [3,4] is a search interface paradigm based on a long-running library tradition of faceted classification [5]. Usability studies done on view-based search systems, such as Flamenco [6,4] and Relation Browser++ [7] have proved the paradigm both powerful and intuitive for end-users, particularly in drafting complex queries. Thus, view-based search presents a promising direction

for semantic search interface design, if it can be successfully combined with Semantic Web technologies.

The core idea of view-based search is to provide multiple, simultaneous views to an information collection, each showing the collection categorized according to some distinct, orthogonal aspect. A search in the system then proceeds by selecting subsets of values from the views, constraining the search based on the aspects selected. As an example, figure 1 shows the view-based search interface of the Veturi [8] yellow pages service discovery portal. Here, the user is looking for sweets, and has specified “marmalade”, “buy” and “Helsinki” as the Patient (Mitä), Process (Prosessi) and Place (Paikka) aspects of the service, respectively.

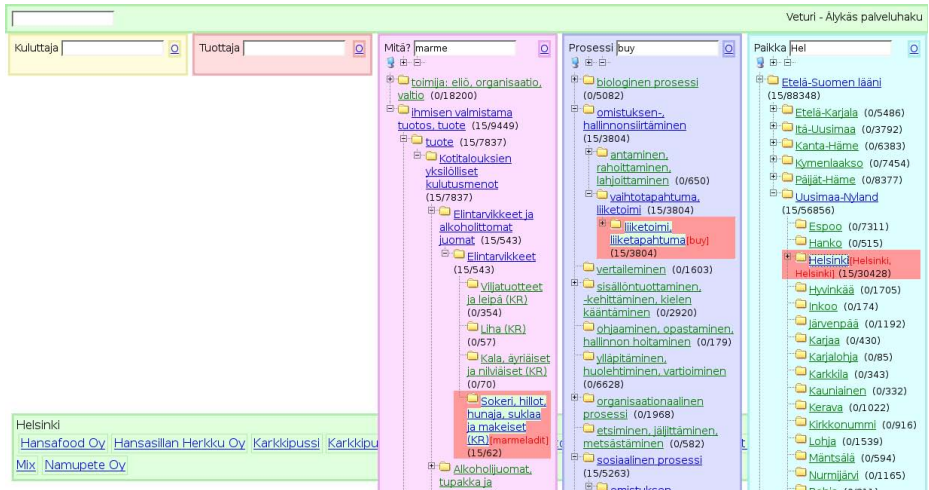


Fig. 1. Locating shops that sell marmalade in Helsinki

A key feature that differentiates view-based search from traditional keyword and Boolean search is the use of a preselected group of categorizing views in both formulating queries and in representing the results. The views give the user the query vocabulary and content classification scheme in an intuitive format. In addition, at each step, the number of hits belonging to each category is shown. Because the search proceeds by selecting these categories as further constraints, the user always knows beforehand exactly how many items will be in the result set after her next move. This prevents the user from making selections that lead to empty or very large result sets, and guides her effectively in constraining the search.

View-based search has been integrated with the Semantic Web in [9,10,11]. In this *semantic view-based search*, the facets are constructed algorithmically from a set of underlying ontologies that are used as the basis for annotating search items. Furthermore, the mapping of search items to search facets is defined using logic rules. This facilitates more intelligent search of indirectly related

items. Another benefit is that the logic layer of rules make it possible to use the same search engine for content of different kinds and annotated using different annotation schemes.

As part of the work, five view-based semantic portals were created. Previous research on the interfaces of the portals [10,11,12] have proved that regarding interface flexibility and extensibility with other semantic techniques, the view-based paradigm provides a versatile base for search on the Semantic Web. The functionalities of the interfaces developed span the whole range of search tasks identified in recent search behavior research[13,14].

Underlying all these portals is the semantic portal tool OntoViews [15], available open source under the MIT license¹. The tool is based on the Service Oriented Architecture (SOA) approach, combining independent Semantic Web Services into a working whole. This article presents the most important of these services: the general semantic view-based search service Ontogator.

Ontogator presents a solution to the following problem: what kind of search engine service and Application Programming Interface (API) are needed for supporting a variety of semantic view-based search interfaces? For a traditional Boolean logic or keyword based search engine such as Google, the API is fairly simple². The functionalities needed of a general view-based search API are much more complex. It should support facet visualization, including hit counting, facet selection, and result visualization in different ways in addition to the search logic.

Ontogator is a service with an XML/RDF-based API that provides an external software agent with all the services needed for performing view-based search. The system with its query language and implementation is described in detail in [16]. In the following, we focus in more detail on the design principles underlying the system, and the issues faced in general while designing and implementing semantic view-based search as an independent, general service.

2 Requirements for a View-Based Search API

Below are listed some services needed from the engine in a view-based semantic portal, such as MuseumFinland [11], for providing the user with a useful view-based user interface (UI).

1. Facets are exposed to the end-user in the UI for making category selections. Therefore, querying facets with hit counts projected on categories is needed.
2. On the view-based UI, clicking on a category link in a facet activates view-based search. The API therefore supports querying by Boolean category search with term expansion along facets, i.e., basic view-based search.
3. Depending on the situation, some metadata of the RDF repository, such as confidential information, should be filtered and consequently not be shown on the UI. Therefore, a mechanism for specifying the form and content of the results is useful.

¹ <http://www.seco.tkk.fi/projects/semweb/dist.php>

² see e.g. <http://www.google.com/apis/reference.html#searchrequest>

4. Reclassifying the result set along different facets and depths is needed when inspecting the hit list. In MuseumFinland, for example, the UI provides the user a link button for each view facet. By clicking it the museum collection artifacts in the hit result set are reclassified along the selected facet, such as Artifact type, Material type, Place of Manufacture, etc. A query mechanism for this is needed.
5. Combining traditional keyword search with view-based search. Research has shown [6,4] that keyword search and view-based search complement each other. In practice, both search paradigms have to be supported simultaneously, and a method for combining the paradigms is needed.
6. Support for knowledge-based semantic search. The search should be intelligent in the sense that the engine can find, using domain knowledge, also content that is only implicitly related with search categories. For example, the underlying knowledge base of MuseumFinland has some 300 rules of common knowledge that tell how artifacts are related to other concepts. If a rule tells that doctor's hats are used in academic ceremonial events, then a search with the category "Ceremonies" in the "Events" facet should retrieve all doctor's hats even when the actual metadata of the hats in the underlying databases does not directly mention ceremonies.

Generalizing these requirements and adding architectural constraints, in the end the following design goals for the system were set:

1. Adaptability and domain independence. Ontogator should easily adapt to variant domains and make use of the semantics of any data.
2. Standards. The query and response interfaces of Ontogator should conform to established Semantic Web standards as independent semantic components.
3. Extensibility. The system architecture should be extensible, especially with regard to querying functionality.
4. Scalability. The system should scale to handle large amounts of semantic metadata (millions of search items).

The challenge in designing the Ontogator search service was to find out how to support these various needs of semantic view-based search in a computationally scalable way. During design, it also became apparent that on the Semantic Web, view category identification poses certain questions in itself. In the following, these points will be discussed in their own sections.

3 Adaptability to Different Domains

A major issue in applying the view-based search paradigm is in how to create the views used in the application as flexibly as possible. On the Semantic Web, domains are described richly using ontologies. However, as in traditional classification systems, hierarchical hyponymy and meronymy relationships are still important for structuring a domain. Therefore, these ontologies typically contain

a rich variety of such elements, most often defined with explicit relations, such as “part-of” and “subclass-of”. This naturally leads to the idea of using these hierarchical structures as bases for views in view-based searching. To carry this out, Ontogator introduces a preprocessing phase termed *view projection*.

The transformation consists of two important parts: projecting a view tree from the RDF graph, and linking items to the categories projected. Originally, these tasks were performed by the Ontodella logic server [17], but recently have been incorporated into Ontogator itself. For both tasks, Ontogator relies on traversing the RDF graph guided by specified rules, picking up relevant concepts and linking them into a view tree based on the relations they have in the underlying knowledge base. The result of this phase is a set of indexed facet structures linked with the actual content items to be searched for. The domain dependent reasoning part of search is performed at this phase and means in practice mapping search items to the search categories.

For describing the view projections, Ontogator uses an RDF-based configuration format. The projection interface was designed to be modular and extensible, so that new projection rule styles and constructs could be created and used interchangeably in the system. Currently, the interface supports rules defined in a simple RDF path language, as well as the Prova³ language, a Java version of Prolog. This makes it possible to keep simple rule definitions simple, but also, if needed, take advantage of the expression power of Prolog.

As an example of the configuration format, a snippet from the Veturi portal, slightly adapted for demonstration purposes, is provided:

```
<ogt:HierarchyDefinition rdf:nodeID="patient">
  <ogt:root rdf:resource="%object;Object"/>
  <ogt:incProperty rdf:resource="%&rdfs;label"/>
  <ogt:subCategoryLink>
    <ogt:ProvaLink rdf:nodeID="coicopSubClasses">
      <ogt:isLeaf>false</ogt:isLeaf>
      <ogt:linkRule>
        rdf(Target, 'coicop:hasParent', Source).
      </ogt:linkRule>
    </ogt:ProvaLink>
  </ogt:subCategoryLink>
  <ogt:subCategoryLink rdf:nodeID="sumoSubClasses"/>
  <ogt:itemLink rdf:nodeID="sumoItems"/>
</ogt:HierarchyDefinition>
```

In the example, in the tree projection phase a “Patient” hierarchy is projected, using two “subCategoryLink” rules for recursively adding subcategories to the view. The first is a simple Prova rule for the COICOP [18] product hierarchy. The second subcategory rule for projecting the Suggested Upper Merged Ontology (SUMO) [19] -based process hierarchy is not actually defined here, but refers to a Prova definition elsewhere in the RDF document. This possibility for rule reuse is a nice property of the RDF model. As an example of a more complex rule, consider the actual definition of the linked rule:

```
% base case, handle categories where we're not told to stop, nor to skip
sumo_sub_category(Source,Target) :-
```

³ <http://www.prova.ws/>

```

Skip = 'http://www.cs.helsinki.fi/group/iwebs/ns/process.owl#skip',
rdf(Target,'rdfs:subClassOf', Source),
not(rdf(Target,'sumo_ui:display',Skip)),
not(sumo_subcategory_not_acceptable(Target)).

% if we're told to skip a category, then do it.
sumo_sub_category(Source,Target) :-
  Skip = 'http://www.cs.helsinki.fi/group/iwebs/ns/process.owl#skip',
  rdf(SubClass,'rdfs:subClassOf', Source),
  rdf(SubClass,'sumo_ui:display', Skip ),
  sumo_sub_category(SubClass,Target).

% don't process MILO categories
sumo_subcategory_not_acceptable(SubClass) :-
  Milo = 'http://reliant.teknowledge.com/DAML/MILO.owl#',
  not(rdf_split_url(Milo,Prop,SubClass)).

% don't process if we're told to stop
sumo_subcategory_not_acceptable(SubClass) :-
  Stop = 'http://www.cs.helsinki.fi/group/iwebs/ns/process.owl#stop',
  rdf( SubClass, 'sumo_ui:display', Stop).

% don't process if someone above us told us to stop
sumo_subcategory_not_acceptable(SubClass) :-
  Stop = 'http://www.cs.helsinki.fi/group/iwebs/ns/process.owl#stop',
  rdf( Y, 'sumo_ui:display', Stop ),
  not( rdf_transitive(SubClass,'rdfs:subClassOf',Y)).

```

Here, while the basis for hierarchy formulation is still the “rdfs:subClassOf” relationship, complexity arises because it is not used as-is. The class hierarchy of the SUMO ontology is designed mainly to support computerized inference, and is not necessarily intuitive to a human end user. To make the hierarchy less off-putting for a user, two additional rules are used, based on configuration information encoded directly into the RDF data model. First, categories in the middle of the tree that make sense ontologically but not to the user should be skipped, bumping subcategories up one level. Second, sometimes whole subtrees should be eliminated. In addition, in the data model there are also classes of the Mid Level Ontology (MILO) [20] extending the SUMO tree. These are used elsewhere to add textual material to the categories for text-based matching, but are not to be directly processed into the tree.

From an algorithmical perspective, in projecting a tree from a directed graph, there are always two things that must be considered. First, possible loops in the source data must be dealt with to produce a Directed Acyclic Graph (DAG). This usually means just dismissing arcs that would form cycles in the projection process. Second, classes with multiple superclasses must be dealt with to project the DAG into a tree. Usually such classes are either assigned to a single superclass or cloned, which results in cloning also the whole subtree below.

The second phase of view projection is associating the actual information items searched for with the categories. Most often, this is just a simple case of selecting a property that links the items to the categories, but it can get more complex than that here, too. Back in the first listing, the third link rule is an “itemLink”, referring to the following rule:

```

<ogt:RDFPathLink rdf:nodeID="sumoItems">
  <ogt:isLeaf>true</ogt:isLeaf>
  <ogt:linkRule>

```

```
    ^sumo:patient^process:subProcess
  </ogt:linkRule>
</ogt:RDFPathLink>
```

This rule is again defined using the simple RDF path format. The backwards path in the example specifies that to locate the service processes associated with a category of objects, one should first locate all processes where the category is specified as the patient type. From there, one can then find the services that contain those subprocesses.

The reason for introducing the projection preprocessing phase is two-fold. First, in this way the Ontogator search engine can be made completely independent of the domain knowledge and of the annotation schema used. It does not know anything about the domain semantics of the original knowledge base or the annotation schema used, but only about semantics of view-based search. Second, during knowledge compilation, efficient indices facilitating computationally scalable semantic view-based search to millions of search items can be created. A problem of the preprocessing approach is that the contents cannot, at least in the current implementation, be updated gradually.

The extensibility of the Ontogator projection architecture is based on combining only a few well defined component roles to create more complex structures. There are in essence only two types of components in the architecture: those linking individual resources to each other, and those producing resource trees. Based on these roles it is easy to reuse components, for example using the same linkers both for item and subcategory links, or creating a compound hierarchy by including individual hierarchies. Using the RDF data model for configuring the projection further supports this, giving a clear format for expressing these combinatory structures, and even making it possible to refer to and reuse common component instances.

4 Category Identification

Because of the projection, categories in semantic view-based search cannot be identified by the URIs of the original resources. First, the same resources may feature in multiple views, such as when a place is used in both a “Place of Use” and a “Place of Manufacture” view. Second, even inside one view, breaking multiple inheritance may result in cloning resources. Therefore, some method for generating category identifiers is needed.

An important consideration in this is how persistent the created identifiers need to be. In a web application for example, it is often useful for identifiers to stay the same as long as possible, to allow the user to long-term bookmark their search state in their browser. A simple approach for generating persistent category identifiers would start by just concatenating the URIs of categories in the full path from the tree root to the current category to account for multiple inheritance. Then an additional URI would have to be added, for differentiating between the semantic sense by which the actual information items are related to the categories, e.g. “Place of Use” and “Place of Manufacture” again.

This will create identifiers resilient to all changes in the underlying ontology knowledge base other than adding or moving categories in the middle of an existing hierarchy. And even in that case, good heuristics would be available for relocating lost categories. This will, however, result in very long category identifiers.

If persistence is not critical, many schemes can be applied to generate shorter category identifiers. In Ontogator, a prefix labeling scheme [21] based on subcategory relationships is used: the subcategories of *a* will be identified as *aa*, *ab* and so on. This scheme was selected because it makes finding out the subcategories of a given category very easy, a useful property in result set calculation, described later. The potential problem here is that even if the order in which subcategories are projected is preserved, adding resources to, or removing them from the ontology may result in categories with different identifiers. That is, a category with the identifier *aba* that used to represent e.g. “Finland” could turn out to represent “Norway”, with no means for the system to know about the change. As the original portals created on top of OntoViews⁴ were fairly static, this was not judged to be a problem outweighing the benefits.

5 Standards: Interfacing with Other Semantic Components

On the Semantic Web, it is important that the interfaces of programs conform to established standards, particularly for semantic services intended to be of general use. To this end, both the queries and results of Ontogator are expressed in RDF. The query interface is defined as an OWL ontology⁴, and is therefore immediately usable by any application capable of producing either RDF, or XML conforming to the RDF/XML serialization.

As for conforming to different functional needs, the interface itself then contains plenty of options to filter, group, cut, annotate and otherwise modify the results returned. These options allow the basic interface to efficiently meet different demands, as evidenced by the wide variety of interfaces[11,8,12] created using the system. For example, when constructing a view-based query for an UI page depicting the facets, one can specify that only the facet structure with hit counts but without the actual hits is returned. On a hit list page the attributes can be selected so that the actual hits are returned classified along the direct subcategories of an arbitrary facet category.

Because Ontogator mainly works with tree hierarchies inherent in ontologies, it is only natural that also the result of the search engine is expressed as an RDF tree. This tree structure also conforms to a fixed XML-structure. This is done to allow the use of XML tools such as XSLT to process the results. This provides both a fall-back to well established technologies, and allows for the use of tools especially designed to process hierarchical document structures. In OntoViews, for example, the XML/RDF results of Ontogator are transformed into XHTML UI pages by using XSLT.

⁴ <http://www.cs.helsinki.fi/group/seco/ns/2004/03/ontogator#>

The need for defining a new kind of tree-based query language, and not using existing query schemes for relational databases, XML, or RDF is due to the nature of the view-based search and to reasons of computational efficiency. In view-based search, the UI is heavily based of tree structures exposing to the end-user versatile information about the search categories and results. Supporting the creation of such structures by a search engine makes application development easier. The search and result construction is also more efficient this way. Firstly, the needed structures can be constructed at the time of the search where the information needed is easily available. Secondly, in this way the indices and search algorithms can be optimized for view-based search in particular. In our first implementation tests, some generic Semantic Web tools such as Jena were used for implementing the search operations, but in the end, special purpose Java programs were developed leading to a much more efficient implementation.

6 Extensibility

The RDF-based query language created for Ontogator was designed to be as flexible and extensible as possible also with regard to querying functionality. The basic query format is based on two components: an items clause for selecting items for the result set, and a categories clause for selecting a subtree of categories to be used in grouping the results for presentation. This format enables flexibly grouping the results using any category clause, for example organizing items based on a keyword query according to geolocations near the user.

The way both clauses work is based on an extensible set of selectors, components that produce a list of matching resource identifiers based on some criteria particular to them. The current implementation allows searching for view categories using 1) the category identifier, 2) the resource URI of which the category is projected and 3) a keyword, possibly targeted at a specific property value of the category. These category selectors can also be used also to select items. In this case the selector selects all items that relate to the found categories. Items can additionally directly be queried using their own keyword and URI selectors. Different selectors can be combined to form more complex queries using special union and intersection selectors.

Ontogator can be extended by defining and implementing new selectors. This provides a lot of freedom, as the only requirement for a selector is that it produce a list of matching items. The selector itself can implement its functionality in any way desired. For example, a selector selecting items based on location could act as a mere proxy, relaying the request to a GIS server using the user's current location as a parameter and returning results directly for further processing.

7 Scalability

The full vision of the Semantic Web requires search engines to be able to process large amounts of data. Therefore, the scalability of the system was an important consideration in the design of Ontogator. With testing on fabricated data, it was

deduced that in general, Ontogator performance degrades linearly with respect to both increasing the average number of items related to a category and increasing the amount of categories as a whole, with the amount of items in isolation not having much effect. As for real-world performance, table 1 lists the results of search performance tests done on the major portals developed. Because the queries used in the different portals differ in complexity, the results do not scale directly with regard to size, but still approximately conform to the results of the earlier tests.

Table 1. Ontogator performance comparison

Portal	Views	Categories	Items	Avg. items / category	Avg. response time
dmoz.org test	21	275,707	2,300,000	8.91	3.50 seconds
Veturi	5	2,637	196,166	128.80	2.70 seconds
MuseumFinland	9	7,637	4,132	5.10	0.22 seconds
SW-Suomi.fi	6	229	152	3.55	0.10 seconds
Orava	5	139	2,142	84.00	0.06 seconds

Of the performance test results, the ones done on the dmoz.org Open Directory Project website catalog data provide an obvious comparison point with current web portals, and confirm that this implementation of view-based search is sufficiently scalable for even large amounts of real life data. This scalability in Ontogator has been achieved using a fast memory-resident prefix label indexing scheme [21], as well as query options restricting result size and necessary processing complexity. These considerations taken are detailed below:

7.1 Indexing

The tree hierarchy -based search as presented here requires that related to a category, direct subcategories, directly linked items, the transitive closure of linked items and the path to the tree root can be computed efficiently. The reverse relation of mapping an item to all categories it belongs to also needs to be efficiently calculated.

Ontogator uses custom Java objects (in memory) to model the direct relations of categories and items. All other data related to the categories and items, such as labels or descriptions are retrieved from an associated Jena⁵ RDF model.

