TKK Dissertations 249
Espoo 2010

VIEW-BASED USER INTERFACES FOR THE SEMANTIC WEB

Doctoral Dissertation

Eetu Mäkelä

Aalto University
School of Science and Technology
Faculty of Information and Natural Sciences
Department of Media Technology

TKK Dissertations 249
Espoo 2010

VIEW-BASED USER INTERFACES FOR THE SEMANTIC WEB

Doctoral Dissertation

Eetu Mäkelä

Doctoral dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Faculty of Information and Natural Sciences for public examination and debate in Auditorium TU2 at the Aalto University School of Science and Technology (Espoo, Finland) on the 26th of November 2010 at 12 noon.

Aalto University
School of Science and Technology
Faculty of Information and Natural Sciences
Department of Media Technology

Aalto-yliopisto
Teknillinen korkeakoulu
Informaatio- ja luonnontieteiden tiedekunta
Mediatekniikan laitos

Distribution:
Aalto University
School of Science and Technology
Faculty of Information and Natural Sciences
Department of Media Technology
P.O. Box 15500 (Otaniementie 17)
FI - 00076 Aalto
FINLAND
URL: http://www.media.tkk.fi/
Tel. +358-9-47001
E-mail: eetu.makela@tkk.fi

ISBN 978-952-60-3446-1
ISBN 978-952-60-3447-8 (PDF)
ISSN 1795-2239
ISSN 1795-4584 (PDF)
URL: http://lib.tkk.fi/Diss/2010/isbn9789526034478/

TKK-DISS-2833

Aalto-Print
Helsinki 2010

Aalto University

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	
☐ Monograph	☒ 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

Abstract

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 built semantic user interfaces. Designing the actual systems to be as adaptable as possible is also discussed.

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

☒ The dissertation can be read at http://lib.tkk.fi/Diss/2010/isbn9789526034478/

Aalto-yliopisto

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	
☐ Monografia	☒ Yhdistelmäväitöskirja (yhteenveto + 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

Tiivistelmä

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 moninäkymähakuportaaleja eri käyttötarkoituksiin ja aineistoille. Saadut tulokset osoittavat että moninäkymähakuparadigma 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.

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	
☒ Luettavissa verkossa osoitteessa http://lib.tkk.fi/Diss/2010/isbn9789526034478/	

Contents

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: 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.

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, 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.

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 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 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 FinnONTO[1] 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 a 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

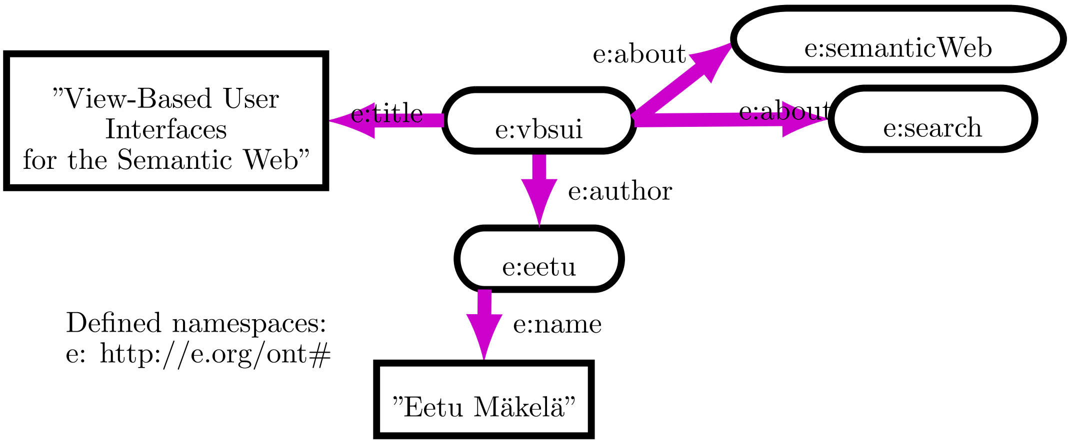

Figure 1.1: A visualization of an example RDF network

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:eetu" 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.