Both direct subcategories and directly linked items are recorded in memory for each category to allow for speedy retrieval. A full closure of linked items is not recorded, but calculated at runtime. To do this, Ontogator makes use of a subcategory closure, gathering together all items in all the found subcategories. The subcategory closure itself is acquired efficiently by making use of the prefix labeling scheme used for the categories. After generation, the labels are stored in a lexically sorted index, so that the subcategories of any given category are

⁵ <http://jena.sourceforge.net/>, the leading Java RDF toolkit, developed under an open source licence at HP labs

placed immediately after it in the index. This way, any subcategory closure can be listed in $O(\log(n) + n)$ time, by enumerating all categories in the index after the queried resource, until a prefix not matching the current resource is found. The use of prefix labeling also means that the whole path from view root to a given category is directly recorded in its label. Another advantage is that the identifiers are short, and easy to handle using standard Java utility classes.

7.2 Result Complexity Management

To decrease result file size as well as result computation complexity, Ontogator provides many options to turn off various result components. If grouping is not wanted, inclusion of categories can be turned off and respectively if items are not desired, their inclusion can be turned off. Turning both off can be used to gain metadata of the query's results, such as number of item or category hits.

The most important of these options, with regards to query efficiency, deals with the hit counts. Turning item hit counting off for categories speeds up the search by a fair amount. Used generally, however, this deprives the tree-views of their important function as categorizations of the data. Therefore, the option makes most sense in pre-queries and background queries, as well as a last effort to increase throughput when dealing with massive amounts of data.

7.3 Result Breadth Management

Result breadth options in Ontogator deal with limiting the maximum number of items or categories returned in a single query. They can either be defined globally, or to apply only to specified categories. With options to skip categories or items, this functionality can also be used for (sub)paging.

In MuseumFinland, a metadata-generating pre-query is used before the actual search query, to optimize the result breadth options used. The query results are used to specify the maximum number of items returned for each shown category — if the result contains only a few categories, more items can be fitted in each category in the user interface.

7.4 Result Depth Management

Depending on the nature of the view-based user interface, hierarchies of different depths are needed. Currently Ontogator supports three subhierarchy inclusion options. These are

- none.** No subcategories of found categories are included in the result. This option is used in category keyword queries: only categories directly matching the given keyword will be returned.
- direct.** Direct subcategories of found categories will be included in the result. This option is used to build the basic views in MuseumFinland.
- all.** The whole subhierarchy of found categories will be included in the result. This option is used to show the whole classification page in MuseumFinland, as well as the main view in Veturi, which give the user an overview of how the items are distributed in the hierarchy.

Similar options are available for controlling if and how paths to the selected category from the view root are to be returned.

With result breadth limits, these options can be used to limit the maximum size of the result set. This is especially important in limited bandwidth environments.

8 Discussion

Several lessons were learned in designing and implementing Ontogator. First, the projection formalism, particularly coupled with the expressive power of Prolog rules provide a flexible base on which to build view projection. However, Prolog is unfamiliar to many programmers. To counter this, projection configuration in Ontogator also allows defining and using simpler formalisms for cases where not so much expressive power is needed.

Second, to increase adaptability and component reuse, the old UNIX motto for creating distinct components that do one thing well, but can be connected to perform complex operations continues to apply. On the Semantic Web, it makes sense for the components to both consume and produce, as well as define their API in RDF and/or OWL.

Third, for scalable tree hierarchy-based search, an efficient index for calculating a transitive closure of items is needed, and it should be possible to curtail result calculation complexity with options. Also, problems of category identification need to be sorted out.

A limitation of the approach was also noted. Ontogator was designed as a stateless SOA service with the expectation that queries would be largely independent of each other. However, for some applications, such as the Veturi interface presented, this expectation does not hold. When navigating the tree hierarchies in Veturi, most queries are just opening further branches in a result tree that is already partially calculated. Currently, the whole visible tree needs to be recalculated and returned. A possible solution using the current architecture would be to maintain in Ontogator a cache of recently calculated result sets for reuse. This would not be a large task, as the API already uses such a cache in calculating category hit counts for the various views inside a single query.

9 Related Implementations

During the timeframe of this research, other implementations of view-based search for the Semantic Web have also surfaced. The Longwell RDF browser⁶ provides a general view-based search interface for any data. However, it supports only flat, RDF-property-based views. The SWED directory portal [22] is a semantic view hierarchy-based search portal for environmental organisations and projects. However, the view hierarchies used in the portal are not projections from full-fledged ontologies, but are manually crafted using the W3C SKOS [23]

⁶ <http://simile.mit.edu/longwell/>

schema for simple thesauri. The portal does, however, support distributed maintenance of the portal data. The Seamark Navigator⁷ by Siderean Software, Inc. is a commercial implementation of view-based semantic search. It also, however, only supports simple flat categorizations.

Acknowledgements

This research was mostly funded by the Finnish Funding Agency for Technology and Innovation Tekes.

References

1. Athanasis, N., Christophides, V., Kotzinos, D.: Generating on the fly queries for the semantic web: The ICS-FORTH graphical RQL interface (GRQL). In: Proceedings of the Third International Semantic Web Conference. (2004) 486–501
2. Catarci, T., Dongilli, P., Mascio, T.D., Franconi, E., Santucci, G., Tessaris, S.: An ontology based visual tool for query formulation support. In: Proceedings of the 16th European Conference on Artificial Intelligence, IOS Press (2004) 308–312
3. Pollitt, A.S.: The key role of classification and indexing in view-based searching. Technical report, University of Huddersfield, UK (1998)
4. 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
5. Maple, A.: Faceted access: A review of the literature. Technical report, Working Group on Faceted Access to Music, Music Library Association (1995)
6. Lee, K.P., Swearingen, K., Li, K., Hearst, M.: Faceted metadata for image search and browsing. In: Proceedings of CHI 2003, April 5–10, Fort Lauderdale, USA, Association for Computing Machinery (ACM), USA (2003)
7. Zhang, J., Marchionini, G.: Evaluation and evolution of a browse and search interface: Relation Browser++. In: dg.o2005: Proceedings of the 2005 national conference on Digital government research, Digital Government Research Center (2005) 179–188
8. Mäkelä, E., Viljanen, K., Lindgren, P., Laukkanen, M., Hyvönen, E.: Semantic yellow page service discovery: The Veturi portal. In: Poster paper, 4th International Semantic Web Conference. (2005)
9. Hyvönen, E., Saarela, S., Viljanen, K.: Application of ontology techniques to view-based semantic search and browsing. In: The Semantic Web: Research and Applications. Proceedings of the First European Semantic Web Symposium (ESWS 2004). (2004)
10. Mäkelä, E., Hyvönen, E., Sidoroff, T.: View-based user interfaces for information retrieval on the semantic web. In: Proceedings of the ISWC-2005 Workshop End User Semantic Web Interaction. (2005)
11. Hyvönen, E., Mäkelä, E., Salminen, M., Valo, A., Viljanen, K., Saarela, S., Junnila, M., Kettula, S.: MuseumFinland – Finnish museums on the semantic web. *Journal of Web Semantics* **3**(2) (2005) 25
12. Sidoroff, T., Hyvönen, E.: Semantic e-government portals - a case study. In: Proceedings of the ISWC-2005 Workshop Semantic Web Case Studies and Best Practices for eBusiness SWCASE05. (2005)

⁷ <http://siderean.com/products.html>

13. Sellen, A., Murphy, R., Shaw, K.L.: How Knowledge Workers Use the Web. In: Proceedings of the SIGCHI conference on Human factors in computing systems, CHI Letters 4(1), ACM (2002)
14. Teevan, J., Alvarado, C., Ackerman, M.S., Karger, D.R.: The perfect search engine is not enough: a study of orienteering behavior in directed search. In: Proceedings of the Conference on Human Factors in Computing Systems, CHI. (2004) 415–422
15. Mäkelä, E., Hyvönen, E., Saarela, S., Viljanen, K.: OntoViews - A Tool for Creating Semantic Web Portals. In: Proceedings of the Third International Semantic Web Conference, Springer Verlag (2004)
16. Saarela, S.: Näkymäpohjainen rdf-haku. Master's thesis, University of Helsinki (2004)
17. Viljanen, K., Käsälä, T., Hyvönen, E., Mäkelä, E.: Ontodella - a projection and linking service for semantic web applications. In: Proceedings of the 17th International Conference on Database and Expert Systems Applications (DEXA 2006), Krakow, Poland, IEEE (2006) To be published.
18. United Nations, Statistics Division: Classification of Individual Consumption by Purpose (COICOP). United Nations, New York, USA (1999)
19. Pease, A., Niles, I., Li, J.: The suggested upper merged ontology: A large ontology for the semantic web and its applications. In: Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web. (2002)
20. Niles, I., Terry, A.: The MILO: A general-purpose, mid-level ontology. In Arabnia, H.R., ed.: IKE, CSREA Press (2004) 15–19
21. Christophides, V., Karvounarakis, G., Plexousakis, D., Scholl, M., Tourtounis, S.: Optimizing taxonomic semantic web queries using labeling schemes. *Journal of Web Semantics* 1(2) (2004) 207–228
22. Reynolds, D., Shabajee, P., Cayzer, S.: Semantic Information Portals. In: Proceedings of the 13th International World Wide Web Conference on Alternate track papers & posters, ACM Press (2004)
23. Miles, A., Brickley, D., eds.: SKOS Core Guide. World Wide Web Consortium (2005) W3C Recommendation Working Draft.



VI

Publication VI

Eero Hyvönen, Tuukka Ruotsalo, Thomas Häggström, Mirva Salminen, Miikka Junnila, Mikko Virkkilä, Mikko Haaramo, Eetu Mäkelä, Tomi Kauppinen, and Kim Viljanen. 2007. CultureSampo – Finnish Culture on the Semantic Web: The Vision and First Results. In: Klaus Robering (editor), *Information Technology for the Virtual Museum – Museology and the Semantic Web*, pages 33–58. LIT Verlag, Berlin. ISBN 978-3-8258-0262-2.

© 2007 LIT Verlag Berlin-Münster-Wien-Zürich-London

CultureSampo—Finnish Culture on the Semantic Web: The Vision and First Results

Eero Hyvönen, Tuukka Ruotsalo, Thomas Häggström,
Mirva Salminen, Miikka Junnila, Mikko Virkkilä, Mikko Haaramo,
Eetu Mäkelä, Tomi Kauppinen, and Kim Viljanen

*Semantic Computing Research Group
Helsinki University of Technology (TKK), Laboratory of Media Technology
University of Helsinki, Department of Computer Science
<http://www.seco.tkk.fi/>
first.last@tkk.fi

Abstract

This paper concerns the idea of publishing heterogenous cultural content on the Semantic Web. By heterogenous content we mean metadata describing potentially any kind of cultural objects, including artifacts, photos, paintings, videos, folklore, cultural sites, cultural process descriptions, biographies, history etc. The metadata schemas used are different and the metadata may be represented at different levels of semantic granularity. This work is an extension to previous research on semantic cultural portals, such as MuseumFinland, that are usually based on a shared homogeneous schema, such as Dublin Core, and focus on content of similar kinds, such as artifacts. Our experiences suggest that a semantically richer event-based knowledge representation scheme than traditional metadata schemas is needed in order to support reasoning when performing semantic search and browsing. The new key idea is to transform different forms of metadata into event-based knowledge about the entities and events that take place in the world or in fiction. This approach facilitates semantic interoperability and reasoning about the world and stories at the same time, which enables implementation of intelligent services for the end-user. These ideas are addressed by presenting the vision and solution approaches taken in two prototype implementations of a new kind of cross-domain semantic cultural portal “CULTURESAMPO—Finnish Culture on the Semantic Web”.

1 Towards Semantic Cross-domain Interoperability

A widely shared goal of cultural institutions is to provide the general public and the researchers with aggregated views to cultural heritage, where the users are able to access the contents of several heterogenous distributed collections of institutions *simultaneously*. In this way, the organizational and technical obstacles for information retrieval between collections and organizations, even between countries and languages could be crossed.

Content aggregation may occur at the syntactic or semantic level. The basis for *syntactic interoperability* is sharing syntactic forms between different data sources, i.e., the metadata schemas such as the Dublin Core Metadata Element Set¹ or the Visual Re-

source Association’s (VRA) Core Categories². Such schemas make it possible to identify different aspects of the search objects, such as the “author”, “title”, and “subject” of a document, and focus search according to these. Syntactic interoperability facilitates, for example, multi- or metasearch³. Here the user types in a query in a metaportal. The query is then distributed to a set of underlying systems and the results are aggregated for the end-user. For example, the Australian Museums and Galleries Online⁴ and Artefacts Canada⁵ are multi-search engines over nation-wide distributed cultural collections. Here the content includes metadata about museum artifacts, publications etc. represented using shared metadata schemas.

Content aggregation at the *semantic level* means that not only the form of the data is shared and in-

¹<http://dublincore.org/documents/1998/09/dces/>

²<http://www.vraweb.org/vracore3.htm>

³http://en.wikipedia.org/wiki/Metasearch_engine

⁴<http://www.amonline.net.au/>

⁵<http://www.chin.gc.ca/>

teroperable, but also the values used in the metadata schema, and that the meanings of the values are semantically defined in terms of ontological structures. The values of metadata fields, such as authors, material types, and geographical locations are taken from a set of shared vocabularies, i.e., ontologies, or if different vocabularies are used, then the mappings between them are available. At this level of content aggregation, reasoning about the ontological relations between content items becomes possible enabling semantic search, semantic browsing, recommendations, explanations, and other “intelligent” services. A prototypical example of this approach is the portal “MUSEUMFINLAND—Finnish Museums on the Semantic Web”⁶ (Hyvönen et al., 2005a), where distributed, syntactically heterogeneous museum collection databases are integrated by a set of seven shared ontologies, and semantic search and browsing services are provided to end-users based on the aggregated knowledge base.

Another distinctive feature between cultural content aggregation systems is whether they deal with metadata that conforms to a *single metadata schema* or *multiple schemas*. For example, the Helmet library system⁷ aggregates public library collections of neighboring cities for the public by using a single metadata format. In the same vein, an online record shop may deal with CD/DVDs whose metadata is represented in a homogeneous way. On the other hand, in a system such as Artefacts Canada, the underlying databases contain items of different kinds, such as art, furniture, photos, magazines etc. whose metadata conform to different schemas. For example, a metadata field representing physical the material of an object is essential for a piece of furniture or artifact but not for a publication.

Semantic web portals have tackled the problem of semantic interoperability usually by sharing metadata schemas. For example, in MUSEUMFINLAND heterogeneous artifact collection databases were made semantically interoperable, but the content was of a single domain (artifacts), and the metadata was based on a single, Dublin core like schema of artifacts. There are paintings and some other forms of art in MuseumFinland collections, but they have been cataloged as pieces of artifacts in the cultural museums participating in the portal, and not as pieces of art. The reasoning routines were based on the annotation schema and the ontologies.

In this paper we investigate the problem of *seman-*

⁶This application is operational at <http://www.museusuomi.fi> with a tutorial in English.

⁷<http://www.helmet.fi>

tic cross-domain interoperability, i.e. how content of different kinds conforming to multiple metadata schemas could be made semantically interoperable. The focus is the cultural domain and content types studied in our work include artifacts, paintings, photographs, videos, audios, narratives (stories, biographies, epics), cultural processes (e.g., farming, booth making), cultural sites, historical events, and learning objects. In this case, the content is cross-domain in nature and, as a result, comes in forms that may be quite different from each other. Mapping them into a Dublin Core like generic metadata framework is problematic. Instead, we propose content aggregation at a semantically more foundational and rich level based on events and thematic roles (Sowa, 2000). The research is being carried out not only in theory, but by implementing real portal prototypes. More specifically, we show how the idea of MUSEUMFINLAND can be extended into a cross-domain semantic cultural portal called “CULTURESAMPO—Finnish Culture on the Semantic Web”. Figure 1 illustrates the positioning of CULTURESAMPO along the distinctions discussed above and its relation to some other portal systems mentioned.

Multi-domain	Artefacts Canada	CultureSampo
Single-domain	Helmet library system	MuseumFinland
	Syntactic interoperability	Semantic interoperability

Figure 1: Portals can be classified in terms of the number of metadata schemas used (vertical axis) and the level of interoperability (horizontal axis).

In the following we first state the vision and goals of creating CULTURESAMPO. After this problems of obtaining semantic cross-domain interoperability are discussed and the idea of using event-based descriptions is proposed as a solution. The discussion is based on experiences gathered in creating two experimental prototypes of CULTURESAMPO. In conclusion, contributions of the work are summarized and directions for further research are proposed.

2 The Vision and Goals of CultureSampo

CULTURESAMPO shares the general goals of MUSEUMFINLAND:

Global view to heterogeneous, distributed contents

The portal supports the usage of heterogeneous and distributed collections and contents of the participating organizations as if there were a single uniform repository.

Intelligent end-user services The system supports *semantic search* based on ontological concepts and *semantic browsing*, where semantic associations between search objects are exposed dynamically to the end-user as recommendation links with explicit explanations. These links are defined in terms of logical rules that make use of the underlying ontologies and collection metadata.

Shared content publication channel The portal should provide the participating organizations with a shared, cost-effective publication channel.

CULTURESAMPO focuses, from the content perspective, especially on material related to the “Golden Era” of the Finnish culture in the 19th century. During this period the notion of Finland as a nation with an original cultural background and history was formed, and the development resulted in the independence of the country in 1917.⁸ A central component of the Finnish cultural heritage has been the national epic Kalevala⁹. It was published originally in 1835 and has been translated into some 60 languages. This epic, based on large collections of folklore¹⁰ collected especially in the eastern parts of Finland, Karelia, has been a continuous source of inspiration in Finnish fine arts, music, sculpture, literature, and other branches of culture. The world of Kalevala also nicely relates to the original agrarian Finnish life and artifacts that are available in museums.

In CULTURESAMPO the Karelian culture is central also because one goal of the work is to reunite Karelian collections using semantic web techniques. These collections have now been distributed in several museums due to the result of the World War II where eastern parts of Finland were annexed to the Soviet Union. The semantic web provides means for re-uniting cultural entities virtually on the semantic web. The problem of distributed cultural heritage due to wars and other reasons is very common in Europe. We envision, that the ideas and techniques developed in CULTURESAMPO could later contribute to creation

⁸Before that Finland had been a part of Sweden (until 1809) and Russia (1809-1917).

⁹<http://www.finlit.fi/kalevala/index.php?m=163&l=2>

¹⁰<http://www.finlit.fi/english/kra/collections.htm>

of cross-national and multi-lingual cultural portals, a kind “CultureEurope”.

The system will also contribute, in a sense, to the tradition of Kaleva translations. It provides first excerpts of Kalevala translated, not into natural languages for the humans to use but for the machine to “understand” in the formal languages of the semantic web, RDF and OWL.¹¹

The latter part of the portal name “Sampo” is the name of the mythical machine-like entity of the Kalevala epic. Sampo gives its owner power, prosperity, everything, but its actual construction and nature is semantically ambiguous and remains a mystery — tens of academic theories about its meaning have been presented. CULTURESAMPO adds still another modern interpretation of what a “Sampo” could be based on the semantic web.

3 Making a Cultural Portal More Intelligent

A major focus of our work in CULTURESAMPO is to study how to provide the end-user with intelligent search and browsing services based on semantically rich cross-domain content originating from different kind of cultural institutions. For example, consider the painting “Kullervo departs for the war” in figure 2 depicting an event in Kalevala. From the end-users’ viewpoint, it could probably be interesting, if this piece of art could be linked with other paintings and photos, with the war theme in art museums, with weapons and accessories in cultural museums, with academic studies about Kalevala and Kullervo, with information about dogs and horses in the museum of natural history, with other (external) information on the web about Kullervo, with the actual poem in Kalevala and related pieces of folk poetry, with movies and videos on a media server, with biographies of the artist, and so on. An interesting line of associations could be created by considering the events, processes, and the Kalevala story that takes place in the picture. In this way, for example, the painting could be linked with content concerning the next or previous events in the Kalevala story. Such associations and viewpoints could be insightful, useful, and even entertaining both when searching for content and when browsing it.

To investigate and test the feasibility of this idea in practise, we are extending the portal MUSEUM-FINLAND into CULTURESAMPO by a sequence of new prototypes. In 2005, the first prototype to be called “CULTURESAMPO I” was designed and

¹¹<http://www.w3.org/2001/sw/>



Figure 2: Kullervo departs for the war. A painting at the Finnish National Gallery.

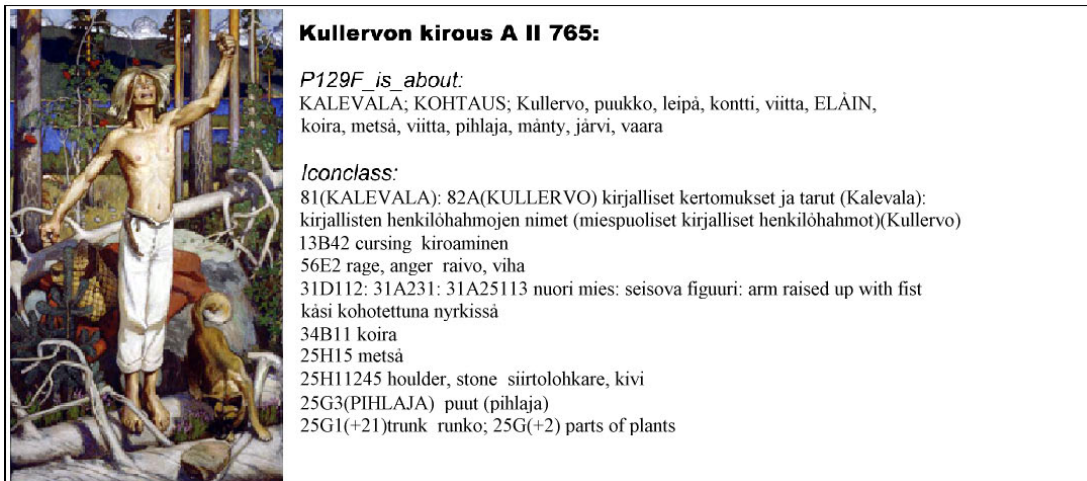


Figure 3: The painting “Kullervo cursing” and its metadata from the Finnish National Gallery.

implemented (Junnila et al., 2006; Junnila, 2006; Salminen, 2006). Figure 3 depicts a painting and its metadata in CULTURESAMPO I. The metadata shown originates from the Finnish National Gallery¹² and describes the subject of the painting in the following way: First, the CIDOC CRM¹³ (Doerr, 2003) property `P129F_is_about` lists the following set of keywords (in Finnish): “Kalevala”, “event”, “Kullervo”, “knife”, “bread”, “knapsack”, “robe”, “animal”, “dog”, “forest”, “rowan”, “pine”, “lake”, and “mountain”. Second, the property “Iconclass” lists a set of ICONCLASS¹⁴ (van den Berg, 1995) notations (categories) describing the subject. This description is partly redundant with the Finnish keywords.

In figure 4 this painting is viewed in CULTURESAMPO I. On the right column, a set of semantic links to other search objects are recommended with explanations created by the logical linking server Ontodella (Viljanen et al., 2006). The figure illustrates a link to a knapsack in the collections of the National Museum of Finland¹⁵, a link to a biography of the artist, and a link to the point in the Kalevala epic where the event of the painting actually takes place.

CULTURESAMPO I was implemented using the same framework as MUSEUMFINLAND, i.e., the OntoViews framework (Mäkelä et al., 2004) including the view-based semantic search engine Ontogator (Mäkelä et al., 2006) and Ontodella (Viljanen et al., 2006). However, in this case much richer cross-domain metadata was used. The test material was limited in size but included examples of artifacts, paintings, photos, videos, biographical information, and narratives such as poems of Kalevala, and descriptions of traditional agrarian processes, such as farming by the slash and burn method.

During this experiment we identified two major obstacles for creating cross-domain semantic cultural portals:

Semantic Interoperability of metadata schemas.

The problem of integrating metadata schemas occurs 1) *horizontally* when integrating schemas of different form semantically and 2) *vertically* when integrating content annotated at different levels of granularity.

Expressive power of metadata schemas. A central research hypotheses underlying CULTURESAMPO is that, from the end-user’s viewpoint,

different processes and events that take place in the society and history should be used as a kind semantic glue by which “insightful semantic links” could be created for the user to browse. This idea was already tested to some extent in MUSEUMFINLAND by creating an artificial event view for the end-user, and by mapping contents of it using logical rules. However, it seemed that a richer and a more accurate knowledge representation method was needed in annotating the contents than traditional metadata schemas.

In the following, our approaches to addressing these problems are outlined.

4 Semantic Interoperability of Metadata Schemas

Re-consider the figure 2. Its metadata may tell e.g. that this painting was created by A. Gallen-Kallela in 1901 in Helsinki. This metadata can be represented, by using RDF triples in Turtle notation¹⁶, in the following way (this example is not based on the actual metadata but is for illustration only):

```
:Kullervo_departs_war
  dc:creator persons:A.Gallen-Kallela ;
  dc:date "1901" ;
  dc:spatial places:Helsinki .
```

The metadata record in a biographical repository, such as the ULAN¹⁷ of the Getty Foundation, may tell us more about the artist in a very different metadata format, e.g.:

```
persons:A.Gallen-Kallela
  :placeOfBirth places:Pori ;
  :timeOfBirth "1865" ;
  :placeOfDeath places:Stockholm ;
  :timeOfDeath "1931" .
```

A problem here is that the number of different properties in metadata schemas easily gets large in cross-domain applications. Furthermore, the meaning of many properties, such as `dc:date` and `dc:spatial` in the metadata schema of paintings and `timeOfBirth/Death` and `placeOfBirth/Death` in the biographical metadata schema of persons may share some meaning, but are still different. We soon realized that when using the schemas for reasoning tasks, the logical rules accounting properly all kinds of combinations

¹²<http://www.fng.fi>

¹³<http://cidoc.ics.forth.gr/>

¹⁴<http://www.iconclass.nl>

¹⁵<http://www.nba.fi/en/nmf>

¹⁶<http://www.dajobe.org/2004/01/turtle/>

¹⁷<http://www.getty.edu/vow/ULANSearchPage.jsp>

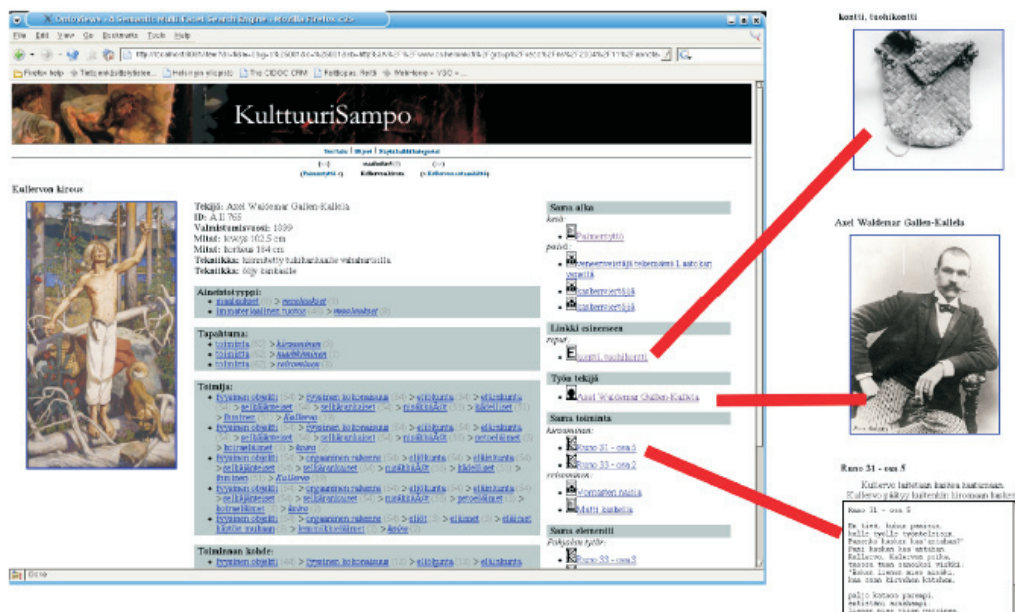


Figure 4: The painting of figure 3 viewed in the semantic portal CULTURESAMPO I. Three semantic recommendation links created by the system are visualized on top of the screenshot.

of properties become complicated, and the number of rules becomes large due to combinatorial explosion. It seems that a more primitive representation of knowledge than traditional metadata schemas is needed for reasoning.

A potential solution approach to solve the problem is to use the CIDOC CRM ontology. The system “provides definitions and a formal structure for describing the implicit and explicit concepts and relationships used in cultural heritage documentation”¹⁸. The framework includes some 80 classes, such as “E22 Man-Made Object”, “E53 Place”, and “E52 Time-Span”, and a large set of some 130 properties relating the entities with each other, such as “P4 Has Time-Span” and “P87 Is Identified By”. Interoperability of cultural content can be obtained by mapping metadata standards to CIDOC CRM.

The focus in CIDOC CRM is in modeling concepts necessary for representing the documentation semantics of different metadata schemas used in the cultural domain, such as Dublin Core. In contrast, in CULTURESAMPO our main focus is to represent *real world knowledge* related to cultural heritage, e.g., the subjects that the paintings in figures 2 and 3 depict. For this purpose, a different kind of knowledge repre-

¹⁸<http://cidoc.ics.forth.gr/>

sentation scheme and large domain ontologies containing tens of thousands of domain concepts and events are needed.

Our solution to the problem of semantic interoperability is to transform different metadata schemas into a shared, more primitive knowledge representation of the real world. In this way, the meaning of `dc:date`, `:timeOfBirth` and `:timeOfDeath` can be made interoperable. By basing reasoning on the more primitive representation, more generic and fewer rules operating a smaller set of properties can be devised. As for the knowledge representation scheme, the idea of representing knowledge in terms of actions and thematic relations between actions and entities was adopted. This general idea has been applied widely in computational linguistics and natural language processing (cf. e.g. (Zarri, 2003)), in knowledge representation research (Sowa, 2000), and also in CIDOC CRM, where events are of central importance, too.

For example, in CULTURESAMPO the three time-relations of the above examples are reduced into only one time-relation relating an instance of an event type, such as “`painting_event`”, “`birth_event`”, or “`death_event`” to a time entity. The meaning of semantically complex properties in metadata schemas

is essentially represented in terms of different events and related entities. For example, the metadata about the painting “Kullervo departs for the war” means that there was a painting event related with A. Gallen-Kallela, the year 1901, and Helsinki by the thematic roles “agent”, “time”, and “place”:

```
:painting_event_45
  rdf:type cs:painting_event ;
  cs:agent persons:A.Gallen-Kallela ;
  cs:time "1901" ;
  cs:place places:Helsinki .
```

Information about the artist’s birth and death dates can be transformed in a similar manner into birth and death events, respectively. In this way, we can not only eliminate various time-related properties from the knowledge base but also aggregate knowledge from different sources on the more primitive knowledge representation level. In this case, for example, event-based biographical knowledge about the life events of A. Gallen-Kallela can be enriched with the knowledge about the paintings he painted.

Solving the semantic interoperability problem of metadata schemas by using a primitive event-based knowledge representation scheme was one of the major challenges in creating the CULTURESAMPO II prototype in 2006. This idea will be described, especially from the semantic browsing viewpoint, in more detail in (Ruotsalo and Hyvönen, 2006).

5 Extending Semantic Representational Power of Metadata Schemas

The idea of using event-based knowledge representations in annotation provides also a solution for creating semantically richer annotations. Event-based annotations have been studied before, e.g., in the context of annotating the subject of photographs (Schreiber et al., 2001) and in representing narratives (Zarri, 2003).

To illustrate this idea, re-consider the painting “Kullervo departs for the war” of figure 2. The subject of content is here annotated by a set of keywords (in Finnish) including “Kullervo”, “horse” and “dog”. A problem from the knowledge representation viewpoint is that the mutual relations of the subject annotations are not known. For example, it is not known whether Kullervo rides a horse, a dog, both of them, or none of them. It is also possible that the dog rides Kullervo, and so on. Events can be used for elaborating the description, if needed, by specifying values

for their thematic roles. In this case, for example, Kullervo would be in the agent role and the horse in the patient role in a riding event. This kind of information can be essential when searching the contents (e.g. to distinguish between riders and riding entities) or when providing the end-user with semantic links and explanations (e.g. to distinguish links to other riding paintings in contrast to other horse paintings).

In CULTURESAMPO content comes not only in different forms but is also annotated at different levels of detail “vertically”. For example, the metadata from a museum database is given as it is and may contain only minimal metadata while some other content may be described in a very detailed manner by using lots of Iconclass notations or manually annotated events. In our case, detailed semantic descriptions are being created, for instance, when translating the Kalevala story into RDF and OWL. Here each Kalevala part of potential interest to the end-user is annotated in terms of events, thematic roles and other metadata. Furthermore, the events may constitute larger entities and have some additional semantic relations with each other. In CULTURESAMPO I this idea was experimented by representing processes and events of two Kalevala poems, in paintings, photos, and cultural processes (Junnila et al., 2006).

In CULTURESAMPO II this work continues with a new modified event-based model. Furthermore, in the new scheme, annotations can be given at three levels of granularity in order to enable vertical interoperability:

Keywords In some cases only keywords are available as subject metadata. At this level the annotation is a set of literal values. Even if ontological annotations have been used (cf. below), literal keywords may be needed for free indexing words.

Keyconcepts Here the annotation is a set of URIs or other unique references to an ontology or a classification system, such as Iconclass. The additional knowledge introduced by keyconcepts w.r.t. using literal keywords is their ontological connections. This enables semantic interoperability, as discussed earlier.

Thematic roles At this level thematic roles between activities and other entities can be specified. The additional knowledge w.r.t. using only keyconcepts is the distinction of the roles in which the keyconcepts are at different metadata descriptions.

Each new level of annotation granularity only adds

new information with respect to the previous level. This means that semantically richer representations can be easily interpreted at the lower level. Event-based descriptions mean at the keyconcept level that only the entity resources that are used in the events are considered, not the properties. At the keyword level, only the literal labels of the annotations at the keyconcept level are considered. This strategy enables, e.g., application and integration of traditional text-based search methods with ontological annotations—a useful feature since much of the content in semantic portals is in textual form in any case (e.g., free text descriptions of collection items, biographical articles, poems etc.).

The main ontology underlying CULTURESAMPO II is the General Finnish Upper Ontology YSO (Hyvönen et al., 2005b) of about 20,000 concepts. This lightweight ontology has been created based on the widely used General Finnish Thesaurus YSA¹⁹. CULTURESAMPO also makes use of extended versions of the ontologies used in MUSEUMFINLAND.

6 The Portal

CULTURESAMPO II provides the end-user with semantic search and browsing facilities in a way similar to MUSEUMFINLAND. Semantic multi-facet search can be used. Since the ontological model is event-centric, the user is provided with a view classifying verb-like event concepts in addition to more traditional views (persons, collections, etc.). Figure 5 illustrates the search interface.

When a search object is selected to viewing, recommended semantic links with explanations are provided for browsing. Also here the event-centric model is evident: most recommendations are based on sharing events and roles. Figure 6 shows a search object page of a photograph for illustration.

In addition, CULTURESAMPO II includes many new forms of semantic visualization, especially w.r.t. geographical information and time lines (Kauppinen et al., 2006). For visualizing search results on the map, Google Maps²⁰ service is used (cf. figure 7). It will be used as a search interface, too, later on. In the same vein, the Simile Time Line²¹ has been incorporated in the user interface using Ajax-technology (cf. figure 8).

CultureSampo I was implemented on our old On-toViews architecture, based on Apache Cocoon²².

¹⁹<http://www.vesa.lib.helsinki.fi>

²⁰<http://maps.google.com/>

²¹<http://simile.mit.edu/timeline/>

²²<http://cocoon.apache.org/>

However, when adding many more cross-linked components to the system in CULTURESAMPO II, such as the timeline, map views, and the new recommendation system, severe limits in the old architecture became apparent.

A major guideline in our work has been to create applications that can be configured to work with a wide variety of RDF data. To accomplish this, we have endeavored to build our applications out of modular components that combine to provide advanced functionality. As CULTURESAMPO II became more complex and started to incorporate components from other projects, there appeared a need for making the individual components smaller and supporting a more complex multidirectional control and configuration flow between them. Apache Cocoon, however, is based on a generally sequential pipeline architecture, which is very limited in its ability to support any multidirectional communication. And while it was possible to make use of modular component libraries on the Java level, there was no architectural support for keeping these components either universal or configurable, which in general resulted in them not being such.

To solve these problems, a new architecture was developed for CultureSampo II based on the well-known Service Oriented Architecture, Inversion of Control and Dependency Injection principles. Specifically, the new platform was built on top of the Apache HiveMind²³ services and configuration microkernel.

7 Discussion

Our work on CULTURESAMPO suggests that using event-based annotations is a promising approach to creating cross-domain semantic portals for several reasons:

1. By using more accurate semantic descriptions semantic search and browsing (recommendation) can be made more accurate and explained in more detail. The semantic accuracy of annotations can be extended in a natural way by the new layer of relational event annotations that explicate the thematic roles between activities and other entities in the description. First tests on CULTURESAMPO I and II seem to indicate that this kind semantic knowledge is vital for semantic information retrieval tasks (search) and for creating insightful semantic linking of contents

²³<http://jakarta.apache.org/hivemind/>

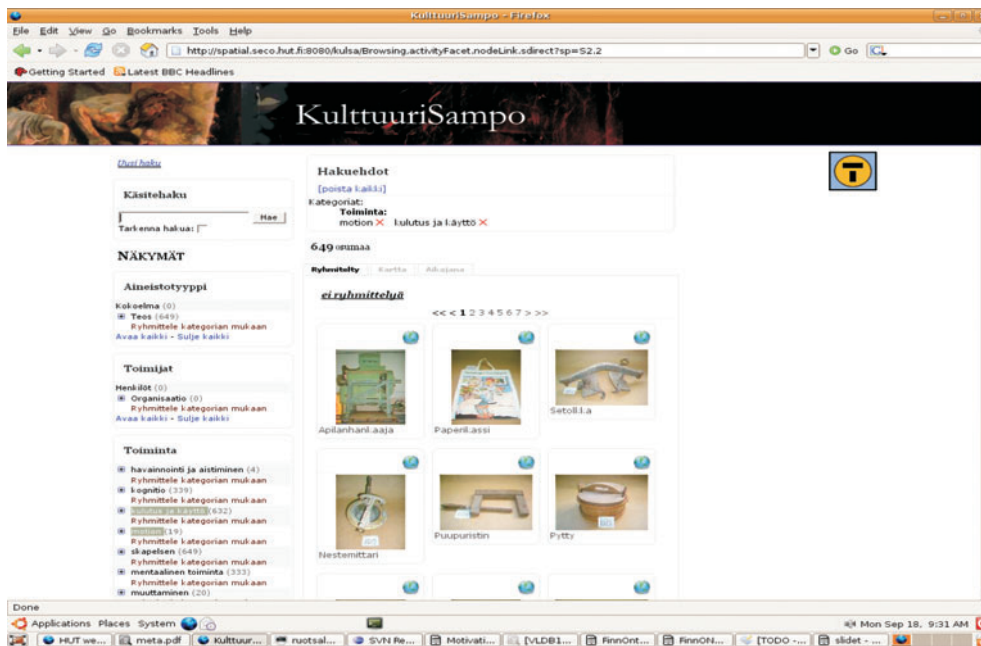


Figure 5: CULTURESAMPO II search page. Views are on left and hits on the right.

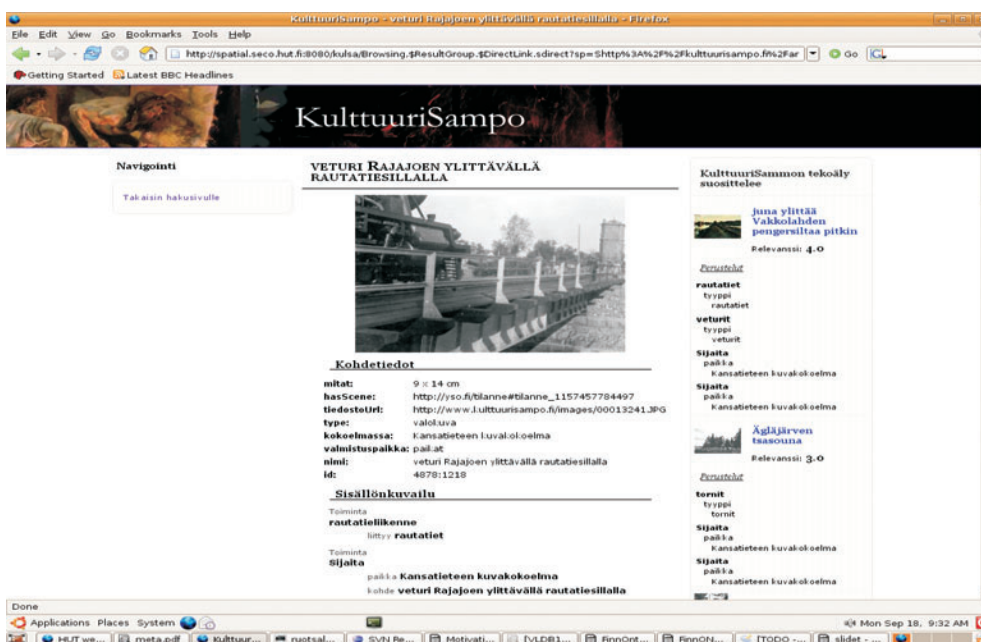


Figure 6: CULTURESAMPO II item page. Metadata is on the left and recommendation links on the right.