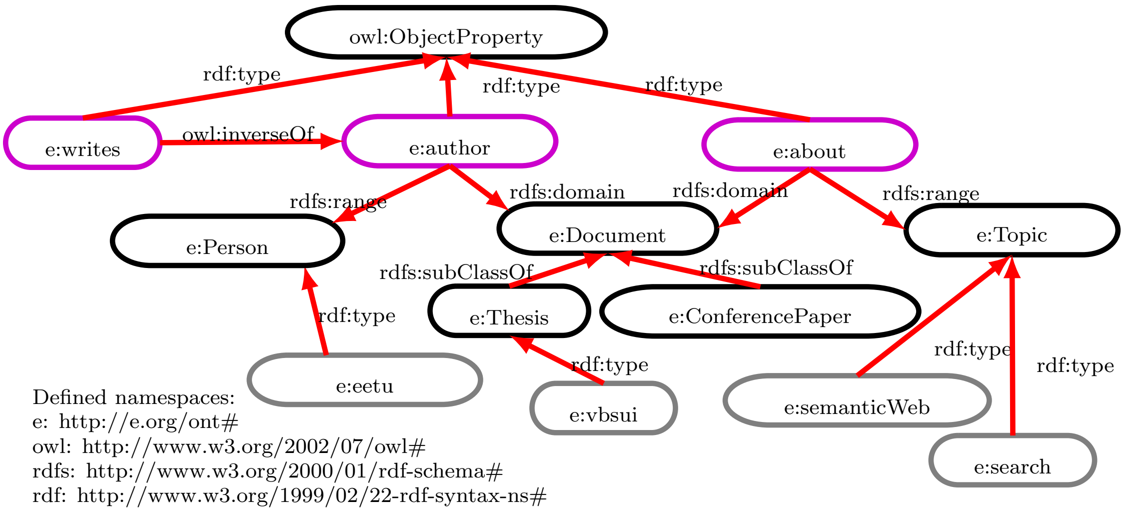

Figure 1.2: An example of an ontology

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 MuseumFinland[2] (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 CultureSampo[3] (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.

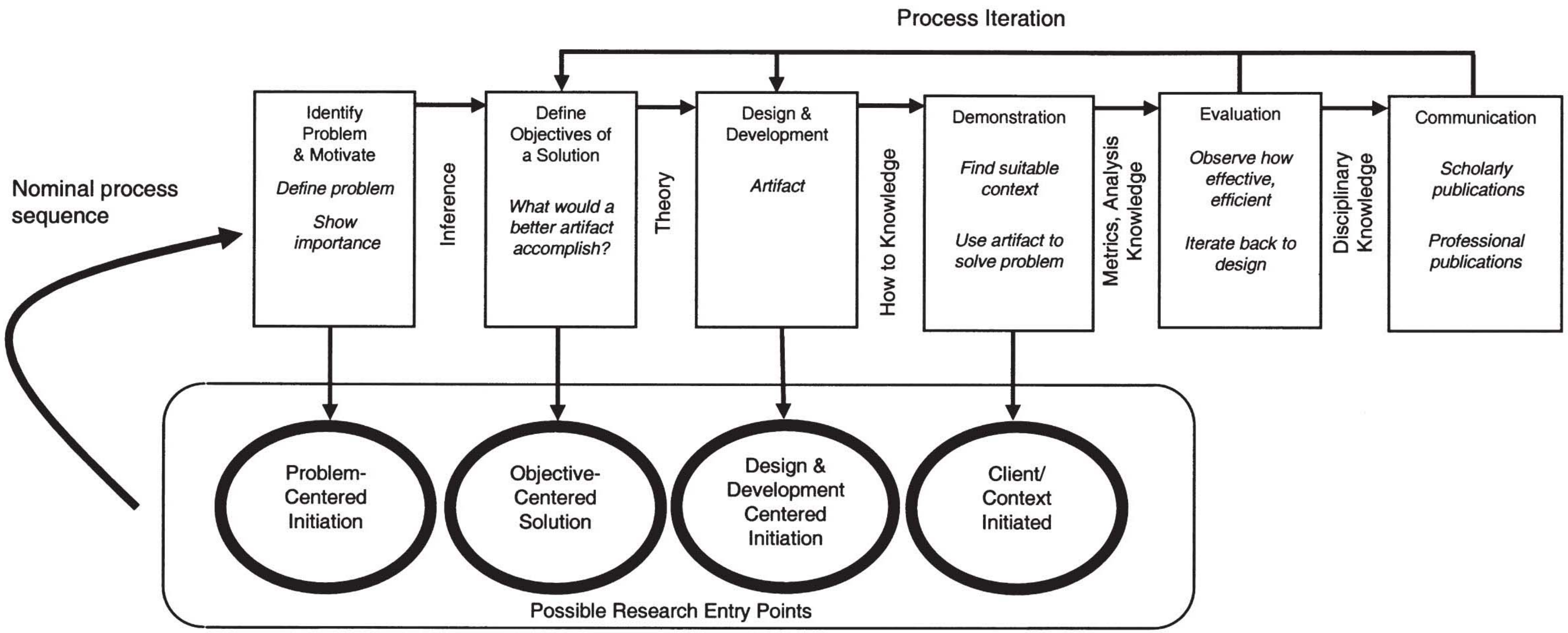

Figure 1.3: A process model of design science research methodology [58]

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 directory[4] and the Yahoo! directory[5], 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.

Athanasis 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 → creates → 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 → creates → 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.

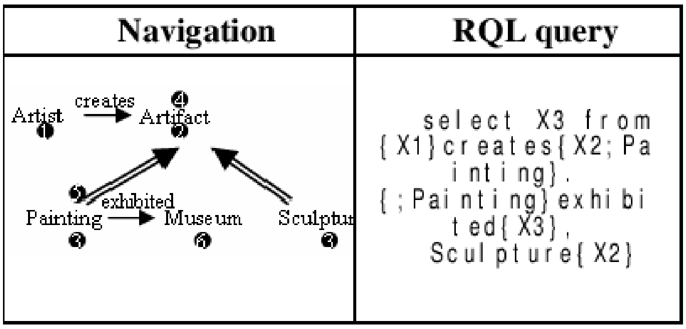

Figure 2.1: A visual formulation of a query in the GRQL interface, along with the generated query language expression [5]

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.

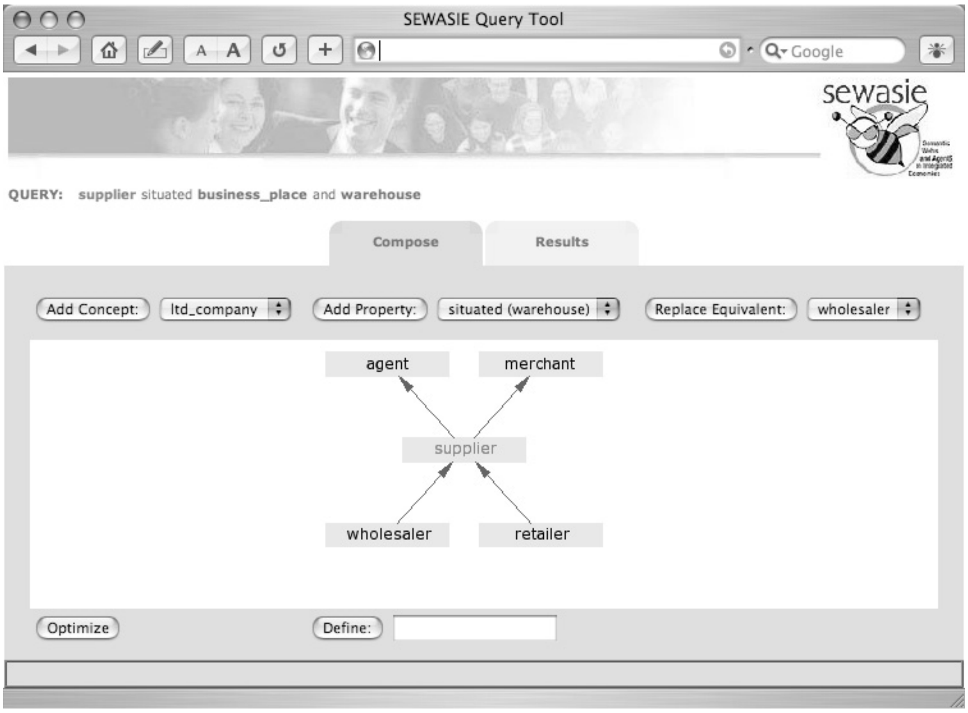

Figure 2.2: The SEWASIE visual tool for query formulation support [11]