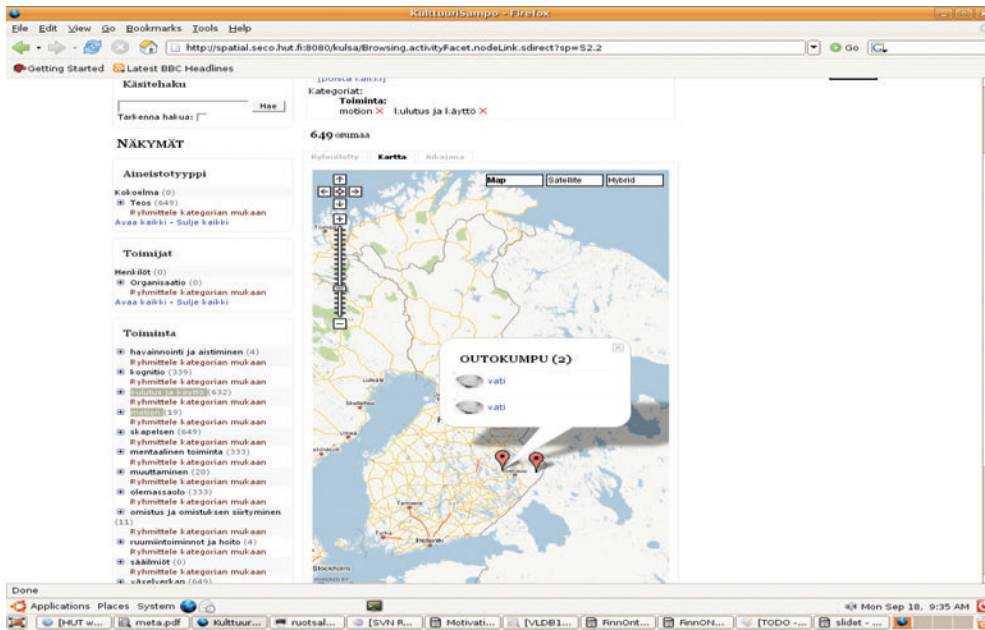


Figure 7: Using Google Maps in CULTURESAMPO II for visualizing search items on the map. The items are positioned based on a place ontology, and interactive to obtain additional information.

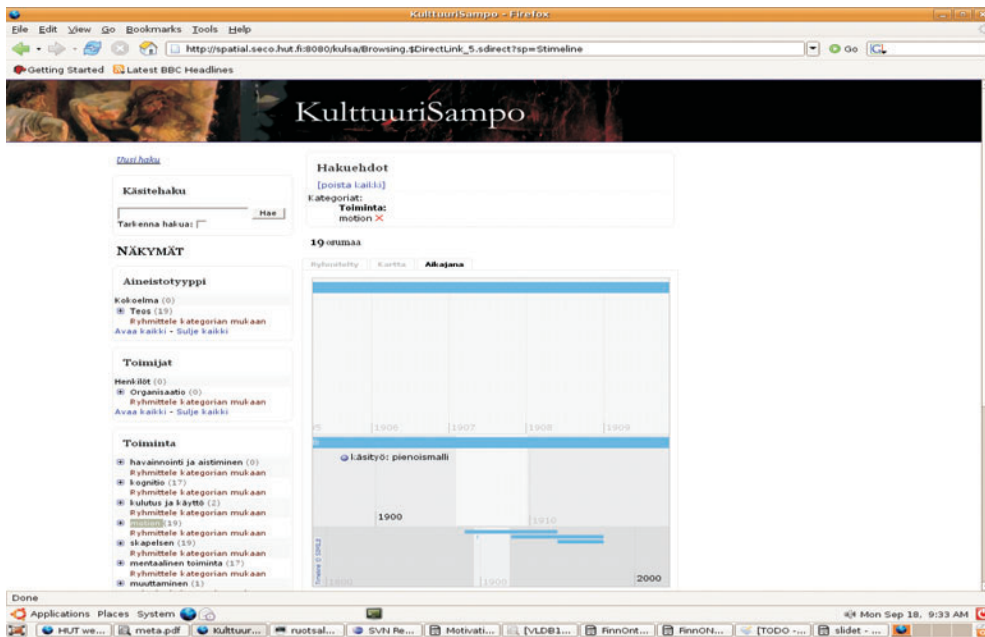


Figure 8: Using Simile Time Line in CULTURESAMPO II for visualizing search items on the time line, and for selecting them for a closer look.

automatically (Junnila et al., 2006; Ruotsalo and Hyvönen, 2006).

2. Event-based descriptions can be used for representing the meanings in terms of happenings and entities of the real world based on different metadata schemas. This enables semantic interoperability.
3. The resulting knowledge representation scheme is simpler in terms of the number of properties than the original set of metadata schemas. This makes it simpler to implement reasoning rules needed for the intelligent services for the end-user.

The price for more accurate semantics is the extra cost of creating the annotations. In CULTURESAMPO I all content was manually crafted. In CULTURESAMPO II a semi-automatic process has been used. At the schema level, the content has been enriched automatically by a separate, rule-based knowledge transformation module. This system transforms, e.g., the metadata of paintings into painting events. At the level of enriching the subject descriptions of the content, enriching has been mostly manual by adding thematic role relations between the entities used in the original databases. For example, to tell that Kullervo rides a horse and not vice versa in figure 2, a riding event with Kullervo and an instance of horse in the proper thematic roles has to be created. In principle, the machine and ontologies could help the annotator in her work, if it is known that usually humans ride horses.

The work of annotating narratives, such as the Kullervo poem in Kalevala and the process of farming by the slash and burn method in CULTURESAMPO I (Junnila et al., 2006) has been done completely manually. However, we are also investigating how language technology can be applied to creating semi-automatically annotations for textual contents (Vehviläinen et al., 2006). It is clear, that development of tools that could help in creating annotations will be of utmost importance in the future.

In some cases, like when annotating unique important materials such as Kalevala, the price for detailed annotations can be paid, while in many other cases it is unrealistic to assume that such representations will be available. In CULTURESAMPO this problem of dealing with materials annotated at different levels of semantic accuracy is addressed by using three layers of annotations: keywords, keyconcepts and thematic roles.

The success of the CULTURESAMPO will finally be judged by the end-users. Empirical usability tests

are needed in order to evaluate the added value of the semantic approach. The first test, based on the current CULTURESAMPO II, has been scheduled for the autumn 2006. The goal of this experiment is to test whether the end-users really find the semantic recommendations generated by the event-based model feasible and helpful.

CULTURESAMPO II is still a research prototype and its current version contains only a few content types and less than 10,000 search objects. For example, in contrast to CULTURESAMPO I, there are no narratives in the system yet, only events. However, new types of content are being included in the scheme and in the system. Another line of development in the system is designing additional conceptual visualization tools. On the reasoning side, spatiotemporal reasoning under uncertainty is being studied (Kauppinen and Hyvönen, 2006) and is being implemented in the system.

We plan to publish CULTURESAMPO on the public web in 2007.

Acknowledgments

Our research is a part of the National Semantic Web Ontology Project in Finland²⁴ (FinnONTO) 2003-2007 funded mainly by the Finnish Funding Agency for Technology and Innovation (Tekes).

References

- M. Doerr. The CIDOC CRM - an ontological approach to semantic interoperability of metadata. *AI Magazine*, 24(3):75–92, 2003.
- E. Hyvönen, E. Mäkela, M. Salminen, A. Valo, K. Viljanen, S. Saarela, M. Junnilla, and S. Kettula. MuseumFinland – Finnish museums on the semantic web. *Journal of Web Semantics*, 3(2):224–241, 2005a.
- E. Hyvönen, A. Valo, V. Komulainen, K. Seppälä, T. Kauppinen, T. Ruotsalo, M. Salminen, and A. Ylisalmi. Finnish national ontologies for the semantic web - towards a content and service infrastructure. In *Proceedings of International Conference on Dublin Core and Metadata Applications (DC 2005)*, Nov 2005b.
- M. Junnilla. Tietosisältöjen semanttinen yhdistäminen toimintakuvausten avulla (Event-based approach to semantic linking of data content). Master's thesis, University of Helsinki, March 6 2006.

²⁴<http://www.seco.tkk.fi/projects/finnonto/>

- M. Junnila, E. Hyvönen, and M. Salminen. Describing and linking cultural semantic content by using situations and actions. In *Semantic Web at Work - Proceedings of the 12th Finnish Artificial Intelligence Conference STeP 2006, Volume 1.*, Nov 2006.
- T. Kauppinen, R. Henriksson, J. Väättäin, C. Deichstetter, and E. Hyvönen. Ontology-based modeling and visualization of cultural spatio-temporal knowledge. In *Semantic Web at Work - Proceedings of the 12th Finnish Artificial Intelligence Conference STeP 2006, Volume 1.*, Nov 2006.
- T. Kauppinen and E. Hyvönen. Modeling and reasoning about changes in ontology time series. In R. Kishore, R. Ramesh, and R. Sharman, editors, *Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems*. Springer-Verlag, Dec 2006. In press.
- E. Mäkelä, E. Hyvönen, and S. Saarela. Ontogator — a semantic view-based search engine service for web applications. In *Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*, Nov 2006.
- E. Mäkelä, E. Hyvönen, S. Saarela, and K. Viljanen. OntoViews – a tool for creating semantic web portals. In *Proceedings of 3rd International Semantic Web Conference (ISWC 2004), Hiroshima, Japan*, November 2004.
- T. Ruotsalo and E. Hyvönen. Knowledge-based recommendation based on heterogenous metadata schemas, 2006. Paper under construction.
- M. Salminen. Kuvien ja videoiden semanttinen sisällönkuvailu (Semantic content description of images and videos. Master's thesis, University of Helsinki, May 2006.
- A. T. Schreiber, B. Dubbeldam, J. Wielemaker, and B. J. Wielinga. Ontology-based photo annotation. *IEEE Intelligent Systems*, 16:66–74, May/June 2001.
- J. Sowa. *Knowledge Representation. Logical, Philosophical, and Computational Foundations*. Brooks/Cole, 2000.
- J. van den Berg. Subject retrieval in pictorial information systems. In *Proceedings of the 18th international congress of historical sciences, Montreal, Canada*, pages 21–29, 1995. <http://www.iconclass.nl/texts/history05.html>.
- A. Vehviläinen, E. Hyvönen, and O. Alm. A semi-automatic semantic annotation and authoring tool for a library help desk service. In *Proceedings of the first Semantic Authoring and Annotation Workshop, ISWC-2006, Athens, GA, USA*, November 2006. To be published.
- K. Viljanen, T. Käsälä, E. Hyvönen, and E. Mäkelä. Ontodella - a projection and linking service for semantic web applications. In *Proceedings of the 17th International Conference on Database and Expert Systems Applications (DEXA 2006), Krakow, Poland*. IEEE, September 4-8 2006.
- G. P. Zarri. Semantic annotations and semantic web using nkrl (narrative knowledge representation language). In *Proceedings of the 5th International Conference on Enterprise Information Systems, Angers, France (ICEIS 2003)*, pages 387–394, 2003.

VII

Publication VII

Eetu Mäkelä, Osma Suominen, and Eero Hyvönen. 2007. Automatic Exhibition Generation Based on Semantic Cultural Content. In: Lora Aroyo, Eero Hyvönen and Jacco van Ossenbruggen (editors), Cultural Heritage on the Semantic Web Workshop, 6th European Semantic Web Conference, ESWC 2009, Heraklion, Crete, Greece, May 31-June 4, 2009, pages 41–52.

© 2007 by the authors

Automatic Exhibition Generation Based on Semantic Cultural Content

Eetu Mäkelä, Osma Suominen, and Eero Hyvönen

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

Abstract. In this paper, we argue for a need to shift focus in semantic search from the items themselves to using them as lenses to wider topics. A system for doing this in the cultural heritage domain is presented, duplicating on the web the way exhibitions in the real world are organized. An interface for specifying such exhibitions is presented, combining a general narrative pattern with semantic autocompletion and the novel concept of domain-centric view-based search. This also solves a number of problems view-based search has previously encountered in the cultural heritage domain. Presented also are multiple visualizations for the exhibition, supporting the user in making sense of the data and in doing exploratory search.

1 Introduction

Traditionally, Internet search has been about finding a document or documents that answer the question posed by the searcher. Semantic Web search systems have mostly also held this viewpoint [1], using properties and concepts in domain ontologies to locate search objects annotated with them. For semantically annotated content analogous to text documents, this works adequately, but for qualitatively different material, it creates problems. To understand why, one must take a step back to look at information needs.

The many classifications of information needs [2–7] all agree that there is a major partition between lookup queries like “For my meal, I need a *white wine* with a *spicy flavor*” and more general information needs such as “tell me all about *spicy white wines*”. The former focuses on selecting, fact finding and question answering, while the latter deals with the more general objective of learning and investigation, containing in addition to searching also tasks such as comparison, interpretation, aggregation, analysis, synthesis and discovery [8]. Depending on domain, at least a significant part (22% [9]), or even the majority (70% [10], 67% [5]) of enquiries for information relate to these more general learning instead of spot queries.

Despite this, search research has only recently begun to move to this expanded domain, termed exploratory search [8]. We propose that a major reason for this is that as long as the information is encoded only inside documents,

learning and investigation searches are adequately catered for by the same functionality as fact finding, i.e. locating all matching documents and then perusing each for relevant data [6].

For semantically annotated content other than information documents, the situation is different. Often the useful information is not the object itself, but the relation between the object and the ontological resources associated with it. Now, for question answering such as what wine to have with a particular food, the answer is still a particular object with particular characteristics, and the old paradigm still works. For the more general type of queries, on the other hand, typical semantic web object databases fall short, as they contain no singular exposition about, e.g. “French spirits”.

However, if looked at from another perspective, the data contains ample information to answer someone wanting to know about French liquors. It is merely encoded differently, distributed across the multiple object annotations and ontologies. To pull this information out, one must move the focus from individual items to the set of objects with particular properties as a whole, and even further. What one actually wants is to look at the combination of the domain concepts “French” and “spirits” through the lens of the items.

Actually, if an interface capable of such can be created, the pieced nature of the information becomes an advantage, as the pieces can be combined to shed light on a much wider variety of topics than anyone could write an explanatory article on. This capability is even further enhanced if the database contains material of multiple different kinds. For example in the cultural heritage domain, with suitable material, one could learn not only about 19th century Finnish crafts, 19th century Finnish paintings etc., but actually of the 19th century Finland as a whole.

Based on this analysis, we argue that to support exploratory search tasks, Semantic Web application designers need to shift focus from object location to the creation of structured, domain-centric presentations based on those items.

2 Looking at Culture Through Its Products

Luckily for interface designers in the cultural heritage domain, there is already a real world counterpart for this functionality to take inspiration from. What is wanted is very similar to how exhibitions in real-world museums function, presenting a particular temporally, spatially and functionally constrained aspect of culture through its objects and art. As such parallels are an excellent cue for understanding the structure of an information presentation, we decided to make as much use of this as possible when designing the interface for our Culture-Sampo¹ [11] cultural heritage portal.

Our idea is to let users create virtual exhibitions that mimic the way real museums are organized, containing themed exhibition rooms of items and displays that together, through the objects, tell the story of a particular subject.

¹ <http://www.kulttuurisampo.fi/>

Our implemented system combines an exhibition specification interface based on view-based query constraining with multiple visualizations grouping the items according to domain facets the user is interested in. In the following, both of these components will be discussed further in their own chapters.

2.1 Specifying the Desired Exhibition

The CultureSampo portal is aimed at the general public. Therefore, our exhibition generation interface had to be as easy to use and understand as possible, while still allowing for a wide variety of different presentations to be generated. To accomplish these goals, we first set the parameters for what kind of exhibition definitions had to be possible. Analyzing real exhibitions, we created a general but verbally understandable structural pattern for describing them, on which we could build our interface. The pattern, with each part optional, is:

Tell me about *item type*
related by *role* to *domain concept* [and ...]
organized by *classification+role* [and *classification+role*].

While constrained and procedurally structured, this pattern still allows for a wide range of exhibitions to be specified, from e.g. “Tell me about weapons” to “Tell me about everything related to 19th century Finland and agriculture, organized by item type and purpose of use“ and “Tell me about toys manufactured in China, organized by time of manufacture and place of use”. Figure 1 shows this in our actual interface. On the left are the exhibition specification controls, layed out to directly reflect our narrative structure.

Domain-Centric View-Based Constraining For the selector components used to fill this pattern, we looked to the recently popularized [12–16] paradigm of view-based search (also known as faceted browsing) combined with semantic autocompletion [17]. Of these, view based search is based on organizing the search data into multiple categorizing views and then picking categories as constraints from the views and has already shown good promise for fulfilling learning type search needs [18].

For our particular needs, the paradigm has a number of user benefits [19]. First, because the collection is visualized along different categorizations, the user is immediately familiarized with its contents and the way they are organized. Functionally, the user gets information on what the possible constraints are and how selecting them will affect the result set. Second, the multiple viewpoints allow the user to start constraining from the perspective most familiar to them. Finally, this visualization already intuitively shows the wider context in which the result set lays, thereby contributing to the users ability to answer questions of the result set as a whole, and not just of individual item.

In addition to interface benefits, the paradigm fits Semantic Web data well. The rich metadata in semantic databases is just the sort of multifaceted data whose exploration the paradigm supports. Also, because the metadata values are

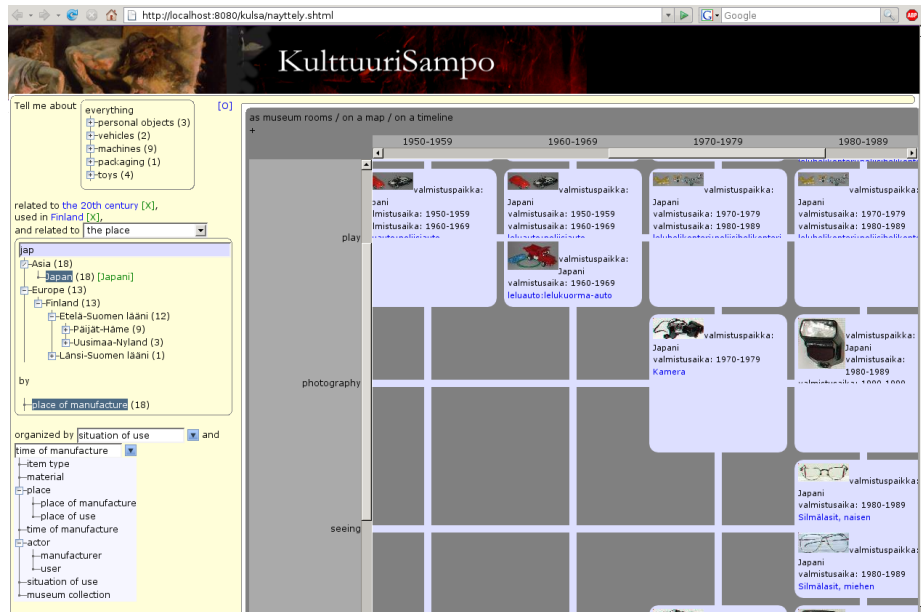


Fig. 1. The CultureSampo user interface, with important elements manually translated into English. The exhibition specification interface is located on the left, while the exhibition itself is visualized on the right. Showing is an exhibition on the types of items Japan exported to Finland in different parts of the 20th century.

resources organized in ontological hierarchies, they provide an excellent basis for creating usable, well-structured categorizing views.

Traditionally in Semantic Web view-based search systems views have been formed by selecting a property, such as “place of manufacture”, and enumerating all the values of that property as selections. In the cultural domain, this has caused problems, as there are typically many content types with differing properties such as “mentioned place” (poem) and “depicted place” (painting, photograph) [14]. Fortunately, our move from the objects to the domain concepts presented us with a natural solution, the novel variation of domain-centric view-based search [20]. Here, the properties are relegated to a secondary role, and the views were built instead based on the ontological ranges of those properties, i.e. the set of topical domain ontologies. In CultureSampo, we ended up with nine views: object types, places, times, actors, events, styles, materials, techniques and museum collections, with attached properties such as place of manufacture, depicted place and place of birth.

In a prior version [11], we discarded the properties from the user interface completely, and our views selected all items related in any way to the domain concepts (e.g. show anything related in any way to Poland). However, without any reference to the properties, the users were lost as to what a selection did and why any particular item was included in the result set. In addition, the

expression power of the interface diminished, as one could no longer e.g. search for items made in Japan but used in Europe.

These problems were solved by two measures. First, in the presentation, for each item an explanation is included of the property-concept relationships that places that item in the result set. Second, the properties were brought back to the views, but in a different form, shown in the place facet of figure 1. Now, a view consists of two selectors: one for selecting the domain concept and another for limiting based on the role (property) that the concept has in relation to the search items. Here, the user is free to search both with and without specifying a role, actually increasing the expressivity of the view-based search paradigm.

In CultureSampo the views are not all constantly visible. This is because here they are used as selectors in the context of a larger pattern, which we wanted to emphasize. Showing many views at once by default would have cluttered the screen, reducing intuitive grasp of the interface. Instead, by default visible are two views, one static for constraining by item type, and another for constraining by a domain concept. The domain view visible in any given moment is selected from a dropdown menu (shown on the left in figure 4(a)). In addition, power-users can also bring up further concurrent views.

View-Independent Semantic Autocompletion The multiple views in the view-based search paradigm make it easy for users to browse their options. However, for users knowing precisely what they want, a shortcut and a single point of entry is desired. In our system, this is accomplished by a semantic autocompletion [17] component, shown in figure 2.

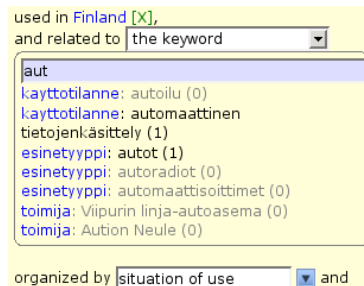


Fig. 2. The view-independent semantic autocompletion component of the CultureSampo exhibition specification interface.