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

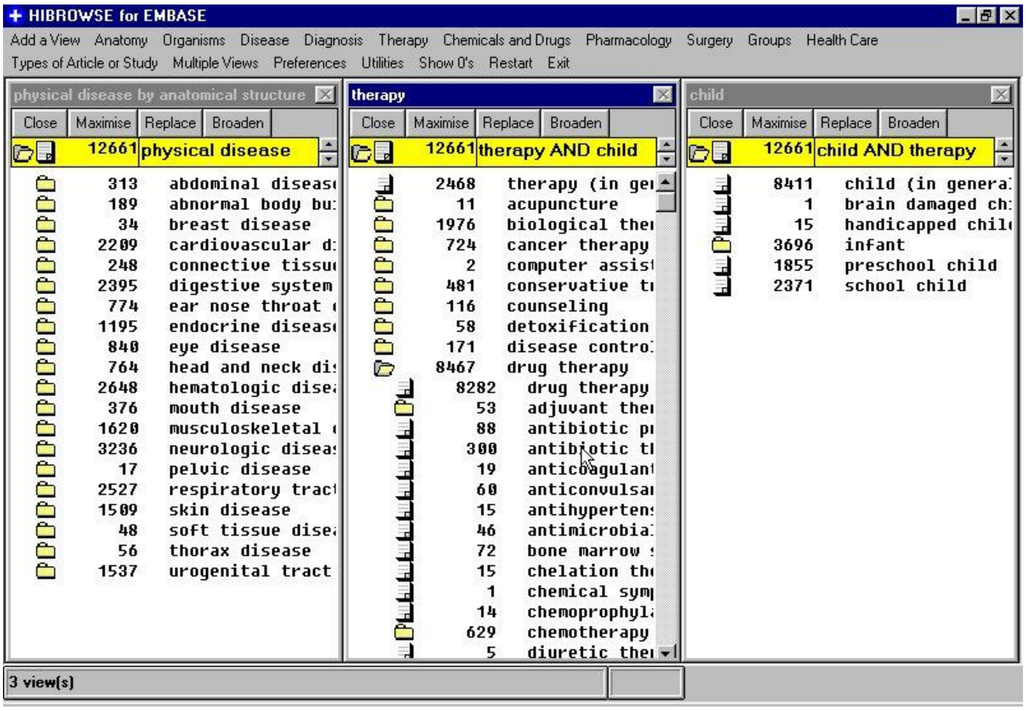

Figure 3.1: The HiBrowse interface, with three hierarchical views [60]

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.