Here, the user merely types in what they are looking for, and the system instantly responds with matching keywords to be used as possible constraints. These are both annotations directly related to the items, as well as matching selections in any of the facets. If the keyword typed gives sufficient specificity for the user, it isn't even necessary to make any further selections, as the query state is also instantly updated, using the union of the matches as a constraint. This

makes it possible for a user to interact with the system in a more experimenting way, typing in a keyword that pops into their mind and immediately seeing if the portal contains any related material, as well as what kind of exhibition it generates.

These keyword-search derived constraints can also be combined with those selected from domain views. For further supporting in-between user behaviours, all the domain views internally support a different form of semantic autocompletion, with the results shown directly in their hierarchical view context. This functionality is depicted in the place facet of figure 1.

2.2 Visualizing the Exhibition

As the user makes choices constraining the material, also the exhibition view is updated. Here, our primary association strived for is of a typical museum, with themed floors and rooms of exhibits, combined with custom presentations.

For the museum room visualization, the same categorizing view structures used for selection are utilized. The idea is simply to project the items in the result set onto a two dimensional matrix whose rows and columns are comprised of a flattened list of concepts in the two domain facets chosen for organization. This way, each cell in the matrix corresponds to room combining two themes, such as “18th century agriculture”, followed in one dimension by “19th century agriculture”, and “18th century hunting” on the other. This matrix is then visualized, either as is for a single-floor museum complex view depicted in figure 1 or row by row, for a more traditional floor and room museum plan, shown in figure 3. While the latter plan allows us to eliminate empty rooms on a floor by floor basis thereby optimizing display area, the single-floor view allows one to also see more large-scale structural changes. In figure 1, for example, one can see how in 1950-1970 most Japanese-made items that made their way into Finland were toys, but beginning in the 70’s there is an increase in the import of high-tech products. Both visualizations are scrollable where they do not fit in the screen at once.

For particular domains, special presentations particularly suited to them are available [21]. In our current system, these are a timeline visualization for the time facet and a map visualization for the place facet. These are shown in figure 4. In both of these visualizations, the second dimension, if specified, is expressed by marker coloring.

3 System Architecture

A primary design requirement for the CultureSampo exhibition interface was to allow the user to explore the data in the portal in a highly interactive and experimenting fashion. To support this, the interface had to be very responsive, updating all views instantly to match user commands. In our implementation, this is achieved through a highly optimized view-based search engine combined

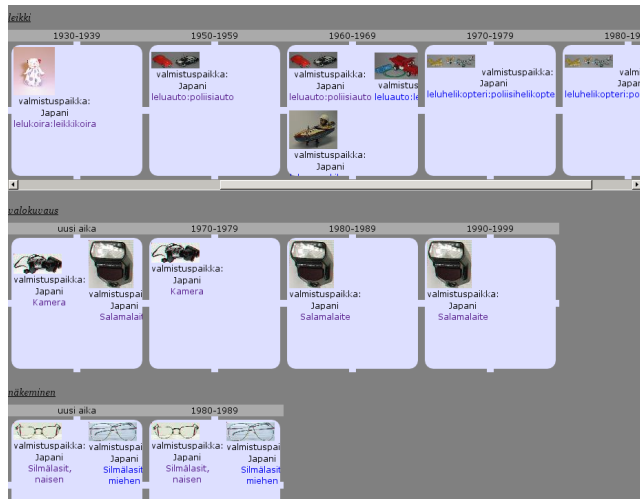
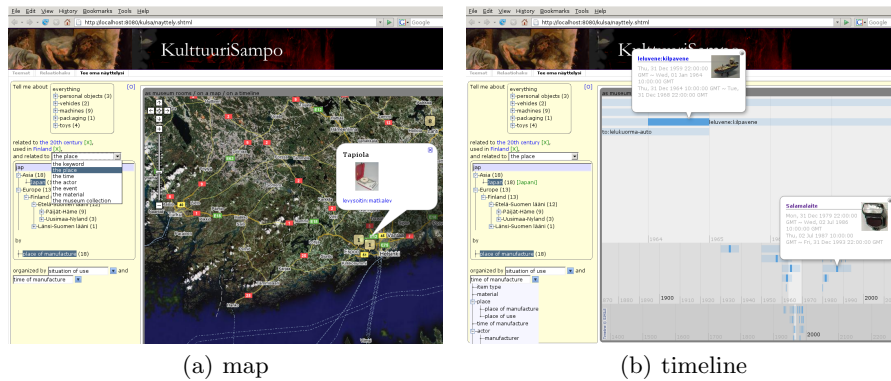


Fig. 3. The results organized according to museum floor and room in CultureSampo



(a) map

(b) timeline

Fig. 4. Special presentation visualizations in CultureSampo

with logic that minimizes the amount of data needed to be sent at any time between the server and the client browser.

Our view-based search engine makes use of existing tools for implementing efficient indices. The hierarchical view trees are indexed using an interval indexing scheme [22] inside an SQL database. This allows transitive constraints to be processed using a single SQL index scan. For textual matching, either SQL or Apache Lucene² indices are used depending on specific needs. On top of these indices, there is a custom processing component that gathers all SQL and Lucene constraints respectively to single executable queries, as well as implements partial query caching for intersection queries. This last functionality is extremely

² <http://lucene.apache.org/>

important for throughput, as an overwhelming majority of queries in view-based search are those updating hit counts in the views, i.e. intersecting the current query with potential future constraints.

To further increase responsiveness, there is logic on both the server and in javascript on the client that minimizes the amount of information that needs to be sent over the network. For example, the children of a node in a view tree are only sent to the client on request. However, the server also keeps a list of nodes that have been sent. This allows their hit counts to be sent in a single batch without explicit individual querying. It also makes it possible for the server-side semantic autocomplete component to know if matched tree nodes are already known by the client or need to be sent along with the text-match response itself. The information itself is sent as javascript objects, automatically mapped from JavaBeans with Direct Web Remoting³. This additionally reduces the amount of information sent, as all layout of the results is done on the browser. For this side of the interface, the system makes use of the widget functionality provided by the Dojo javascript library⁴. This allows us to use general and reusable widgets [23] with attached HTML and CSS templates for the views and other interface components, instantiated on need as they are brought up.

4 Related Work

Many of the view-based search systems already mentioned [12–16] support exploratory search to some extent. Of these, one in particular needs to be mentioned. The Exhibit system [12] by the Simile project has clearly been designed with similar explorative search goals in mind and has an interface strikingly similar to our own, even down to providing map and timeline visualizations. Relating to our exhibition room visualization, they provide a grouping similar to our floors plus rooms plans, but not our two-dimensional matrix.

However, there is also a major difference between the two systems, indicative of the major importance our shift of focus has had. That is, Exhibit still follows the traditional viewpoint of concentrating only on the search objects themselves. Being based on traditional view-based search, the system only really supports a single item type at a time, being susceptible to the problem of view proliferation as the number of annotation schemas and consequently projected properties increases. Not utilizing domain-centric view-based search also means that its grouping and visualization capabilities cannot be used to shed light on domain concepts — a major idea of our system.

Architecturally, the two systems are quite different. Exhibit exists completely in javascript on the client side and supports only a few thousand items and simple flat classifications, being intended as a lightweight solution for casual users to publish their structured data on the web. Our system on the other hand is intended for publishing much larger collections and includes optimized interplay between the client and server to support this.

³ <http://getahead.org/dwr/>

⁴ <http://dojotoolkit.org/>

On the subject of multi-type view-based search in the cultural domain, the /facet -system [14] utilized in the MultimediaN project [24] also tackles this problem. Their choice is mostly to simply promote the “item type” facet as a first choice, as choosing from it drastically reduces the amount of property facets available. However, this only alleviates the problem, and diminishes the freedom of the user in starting from the facet most natural to them. Otherwise, the system also contains a functionality through which complex constraints spanning multiple types can be formed, allowing one to specify for example a pattern such as “Find all paintings by artists living in Paris”. While this was deemed too complex for our needs, in other fields it might supplement our pattern-filling approach nicely.

5 Conclusions and Future Work

In this paper, we have argued the need for a shift of focus in semantic search from item location to presentation generation and support for exploratory search. In particular, we argue that often what is interesting in semantic databases are not the items themselves, but how they shed light on a theme described by a particular combination of domain concepts.

For the cultural heritage domain, museum exhibitions offer a suitable parallel to this idea. We have taken advantage of this in our CultureSampo portal, combining an intuitive, yet expressive exhibition generation interface with different kinds of exhibition visualizations. On the exhibition generation side, our major contribution is the narrative query pattern for forming exhibitions combined with the concept of domain-centric view-based search, which allow us to cater to both searching for items having particular properties, as well as pure domain exploration.

On the exhibition visualization side, we have created a simple, general-purpose visualization, as well as complemented it with special purpose visualizations. Even these simple visualizations already give significant support for a user wanting to make sense of the data. However, here there is still also much more that could be done. Our next user interface functionality will be to allow the user to select some rows or columns from the matrix for specific comparison and study. It may also be possible to aid such comparison work by automatically extracting from the data meaningful differences and similarities between neighboring exhibition rooms, such as “18th century agricultural items are more often made of wood than 19th century ones”.

While our current user interface has been created exclusively for the CultureSampo portal, the actual architecture and functionality is very general, modular and configurable. The interface has also already been tested with alternative materials. Pursuant to this, we are studying ways to make the application configuration as effortless as possible, in order to provide the functionality we have over Exhibit also for generic Semantic Web content.

In addition to implementing these new functionalities, we also plan to conduct more thorough understandability and usability tests on this interface, as compared to competing choices.

Acknowledgements

This research is part of the National Finnish Ontology Project⁵ (FinnONTO) 2003-2007, funded by the National Technology Agency (Tekes) and a consortium of 36 companies and public organizations.

References

1. Hildebrand, M., van Ossenbruggen, J., Hardman, L.: An analysis of search-based user interaction on the semantic web. Technical report, Centrum voor Wiskunde en Informatica (NL) (2007)
2. Wilson, T.D.: Information needs and uses: fifty years of progress. In Vickery, B., ed.: *Fifty years of information progress: a Journal of Documentation review*, London, Aslib (1994) 15–51 <http://informationr.net/tdw/publ/papers/1994FiftyYears.html>.
3. Belkin, N.J., Marchetti, P.G., Cool, C.: Braque: design of an interface to support user interaction in information retrieval. *Inf. Process. Manage.* **29**(3) (1993) 325–344
4. Cool, C., Belkin, N.J.: A classification of interactions with information. In Bruce, H., Ingwersen, P., Vakkari, P., eds.: *Emerging frameworks and methods; Proceedings of the 4th international conference on conceptions of Library and Information Science (COLIS4)*, Greenwood Village, CO: Libraries Unlimited (2002) 1–15
5. Choo, C.W., Detlor, B., Turnbull, D.: Information seeking on the web: An integrated model of browsing and searching. *First Monday* **5**(2) (2000) <http://firstmonday.org/issues/issue5.2/choo/index.html>.
6. Jansen, B.J., Smith, B., Booth, D.: Learning as a paradigm for understanding exploratory search. In: *Proceedings of the SIGCHI 2007 Exploratory Search and HCI workshop*. (2007)
7. Anderson, L., Krathwohl, D.: *A taxonomy for Learning, Teaching and Assessing: A Revision of Bloom’s Taxonomy of Educational Objectives*. Longman, New York (2000)
8. Marchionini, G.: Exploratory search: from finding to understanding. *Commun. ACM* **49**(4) (2006) 41–46
9. Cole, P.F.: The analysis of reference query records as a guide to the information requirements of scientists. *Journal of Documentation* (14) (1958) 197–207
10. Wilson, T.D.: Current awareness services and their value in local government. In: *Proceedings of the 40th Congress of the Federation Internationale de Documentation*. (1980)
11. Hyvönen, E., Ruotsalo, T., Häggström, T., Salminen, M., Junnila, M., Virkkilä, M., Haaramo, M., Kauppinen, T., Mäkelä, E., Viljanen, K.: CultureSampo—Finnish culture on the semantic web. The vision and first results. To appear in: K. Robering (Ed.), *Information Technology for the Virtual Museum*. LIT Verlag, 2007 (2007)

⁵ <http://www.seco.tkk.fi/projects/finnonto/>

12. Huynh, D., Karger, D., Miller, R.: Exhibit: Lightweight structured data publishing. In: 16th International World Wide Web Conference, Banff, Alberta, Canada, ACM (2007)
13. Schraefel, M.C., Wilson, M., Russell, A., Smith, D.A.: mSpace: improving information access to multimedia domains with multimodal exploratory search. *Commun. ACM* **49**(4) (2006) 47–49
14. Hildebrand, M., van Ossenbruggen, J., Hardman, L.: /facet: A browser for heterogeneous semantic web repositories. In: *The Semantic Web - Proceedings of the 5th International Semantic Web Conference2006*. (2006) 272–285
15. Oren, E., Delbru, R., Decker, S.: Extending faceted navigation for rdf data. In: *The Semantic Web - Proceedings of the 5th International Semantic Web Conference2006*. (2006) 559–572
16. Mäkelä, E., Hyvönen, E., Sidoroff, T.: View-based user interfaces for information retrieval on the semantic web. In: *Proceedings of the ISWC-2005 Workshop End User Semantic Web Interaction*. (2005)
17. Hyvönen, E., Mäkelä, E.: Semantic autocompletion. In: *Proceedings of the first Asia Semantic Web Conference (ASWC 2006)*, Beijing, Springer-Verlag, New York (2006)
18. White, R.W., Muresan, G., Marchionini, G.: Evaluating advanced search interfaces using established information-seeking models. *Information Processing and Management* (2007) to appear in special issue on Evaluating Exploratory Search Systems.
19. Mäkelä, E.: View-based search interfaces for the semantic web. Master’s thesis, University of Helsinki (2006)
20. Mäkelä, E., Ruotsalo, T., Hyvönen, E.: Domain-centric view-based search. Submitted for review. (2007)
21. Alonso, O., Baeza-Yates, R., Gertz, M.: Exploratory search using timelines. In: *Proceedings of the SIGCHI 2007 Exploratory Search and HCI workshop*. (2007)
22. Christophides, V., Plexousakis, D., Scholl, M., Tourtounis, S.: On labeling schemes for the semantic web. In: *WWW ’03: Proceedings of the 12th international conference on World Wide Web*, New York, NY, USA, ACM Press (2003) 544–555
23. Mäkelä, E., Viljanen, K., Alm, O., Tuominen, J., Valkeapää, O., Kauppinen, T., Kurki, J., Sinkkilä, R., Käsälä, T., Lindroos, R., Suominen, O., Ruotsalo, T., Hyvönen, E.: Enabling the semantic web with ready-to-use mash-up components (2007) Submitted for review.
24. Schreiber, G., Amin, A., van Assem, M., de Boer, V., Hardman, L., Hildebrand, M., Hollink, L., Huang, Z., van Kersen, J., de Niet, M., Omelayenko, B., van Ossenbruggen, J., Siebes, R., Taekema, J., Wielemaker, J., Wielinga, B.J.: Multimedial e-culture demonstrator. In: *The Semantic Web - Proceedings of the 5th International Semantic Web Conference2006*. (2006) 951–958

VIII

Publication VIII

Eero Hyvönen, Eetu Mäkelä, Tomi Kauppinen, Olli Alm, Jussi Kurki, Tuukka Ruotsalo, Katri Seppälä, Joeli Takala, Kimmo Puputti, Heini Kuittinen, Kim Viljanen, Jouni Tuominen, Tuomas Palonen, Matias Frosterus, Reetta Sinkkilä, Panu Paakkari, Joonas Laitio, and Katariina Nyberg. 2009. 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 by the authors

CultureSampo—Finnish Culture on the Semantic Web 2.0: Thematic Perspectives for the End-user

Eero Hyvönen, Eetu Mäkelä, Tomi Kauppinen, Olli Alm, Jussi Kurki, Tuukka Ruotsalo,
Katri Seppälä, Joeli Takala, Kimmo Puputti, Heini Kuittinen, Kim Viljanen,
Jouni Tuominen, Tuomas Palonen, Matias Frosterus, Reetta Sinkkilä,
Panu Paakkarinen, Joonas Laitio, Katariina Nyberg

Semantic Computing Research Group (SeCo)

Helsinki University of Technology (TKK) and University of Helsinki

first.last@tkk.fi, <http://www.seco.tkk.fi/>

Abstract

We present an overview of CultureSampo, an ambitious system for creating a collective semantic memory of the cultural heritage of a nation on the Semantic Web 2.0, combining ideas underlying the Semantic Web and the Web 2.0. The system addresses the semantic web challenge of aggregating highly heterogeneous, cross-domain cultural heritage collections and other contents into a semantically rich intelligent system for human and machine users. At the same time, CultureSampo is an approach to solve the social and practical Web 2.0 challenge of organizing the underlying collaborative ontology development and content creation work of memory organizations and citizens. This paper focuses on CultureSampo's search, recommendation, and visualization services for the end-users. The key idea here is to access cultural heritage on the Semantic Web through nine "thematic perspectives", such as places on the maps, the social network of cultural persons, timelines, and narrative texts, e.g. biographies and literary works.

1. Publishing Collections Collaboratively on the Semantic Web 2.0

CultureSampo (<http://www.seco.tkk.fi/applications/kulttuurisampo/>) (Hyvönen et al., 2008) is a publication system and a portal by which memory organizations and citizens can publish their collections on the Semantic Web in a collaborative way. CultureSampo extends our earlier application "Museum-Finland—Finnish Museums on the Semantic Web" (Hyvönen et al., 2004, Hyvönen et al., 2005) (<http://www.museosuomi.fi/>), a system for publishing artifact collections on the Semantic Web, by sup-

porting publication of different kind of cross-domain contents, both material and immaterial. The system also introduces many novelties to both end-users and publishers.

Creating a system like CultureSampo is challenging due to two major reasons:

Semantic challenges. Cultural heritage content is semantically heterogeneous and available in various forms (documents, images, audio tracks, videos, collection items, learning objects etc.), concern various topics (art, history, handicraft etc.), is written in different languages, and is targeted to both nonprofessionals and experts. Furthermore, the content is semantically interlinked, as depicted in figure 1.1. For example, the content may contain a person’s narrative biography, works of art she created, places of interest where she lived in, Wikipedia articles or novels about or by the person, social connections to other persons, and events in the history that the person was related to. In our work, semantic web technologies (<http://www.w3.org/2001/sw/>) are used to address these challenges.

Organizational challenges. Museums, archives, libraries, media organizations, associations, and individual citizens create cultural heritage content independently from each other. This has led to a situation, where metadata produced by different organizations is usually incompatible with each other in terms of metadata schemas, vocabularies, and cataloging conventions. In our work, ideas underlying the Web 2.0 are used to support collaboration and promote interoperability in distributed content creation.

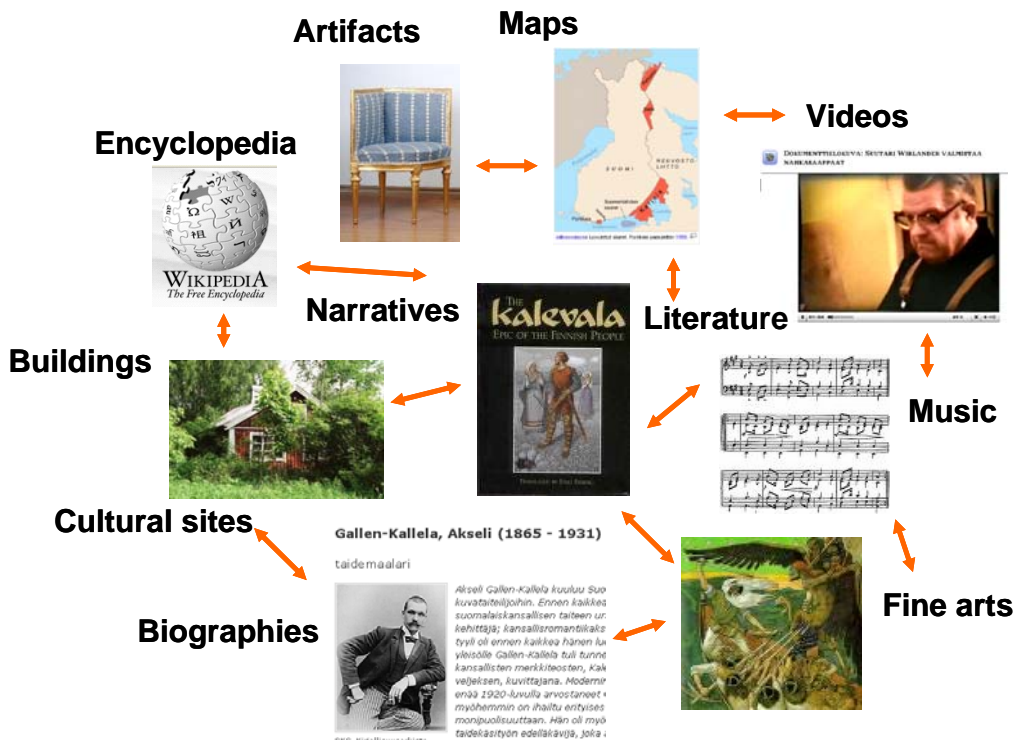


Figure 1.1 Cultural heritage is semantically heterogeneous and mutually linked.

CultureSampo has been developed since 2004 as a part the FinnONTO project (2003–2007, 2008–2010) (<http://www.seco.tkk.fi/projects/finnonto/>). The goal of this work is build a national level semantic web content infrastructure and demonstrate its usefulness in practical applications. The first public prototype of CultureSampo was published in September 2008 and can be used in three languages on the web at <http://www.kulttuurisampo.fi/>.

CultureSampo consists of three components (cf. figure 1.2):

1. **Collaborative ontology infrastructure.** The basis of CultureSampo is the national FinnONTO infrastructure (Hyvönen et al., 2008b). It includes a collaboratively created system of cross-domain ontologies and related ontology services for utilizing them cost-efficiently as services. The ontologies and

the services were published as the National Ontology Service ONKI (<http://www.yso.fi/>) on September 12, 2008.

2. **Content production system.** Our content creation model consists of a set of metadata models and a content creation process for producing and harvesting content from museums, libraries, archives and other organizations, as well as from individual citizens and (inter)national Web 2.0 sources.
3. **Semantic Web 2.0 portal.** The portal itself is unique in its use of versatile cross-domain semantic models, new semantic searching and browsing methods, and semantic visualizations for the end-users.

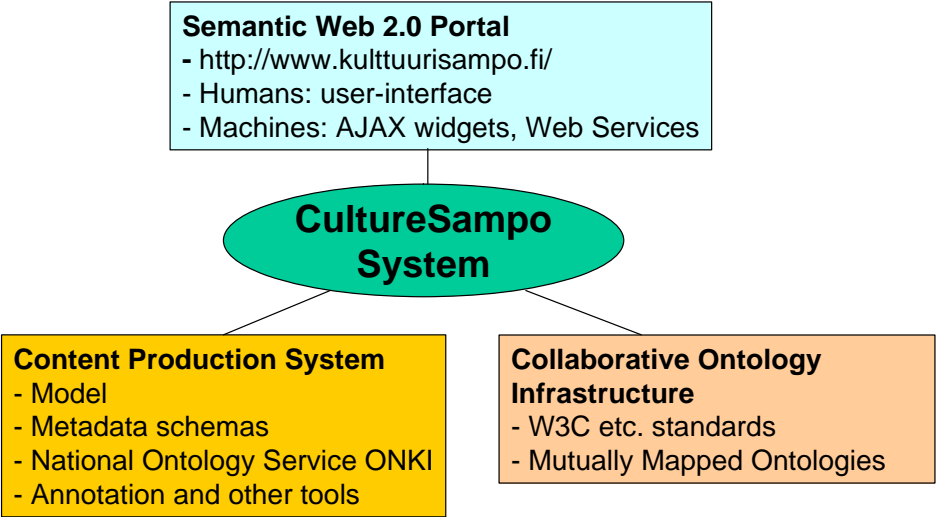


Figure 1.2 Three components of CultureSampo.

In the following, these three components are overviewed with an emphasis on the end-user interface of the portal. After this, contributions of the work to related research are summarized.

2. Collaborative Ontology Infrastructure

The ontologies of the FinnONTO infrastructure constitute an integral part of CultureSampo. They complement the generic, logic based W3C semantic web recommendations, such as RDF, OWL, and SPARQL, with domain specific concept descriptions in different domains. Most of the FinnONTO ontologies were developed by transforming nationally used thesauri into lightweight ontologies. The process was not purely mechanical, like e.g. in (van Assem, 2006), but also manual processing was required in order to refine the semantic thesaurus relations into full-blown subsumption hierarchies. In the FinnONTO model, the ontologies in different domains are developed in a distributed fashion by collaborating expert groups of different fields, and are mapped together to form a large national ontology called KOKO encompassing all domains. KOKO includes e.g. an upper ontology YSO (20 600 concepts), a cultural heritage ontology MAO (6800 concepts), an agriforestry ontology AFO (5500 concepts), an applied art ontology TAO (2600 concepts), and a photography ontology VALO (1900 concepts). Figure 2.1 illustrates the KOKO system of mutually mapped ontologies where YSO constitutes the upper ontology.

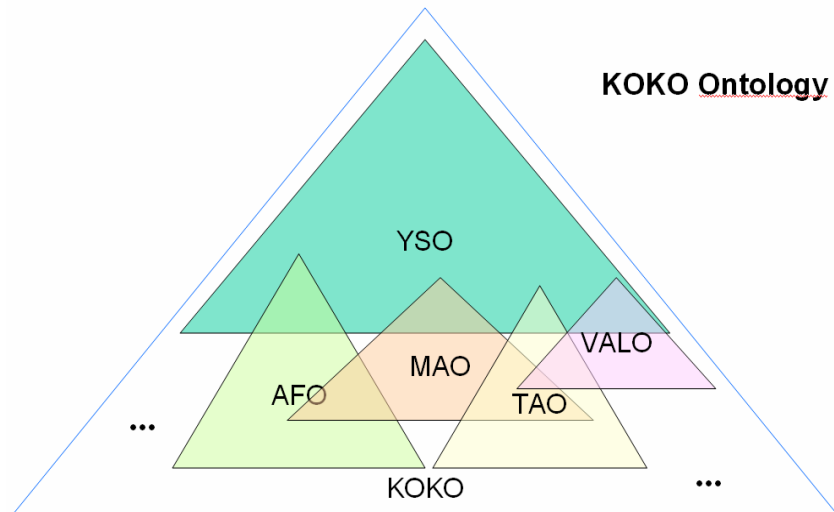


Figure 2.1. KOKO system of mutually mapped cross-domain ontologies.

The ontologies are provided to end-users not only in RDF/OWL form, as usual, but as ready to use semantic web widgets (Mäkelä et al., 2007) using Web 2.0 AJAX APIs, as well as through conventional Web Services. In addition to KOKO, CultureSampo also utilizes a geographical registry of 800,000 places in Finland, a spatiotemporal ontology of historical Finnish municipalities 1865–2007 (Kauppinen et al., 2008), and an ontology of persons and organizations. Furthermore, international vocabularies, such as the Iconclass (<http://www.iconclass.nl/>), the Art and Architecture Thesaurus (AAT) (http://www.getty.edu/research/conducting_research/vocabularies/aat/), and the Union List of Artists Names (ULAN) (http://www.getty.edu/research/conducting_research/vocabularies/ulan/) are used.

3. Content Production System

CultureSampo contains cultural objects of nearly 30 different content types including artifacts, paintings, drawings, sculptures, pieces of abstract art, novels, comics, web pages, three types of folklore, five types of folk music, photos, aerial photos, persons, organizations, biographies, historical events, skills, videos, buildings, and archeological sites. These content types are represented using 18 different metadata schemas. The aggregated knowledge base contains 52,000 cultural objects and 235,000 other resources, such as ontological class concepts and place instances. The cultural objects are described by 784,000 RDF property triples.

The content is enriched using reasoning, resulting in some 10 million property triples. The enriched knowledge base is used for intelligent information retrieval and for creating semantic recommendation links between objects. The content is represented using RDF and OWL, and SPARQL is used for querying recommendations. The system also utilizes external web resources through web services: all Wikipedia articles in English and Finnish that have coordinate information, as well as photographs of the Panoramio service Panoramio (<http://www.panoramio.com/>) can be found on CultureSampo's map views.

These information sources have diverse ownerships. The contents come from 22 Finnish museums, archives, and libraries, most of which produce their contents independently from each other using heterogeneous cataloging systems and practices. Wikipedia and Panoramio content is created (inter)nationally by the public. CultureSampo also has a commenting facility by which individuals can contribute new knowledge to individual content items, e.g. identify persons in an old photograph of a museum collection. In these ways, citizens are able to contribute to the national semantic memory. Furthermore, distributed content production based on the SAHA editor (Valkeapää et al., 2007) has been used internally in the system by the participating organizations.



Figure 3.1. CultureSampo system in a nutshell.

Figure 3.1 depicts the CultureSampo system as a whole. In the center is the KOKO ontology and other infrastructure ontologies. The collection items (cf. figure 1.1) around the ontologies are attached to the ontologies by metadata. The content providers depicted around the circle, i.e. CultureSampo, publish metadata locally and independently using shared metadata schemas and ontologies. The result is a large global semantic RDF network linking different contents together in ontologically meaningful ways.

From a semantic modeling viewpoint, one research focus of our work has been event- and process-based annotations used in artificial intelligence and knowledge representation (Sowa, 2000). In our case, events have been used for modeling cultural processes and narrative stories (Junnila et al., 2008) and for metadata schema integration (Ruotsalo, Hyvönen, 2007). The KOKO ontology was designed to support this by clearly separating events and processes from other concepts as in Dolce (Gangemi et al., 2002)

In some metadata schemas of CultureSampo it is possible to annotate content using processes in terms of events, subevents and their sequences. The model in use in the prototype is a simplified version of our earlier schema (Junnila et al., 2008). The portal then automatically generates an interactive representation of the process as a kind of a temporal table of contents. This system is used in the prototype for creating skill models, cultural process models, and documentation of processes in videos:

Semantic skill models. An example of a skill model is the model “Production of Ceramics” produced by experts at the University of Applied Arts in Helsinki. It illustrates and explains the composition of different work phases when manufacturing ceramics. At each phase, semantic recommendations to other relevant CultureSampo contents can be created automatically. For example, links to products in collections that were manufactured using the same techniques, are automatically obtained.

Semantic models of cultural processes. There is a similar kind of chronological model “A Year on a Farm” of the seasonal farming events and processes taking place at a typical farm in Finland. Again, tools and other materials from CultureSampo are linked automatically with related events. This model was created by a farmer employed at the Finnish Museum of Agriculture. The permanent exhibition of this museum is actually organized using the same idea of presenting farming events taking place during different yearly seasons.

Semantic documentation on videos. The annotation model can be applied also to documenting instances of actual skill events or processes documented on a video. The case example available on the

portal describes how the shoemaker Onni Wirlander manufactured a pair of traditional leather boots. The video was produced and the annotation created by the Espoo City Museum using the SAHA editor connected to the ONKI ontology services. The semantic model describes what happens at different (sub)sequences on the video. Semantic search can find not only the video as a black box, as in systems such as YouTube, but also points of interest inside the video. The video can be viewed directly starting from different points of interest. This is important with longer videos.

The examples above demonstrate our new ideas of representing and storing immaterial, procedural cultural heritage in the memory system, here descriptions of handicraft skills and cultural processes. Typical cultural heritage portals contain metadata only about static objects such as artifacts.

Content creation in CultureSampo, both ontologies and the metadata, is based on distributing the work to organizations and citizens in a Web 2.0 fashion. In this strategy, extra costs can be minimized by reorganizing the work done already in the organizations and in public. The work is supported by a number of generic FinnONTO tools, such as the metadata editor SAHA, information extraction tool POKA (Valkeapää et al., 2007), and the semantic content validator VERA (<http://www.seco.tkk.fi/services/vera/>).

Nine perspectives into cultural heritage

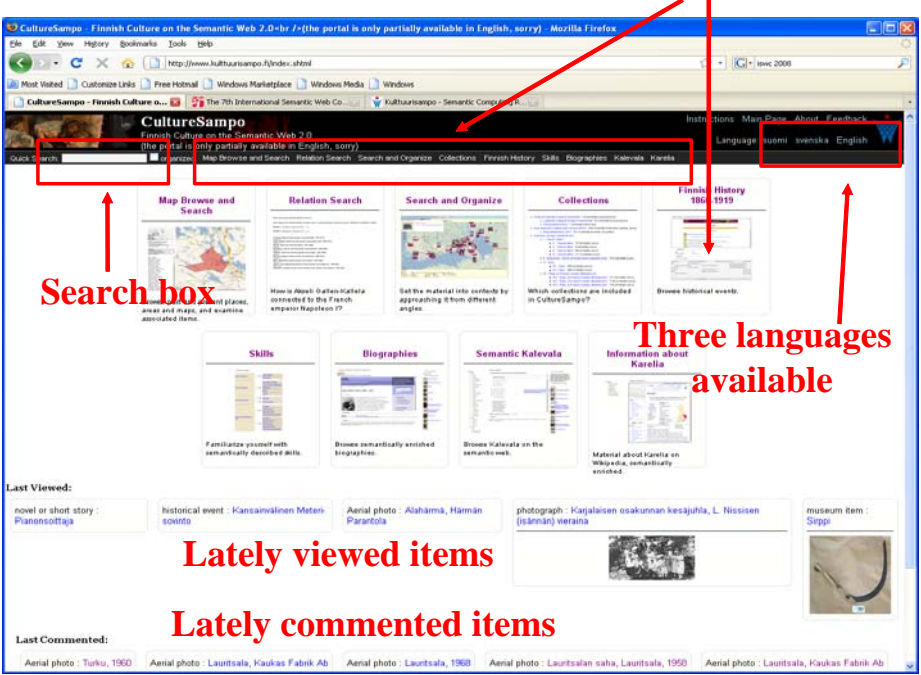


Figure 4.1 Front page of CultureSampo.

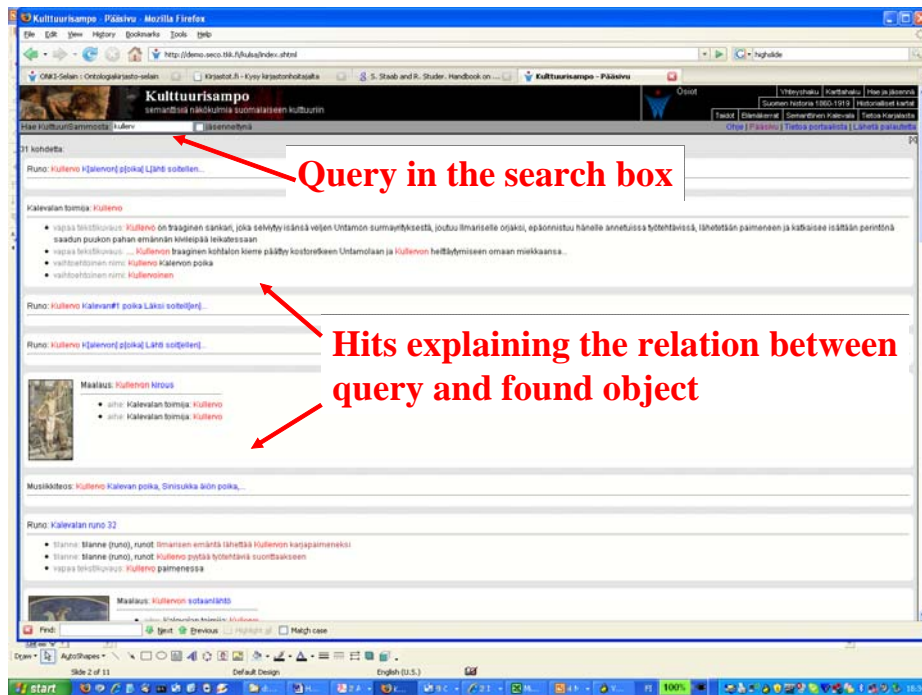


Figure 4.2 Using the semantic search box.

4. Semantic Web 2.0 Portal: Thematic Perspectives

Figure 4.1 depicts the front page of the portal CultureSampo. The system is multilingual: Finnish, Swedish, or English can be selected by the links in the upper right corner. However, because nearly all contents are in Finnish and translations of some parts of the Finnish KOKO ontologies are not available, the system is not equally powerful in other languages. On the bottom of the page, lately viewed objects from the knowledge base are shown, as well as objects that have been commented lately by end-users.

In the upper left corner, there is a Google-like search box for typing in a search query. CultureSampo utilizes semantic autocompletion (Hyvönen, Mäkelä, 2006) in order to guess the possible query words that the user is aiming at. Semantic query expansion based on the ontologies is used in order to enhance recall. Furthermore, the underlying ontologies are used to organize the hit results into meaningful categories. For example, in figure 4.2 the user is typing into the search box “Kullervo...” aiming probably at the hero “Kullervo” of the Kalevala epic. The hit results are categorized by the roles connecting the hero and the matched objects, e.g., paintings depicting him, runes telling about him etc.

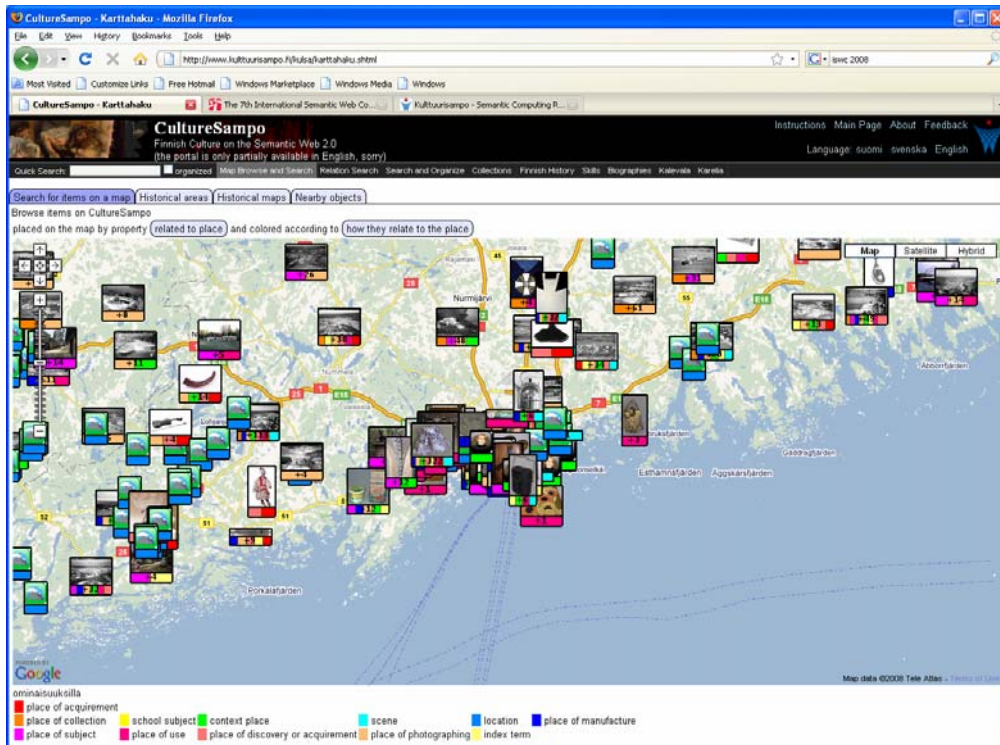


Figure 4.3 Collection objects on the map based on 12 different spatial relations.

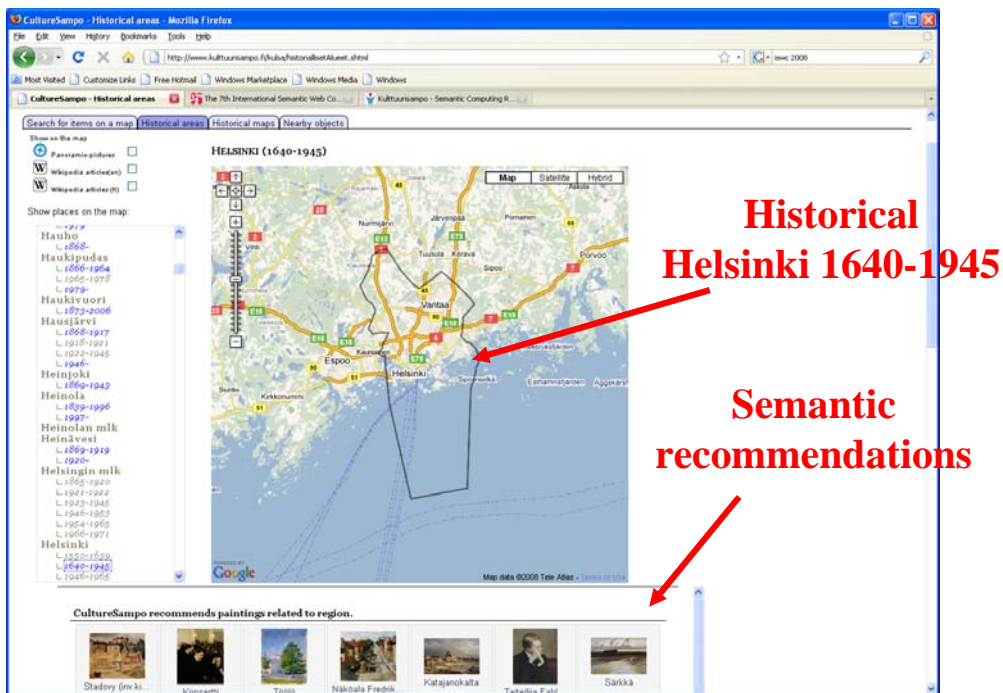


Figure 4.4 Historical areas tab.

A major novelty of the portal is to provide the end-user with an access to the cultural RDF knowledge base through nine “thematic perspectives”. They are available at the main entry page and as choices in the menu bar (cf. figure 4.1 in the middle). These perspectives are overviewed below in order to illustrate the underlying ideas and functionalities available. New perspectives can be added into the system fairly easily, because the underlying architecture is based on a set of general service components, such as the semantic recommending system.

Perspective 1: Maps Search and Browse Views

There are four map views [14] available using Google Maps. Each view has its own tab:

1. The tab “Search for Items on the Map” shown in figure 4.3 displays any collection object with coordinate information on Google Maps, and tells the semantic relations of objects to the places. There are 12 different spatial relations in use, such as “place of acquirement”, “place of subject”, and “place of manufacture”.
2. The tab “Historical Areas” is used for finding old Finnish counties with their digitized limits on the maps, based on the spatiotemporal Finnish place ontology SAPO [23]. In figure 4.4, a link directory to old places is shown on the left, and the user has selected the historical Helsinki. Semantic recommendation links to related collection items are automatically shown on the bottom of the page with short explanations about their relation to the place.

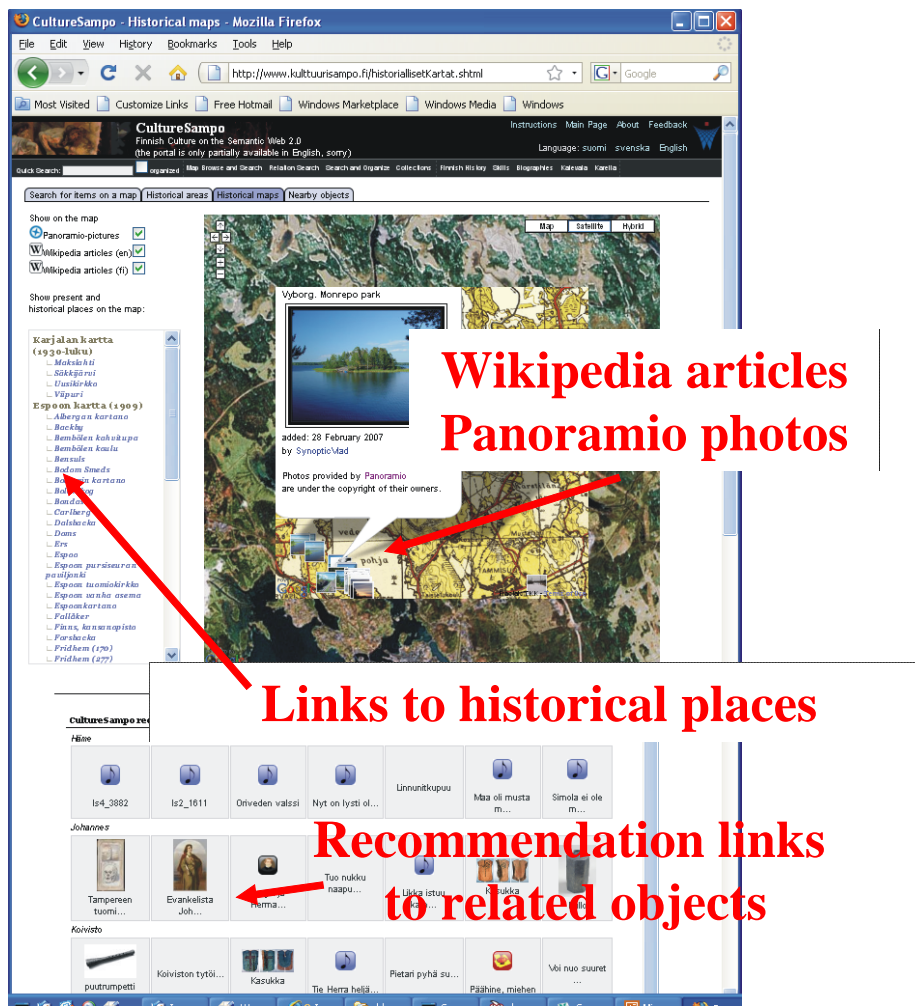


Figure 4.5 Historical maps on Google Maps.

3. The tab “Historical Maps” [14] is used for viewing historical maps layered semi-transparently over modern Google Maps. Wikipedia articles in Finnish or English with coordinate information, and photos from the Panoramio service can be seen on the maps and can be opened by clicking on them. For example, in figure 4.5 old Finnish Karelian maps are viewed semi-transparently on top of modern Russian Google Maps [23] (this area is part of Russia today). The user has located the historical Finnish park “Monrepos”, and found a modern Panoramio photograph there, taken by a contemporary Russian citizen. On the left, there is also the index of old Finnish places on the maps as direct selection links. Semantic recommendations to collection items related to historical places on the maps are displayed

below. In this case, for example, links to pieces of folk music and poems collected from the region can be seen, as well as related paintings, artifacts, and old aerial photographs.

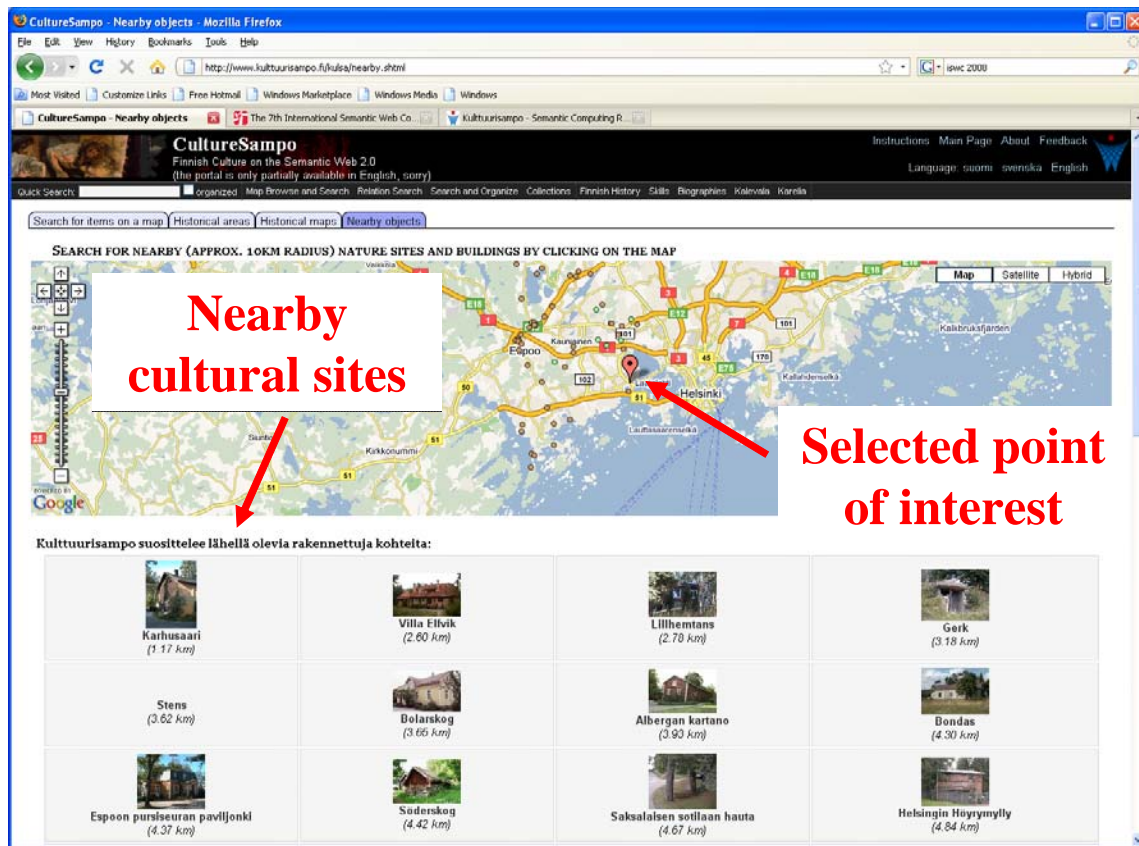


Figure 4.6 Finding nearby cultural sites.

- The fourth tab “Nearby objects” is used for finding nearby objects of interest: the user clicks a point and the system finds nearby cultural sites of interest (cf. figure 4.6).

Perspective 2: Relational Search

The relational search perspective is a demonstration of relational search [31] also called association identification or knowledge discovery [32]. Here the idea is not to search for objects but associative relation chains between objects. We used the ULAN registry of 120,000 artists and organizations with 390,000 names. Here the user can type in two names, using semantic autocompletion, and CultureSampo tells how the persons or organizations are related to each other by the social network based on some 50 different social roles (e.g., parent-of, teacher-of, patron-of etc.). The underlying social RDF/OWL network can also be browsed by a graphical network browser. For example, in figure 4.7 the user typed in Napoleon I (the French emperor) and Akseli Gallen-Kallela (a Finnish artist), and CultureSampo found a social path of 7 steps between the persons. The browsable social network of Napoleon I is depicted on the right hand side window.

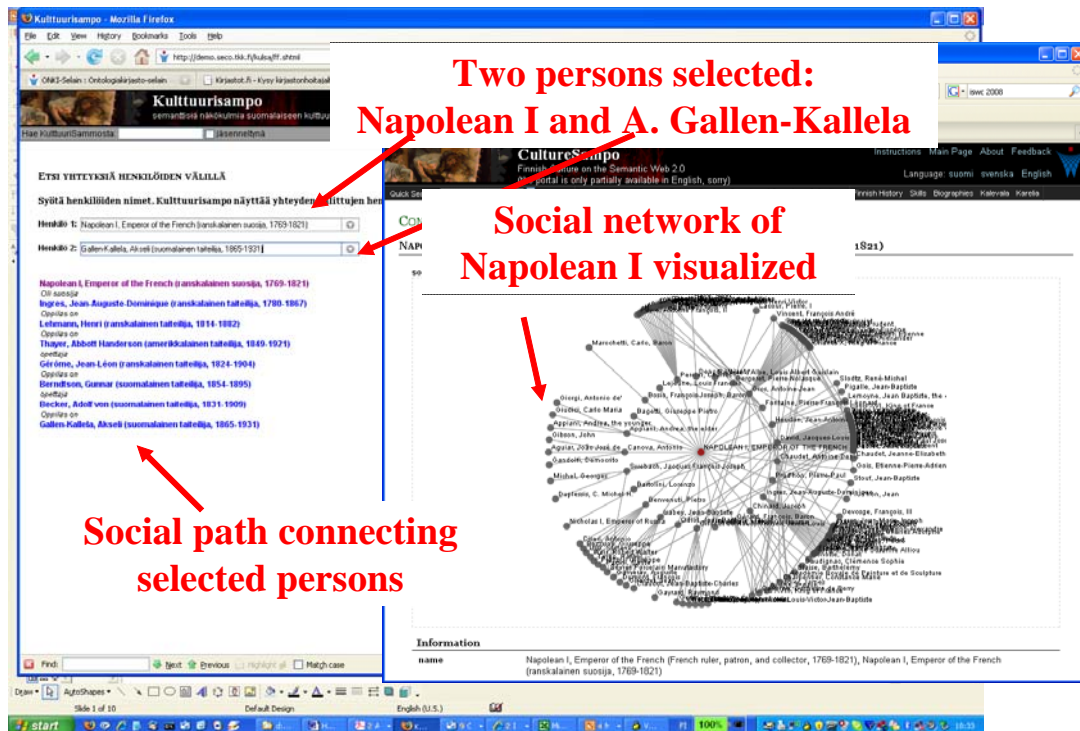


Figure 4.7 Answering the question "How is Napoleon I, the French emperor, related to Akseli Gallen.-Kallela, the Finnish artist?" by relational search.

Perspective 3: Search and Organize

In the search and organize perspective, the idea is to move beyond locating interesting singular items. Using the tools of this perspective, it is possible to analyze the contents and create organized presentations that are able to bring out interesting trends and information contained in the data as a whole. For example, it is possible to find out what were the most popular themes in Finnish fiction in the year 2007. In figure 4.8, the end-user is analyzing how beard fashions have changed during the ages by first searching for collection items depicting beards in different ways, and then projecting the results on a timeline. In figure 4.9, images of churches are visualized on the map in order to investigate the geographical distribution of different kind of churches in Finland.

The user interface of the perspective is divided into two functional parts. On the top, constraints for the result set are specified and changed. For example, in figure 4.0, the user has formulated the query "Tell me about photographs related to the keyword church".. On the bottom, the constrained result set can be organized, grouped and visualized according to different ontological facets as lists, on a map, or on a timeline. For example, it is possible to view the results "as a list according to the most popular theme", "according to time of manufacture on a timeline" or "colored according to style and placed on a map according to place of manufacture". For both query constraining and organizing, the perspective makes use of domain-centric faceted search (Mäkelä et al., 2007), a generalization of the faceted search paradigm to heterogeneous data.

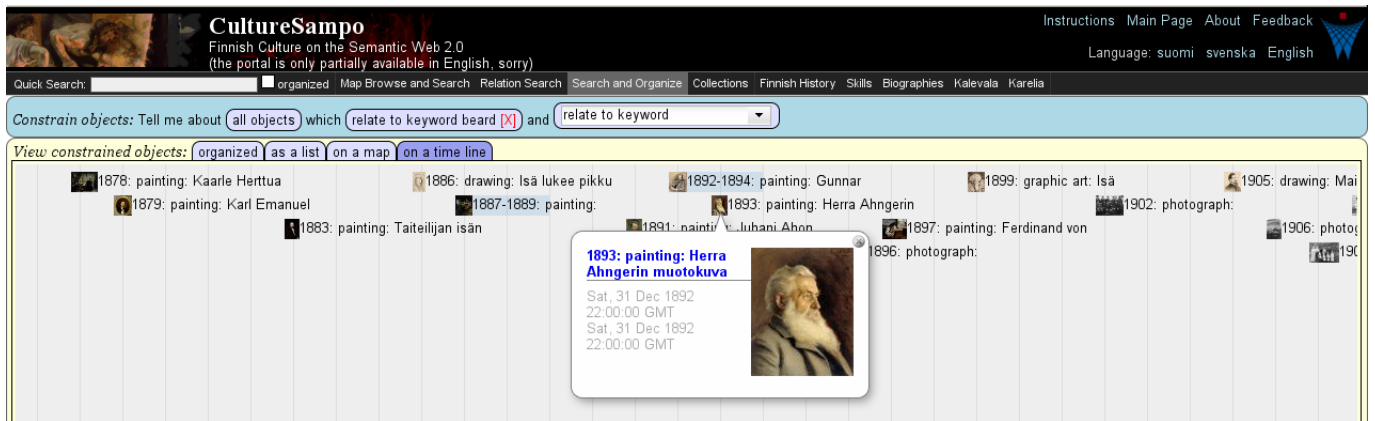


Figure 4.8 How have beard fashions changed during the ages? Paintings, photographs and other visual objects relating to the keyword 'beard' rendered on a timeline.

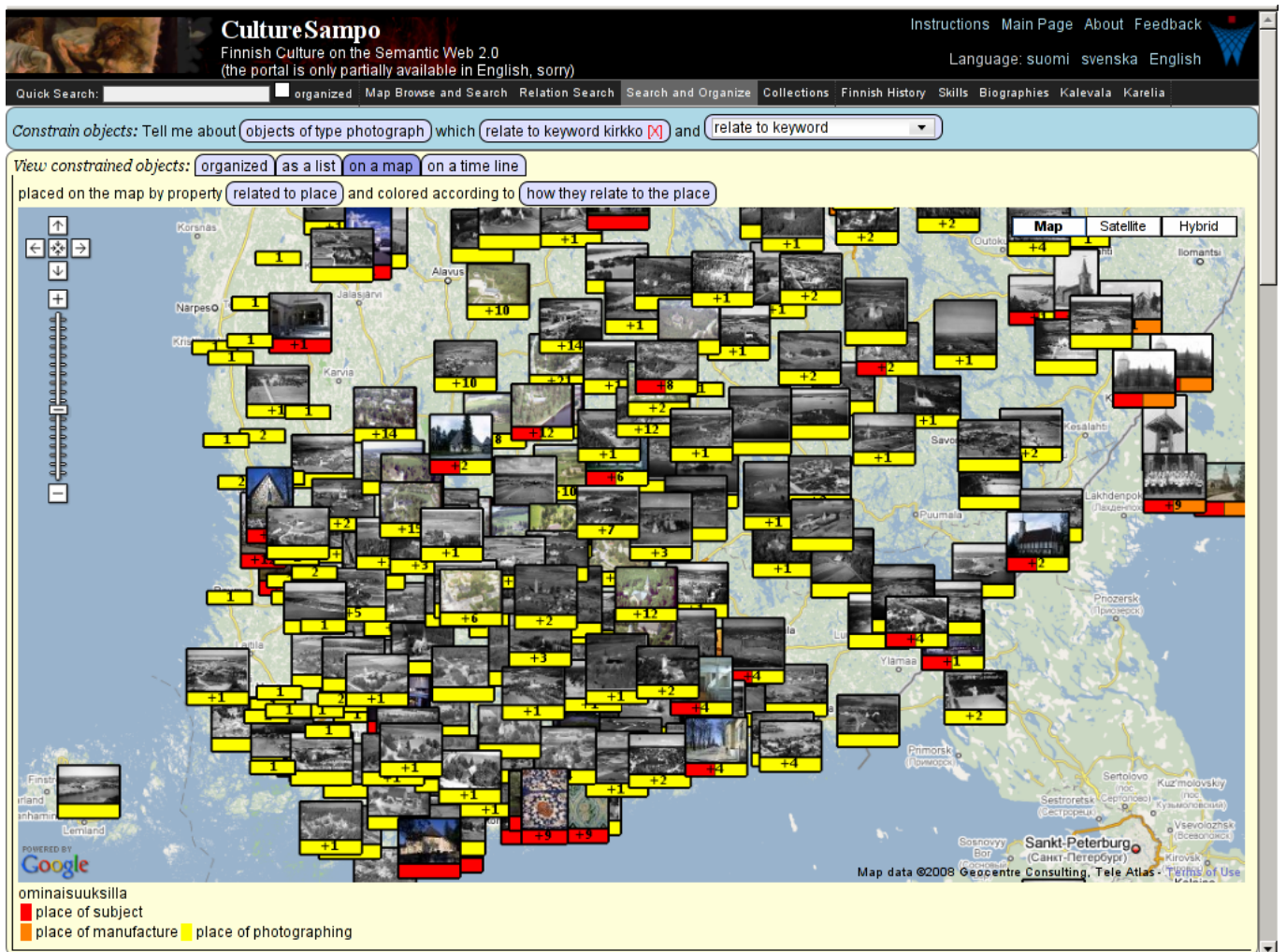


Figure 4.9 Is Finland a land of churches? Photographs relating to the keyword 'kirkko' (church in Finnish) rendered on a map.

Perspective 4: Collections

Here the contents can be accessed based on an organizational view. Each participating organization has an automatically generated home page in the system with links to its collections and the actual collection items.

Perspective 5: Finnish History

This view is based on an ontology representing events in the Finnish history [33]. These events are of interest on their own, but are also used to create semantic recommendations to other contents, e.g. to biographies of persons participating in the events. The history ontology in the prototype system contains 220 events on a timeline in 1860–1919. The content originates from the Agricola portal (<http://agricola.utu.fi/>) maintained by history researchers.

The screenshot displays the CultureSampo web application interface. At the top, the browser window shows the URL <http://www.kulttuurisampo.fi/kuisa/video.shtml?entURI=http://www.seco.59.fi/2/applications/saha%20stara>. The main content area features a video player titled "DOCUMENTARY FILM: SUUTARI WIRLANDER VALMISTAA NAHKASAAPPAAT". To the left of the video is a "Semantic process description" table with a timeline of events:

Time	Description
[0:20-2:40]	Pikilangan valmistus
0:45	Langan kerääminen
1:30	Langan pikaaminen pikilapulla hankaamalla
[2:40-3:26]	Puunsuojien valmistus
2:40	Naulalastujen vuoleminen
3:14	Muutoksen vaihtaminen laatuista

To the right of the video is a "Semantic recommendations" section with a table:

kohtauksen alkua (minseki)	tilanne
10:23	ompele
11:32	Takasauman ompelu
11:32	Päällisen ompelu varteen

Below the video player, there is a "Dynamic information about the video scene" section with a table:

kohtauksen alkua (minseki)	tilanne
10:23	ompele
11:32	Takasauman ompelu
11:32	Päällisen ompelu varteen

The interface also includes a search bar, navigation links, and a sidebar with "Kulttuurisampo suosittellee" recommendations.

Figure 4.9 Semantic video viewing with dynamic recommendations.

Perspective 6: Skills and Cultural Narratives

Figure 4.9 depicts the idea of documenting traditional skills semantically on videos, and providing the documentation through a semantic video viewer. The use case here is the shoemaker Wirlander producing a pair of leather boots. This process has been described semantically as a sequence of hierarchical events that take place in the video in different parts. The corresponding “table of contents” is automatically generated for the end-user on the left. The user can then view any part of the video by selecting items from the list.

In the screenshot, the user is viewing the part “sharpening of the knife”. At the same time, the system recommends items of interest in the collections, in this case different knives belonging to the collections of the museums. Each part of the video that forms a meaningful entity is a search object of its own, and can be found through the general search engine of the system.

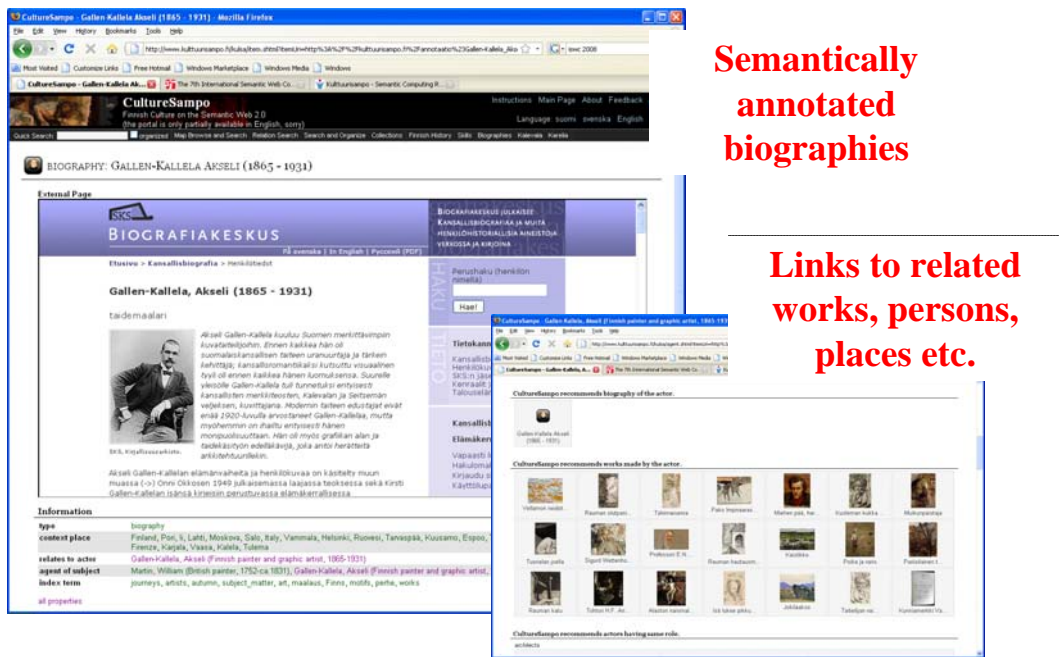


Figure 4.10 Biographical perspectives to cultural heritage.

Perspective 7: Biographies

In this view, biographies of the National Biography are used to access CultureSampo contents. When reading the biographies, related contents are shown as recommendations based on the concepts extracted from the text using the information extraction tool POKA (<http://www.seco.tkk.fi/tools/poka/>). Figure 4.10 depicts the situation where the user is reading the biography of Akseli Gallen-Kallela, retrieved from the server of the Finnish Literature Society (SKS), and provides her with links to e.g. the art works of the artist as well as biographies of other persons related to his life.

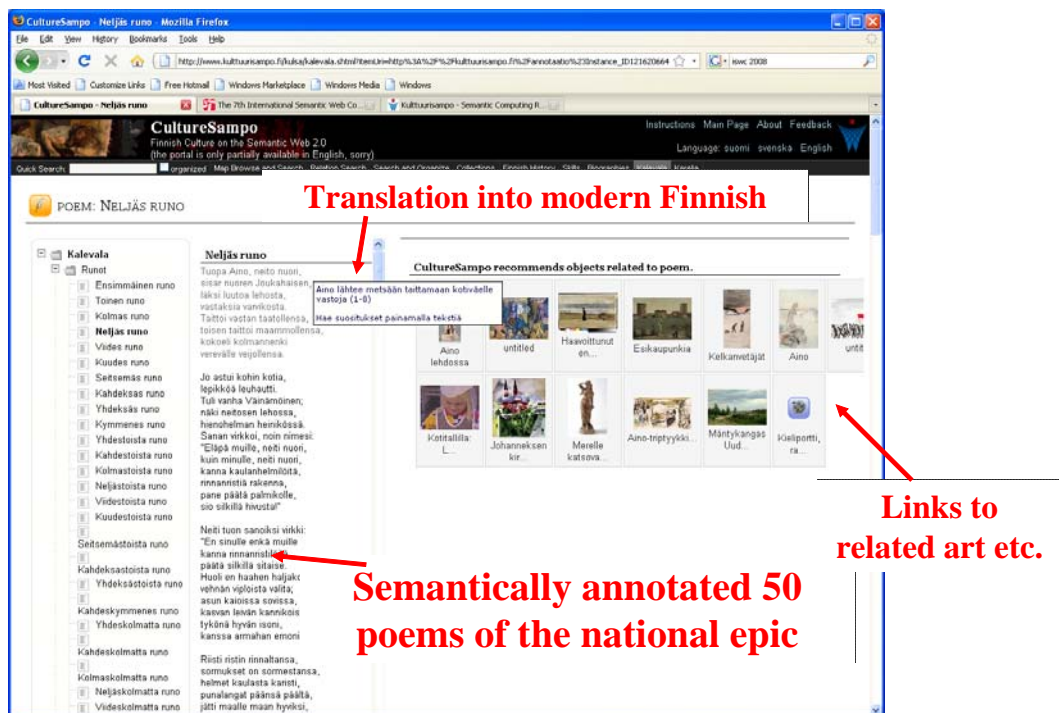


Figure 4.11 Semantic Kalevala epic.

Perspective 8: Semantic Kalevala

This view contains a semantically annotated version of the national epic of Finland, Kalevala, that is related in many ways to Finnish art and culture. The epic also has interesting links to old Finnish folklore, on which it is actually based, and to folk music lyrics. Thousands of runes and pieces of folk music are available in the portal. When reading Kalevala, annotations related to its subsequences can be viewed to help reading, and semantic recommendations to related materials in CultureSampo are automatically produced [17].

For example, in figure 4.11 the user is reading a part of the fourth rune in Kalevala. A part of the rune is selected, and a modern Finnish explanation of the part can be viewed. Furthermore, the system is suggesting semantic links to objects related to the part selected, in this case e.g. to sculpture depicting the fictive person Aino mentioned in the selected poem part.

Perspective 9: Karelia

This last thematic perspective contains Wikipedia articles about the Karelia area in Finland that has been influential to the Finnish culture. For example, lots of folklore has been collected into national archives from this area, and the Kalevala epic is strongly associated there. Like in the biographies view, the POKA system is used for extracting concepts from the texts (here web pages). Based on the extracted concepts, semantic recommendations to related contents are created for additional information (cf. figure 4.12).

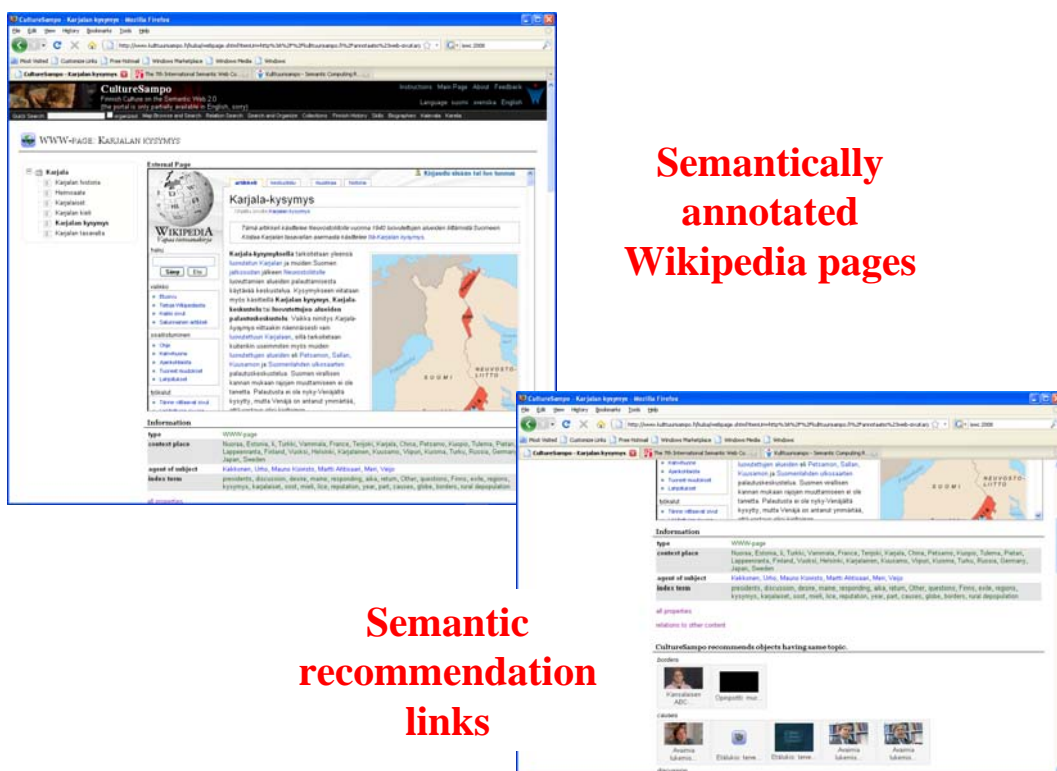


Figure 4.12 Karelia perspective.

In addition to human end-users, the system can be used by machines via AJAX interface and widgets. The application use case here is that collaboratively aggregated and semantically enriched national knowledge base can be used not only by the CultureSampo portal but also by other portals and systems on the web. To facilitate this we introduce the cost-effective idea of utilizing ready to use Web 2.0 mash-up services in the same spirit as Google Ads or Maps are used on external web pages and applications. In this way, museums, libraries, tourism portals, news papers, individual citizens, and other users can include CultureSampo materials, such as semantic search results and recommendation links, on their web pages using mash-ups. This clearly benefits everybody: the materials of the CultureSampo collaborative network get more visi-

bility and the external users can enrich their own materials for "free": only 1–2 lines of Javascript code is needed.

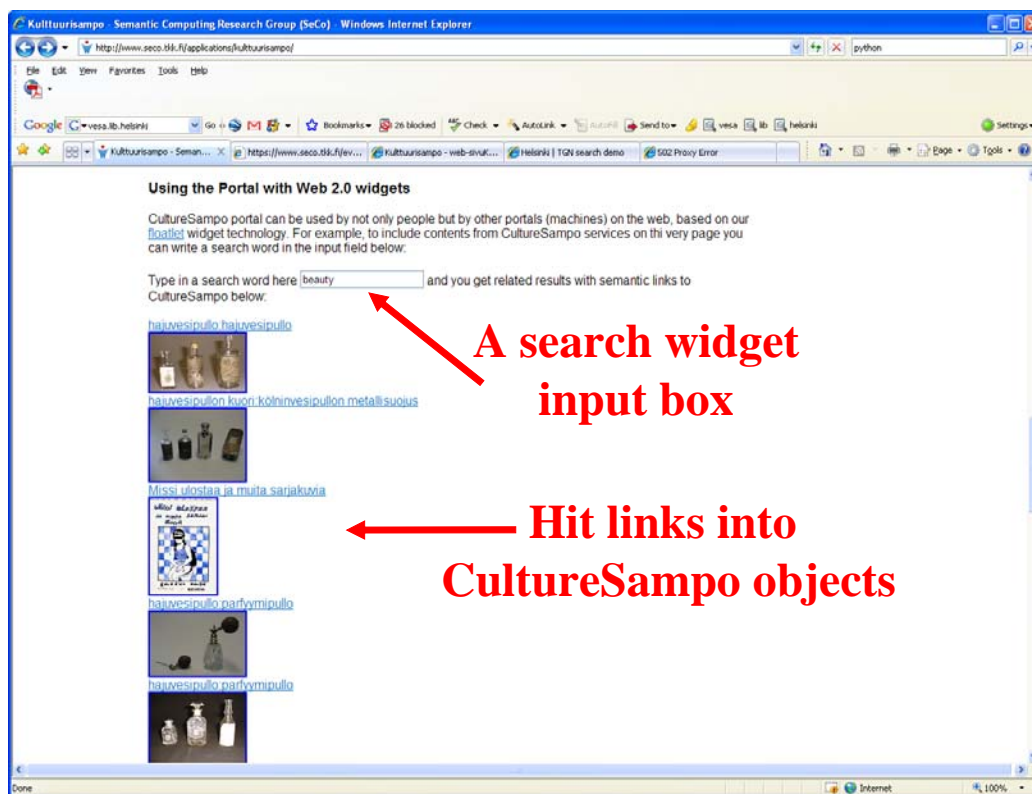


Figure 4.13 Semantic CultureSampo Widget.

An example mash-up is depicted in figure 4.13. Here the user types in a search word in the input field. After each character input, the CultureSampo service is queried and the matching results are displayed below the search field. In this case, the user is typing "beauty" in English, and bathroom equipments and some paintings related to the concept are returned. The actual contents are available only in Finnish. However, search can be performed successfully because parts of the underlying ontologies are available in English and Finnish.

5. Discussion

The vision and design of CultureSampo goes beyond current semantic web portals for cultural heritage (Hyvönen, 2009). The contributions of the work underlying the CultureSampo system have many facets and have been made during several years. The contribution of this paper is to give an overview of the whole system, its philosophy, and the new thematic perspective based end user interface, with pointers to papers and web sites explaining the approaches and software developed in more detail.

To summarize our work, CultureSampo contributes to research and development of semantic portals for cultural heritage especially in the following ways:

1. **Cross-domain content, ontologies, and metadata.** The system is highly cross-domain with dozens of content types and metadata schemas. Usually only one schema such as Dublin Core or VRA Core (<http://www.vraweb.org/projects/vracore4/>) is used in cultural portals (cf. e.g. (Schreiber et al., 2006; Wang et al., 2008)).
2. **Event-based narrative semantic models.** CultureSampo makes use of sophisticated semantic annotation models including events and processes.

3. **Semantic search and recommending.** The system uses new kinds of semantic search and recommendation techniques.
4. **Semantic visualizations.** The system has an exceptionally versatile selection of semantic visualizations available, such as different map views, timelines, graphs, process visualization, and semantic video viewing.
5. **Collaborative ontology development.** The system is based on a large nationwide collaboratively maintained infrastructure of ontologies and ontology services.
6. **Collaborative metadata creation.** The system includes a model of and tools for collaborative semantic content creation.
7. **Machine semantics and services.** The services can be made available for machines.
8. **Large collaboration network.** The system has been developed on a national level with contents from over 20 memory organizations, and includes content from many international sources.
9. **Multilinguality.** Although the contents are in Finnish, the system can be used in two other languages.

Two user evaluation studies concerning semantic recommendations of an earlier version of CultureSampo have been performed with some promising results [11]. However, end user evaluation of the interfaces of the new prototype with its various features has not yet been done.

The portal is scalable in terms of its Web 2.0 content creation model and different types of content. An explicit concern in implementing the portal has been the computational efficiency in terms of speed and memory consumption, since a national level cultural heritage portal, if successful, will have lots of content inside and a large number of simultaneous users. For this goal, our earlier memory- and Prolog-based search engine used in MuseumFinland was replaced by conventional search engine technology, Apache Lucene, that was configured to do semantic search using semantic indexing. The CultureSampo search engine has been tested with a knowledge base of 20 million objects resulting in response times of less than 2 seconds on ordinary PC hardware.

Acknowledgements

Several researchers, including, Airi Hortling, Jouni Hyvönen, Ellen Karhulampi, Suvi Kettula, Helena Mäkimattila, and Jari Väätäinen, have contributed to developing CultureSampo contents and the related infrastructural components. This work is a part of the national FinnONTO research project 2003–2007, 2008–2010, funded mainly by Tekes and a consortium of 38 companies and public organizations. The work is also partly funded by the EU FP7 SmartMuseum project 2008–2010, and the Finnish Cultural Foundation (2008–2010).

References

- Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L.: Sweetening ontologies with DOLCE. In: Proc. of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, Springer–Verlag (2002)
- Hyvönen, E., Ruotsalo, T., Häggström, T., Salminen, M., Junnila, M., Virkkilä, M., Haaramo, M., Kauppinen, T., Mäkelä, E., Viljanen, K.: CultureSampo—Finnish culture on the semantic web. The vision and first results. In: Klaus Robering (Ed.), Information Technology for the Virtual Museum. LIT Verlag. (2008a)
- 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 (2008b)

- 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)
- Hyvönen, E., Takala, J., Alm, O., Ruotsalo, T., Mäkelä, E.: Semantic Kalevala—accessing cultural contents through semantically annotated stories. In: Proceedings of the Cultural Heritage on the Semantic Web Workshop at the 6th International Semantic Web Conference (ISWC 2007), Busan, Korea. (2007)
- Hyvönen, E.: Semantic portals for cultural heritage. In: Handbook of Ontologies, 2. edition, Springer-Verlag (March 2009).
- Hyvönen, E., Mäkelä, E.: Semantic autocompletion. In: Proceedings of the First Asia Semantic Web Conference (ASWC 2006), Beijing, Springer-Verlag (2006)
- Hyvönen, E., Mäkelä, E., Salminen, M., Valo, A., Viljanen, K., Saarela, S., Junnila, M., Kettula, S.: Museum-Finland—Finnish museums on the semantic web. *Journal of Web Semantics* 3(2) (2005) 224–241
- Hyvönen, E., Junnila, J., Kettula, S., Mäkelä, E., Saarela, S., Salminen, M., Syreeni, A., Valo A., and Viljanen, K.: Finnish Museums on the Semantic Web. User's Perspective on MuseumFinland. Proceedings of Museums and the Web 2004 (MW2004), Archives & Museum Informatics (2004).
- Junnila, M., Hyvönen, E., Salminen, M.: Describing and linking cultural semantic content by using situations and actions. In: Klaus Robering (Ed.), *Information Technology for the Virtual Museum*. LIT Verlag. (2008)
- Kauppinen, T., Väättänen, J., Hyvönen, E.: Creating and using geospatial ontology time series in a semantic cultural heritage portal. In: Proceedings of the ESWC 2008, Tenerife, Spain, Springer-Verlag (2008)
- Kurki, J., Hyvönen E.: Relational semantic search: Searching social paths on the semantic web. In: Poster Proceedings of the ISWC + ASWC 2007, Busan, Korea. (2007)
- Mäkelä, E., Viljanen, K., Alm, O., Tuominen, J., Valkeapää, O., Kauppinen, T., Kurki, J., Sinkkilä, R., Käsälä, T., Lindroos, R., Suominen, O., Ruotsalo, T., Hyvönen, E.: Enabling the Semantic Web with Ready-to-Use Web Widgets. In Nixon, L., Cuel, R., Bergamini, C., eds.: *First Industrial Results of Semantic Technologies, proceedings, co-located with ISWC 2007 + ASWC 2007, Busan, Korea. (2007) CEUR Workshop Proceedings, Vol 293.*
- Mäkelä, E., Suominen, O., Hyvönen, E.: Automatic exhibition generation based on semantic cultural content. In: Proc. of the Cultural Heritage on the Semantic Web Workshop at ISWC + ASWC 2007. (2007)
- Ruotsalo, T., Hyvönen, E.: An event-based approach for semantic metadata interoperability. In: Proceedings of the ISWC 2007 + ASWC 2007, Busan, Korea, Springer-Verlag (2007b)
- Schreiber, G., Amin, A., van Assem, M., de Boer, V., Hardman, L., Hildebrand, M., Hollink, L., Huang, Z., van Kersen, J., de Niet, M., Omelayenko, B., van Ossenbruggen, J., Siebes, R., Taekema, J., Wielemaker, J., Wielinga, B.J.: MultimediaN e-culture demonstrator. In: *The Semantic Web – Proceedings of the 5th International Semantic Web Conference 2006. (2006) 951–958*
- Staab S., Studer R. (eds.) *Handbook on Ontologies*. Springer-Verlag, March 2009 (2. edition).
- Sheth, A., Aleman-Meza, B., Arpinar, I.B., Bertram, C., Warke, Y., Ramakrishnan, C., Halaschek, C., Anyanwu, K., Avant, D., Arpinar, F.S., Kochut, K.: Semantic association identification and knowledge discovery for national security applications. *Journal of Database Management on Database Technology* 16(1) (Jan–March 2005) 33–53
- Sowa, J.: *Knowledge Representation. Logical, Philosophical, and Computational Foundations*. Brooks/Cole (2000)
- Valkeapää, O., Alm, O., Hyvönen, E.: Efficient content creation on the semantic web using metadata schemas with domain ontology services. In: Proceedings of ESWC 2007, Innsbruck, Austria, Springer-Verlag (2007)
- van Assem, M., Malaise, V., Miles, A., Schreiber, G.: A method to convert thesauri to SKOS. In: Proceedings of the Third European Semantic Web Conference (ESWC'06), Springer-Verlag (2006)

- Viljanen, K., Tuominen, J., Hyvönen, E.: Distributed semantic content creation and publication for cultural heritage legacy systems. In: Proceedings of the 2008 IEEE International Conference on Distributed Human- Machine Systems, Athens, Greece, IEEE Press (2008)
- Viljanen, K., Tuominen, J., Käsälä, T., Hyvönen, E.: Distributed semantic content creation and publication for cultural heritage legacy systems. In: Proceedings of the 2008 IEEE International Conference on Distributed Human- Machine Systems, Athens, Greece, IEEE Press (2008)
- Wang, Y., Stash, N., Aroyo, L., Gorgels, P., Rutledge, L., Schreiber, G.: Recommendations based on semantically-enriched museum collection. *Journal of Web Semantics* (2008)