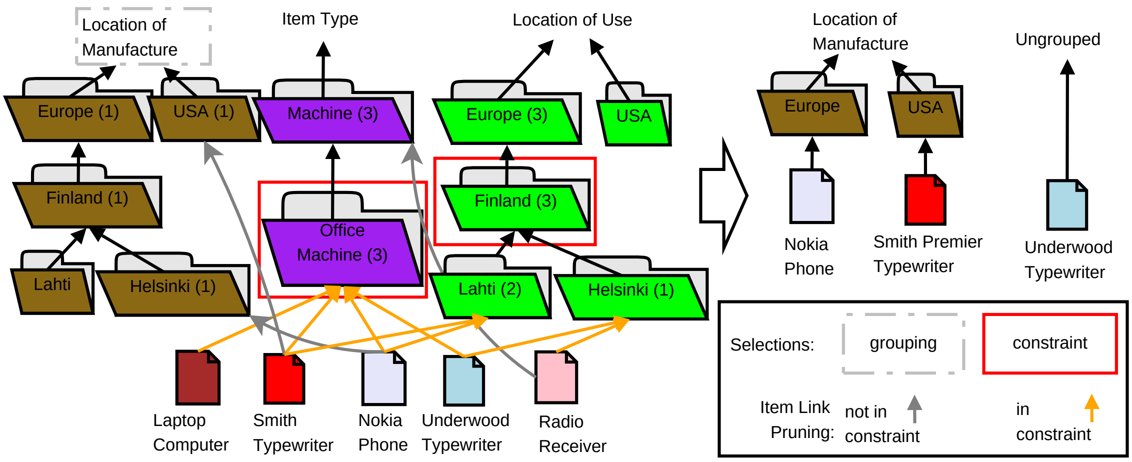

Figure 3.2: A conceptual overview of view-based querying

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

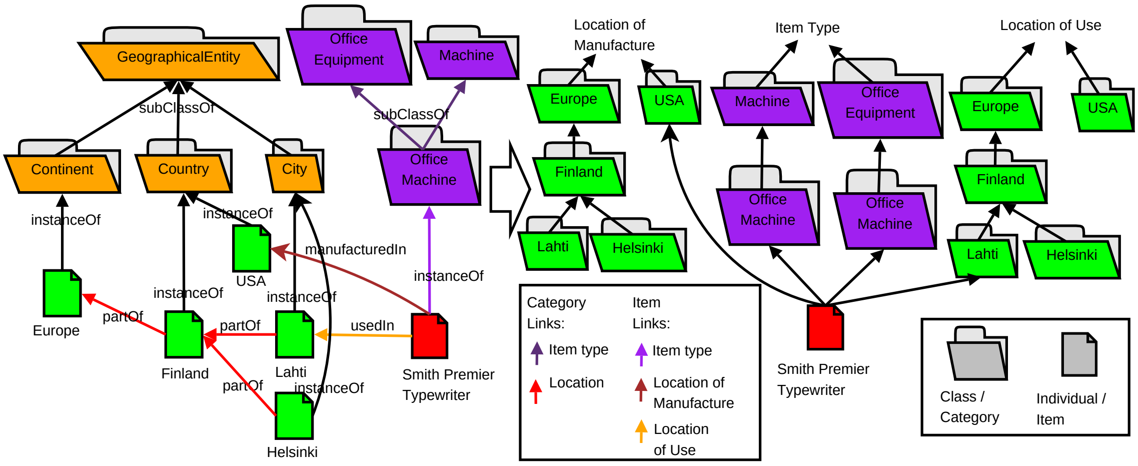

Figure 3.3: An example of view projection

"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 Autocompletion

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 autocompletion was developed for this purpose, and its combination with view-based search studied.

The different forms of semantic autocompletion developed are presented exhaustively in paper IV. Shortly, the idea of autocompletion 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 autocompletion then is to extend traditional syntactic autocompletion to take into account semantic information.

For example, syntactic autocompletion 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.

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 from, 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

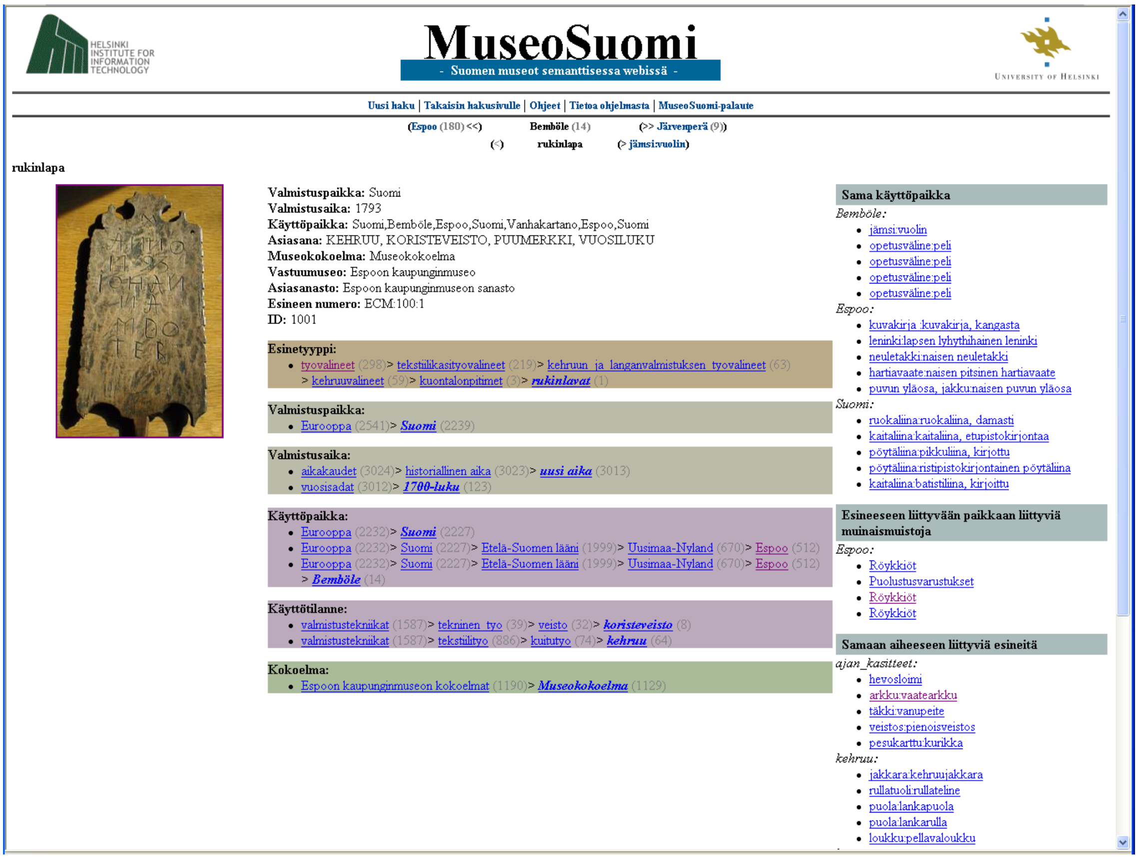

Figure 4.4: The item view of MuseumFinland

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" ("Sama 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 AJAX[6] techniques. This user interaction pattern tightly integrates keyword

[6] Asynchronous 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.

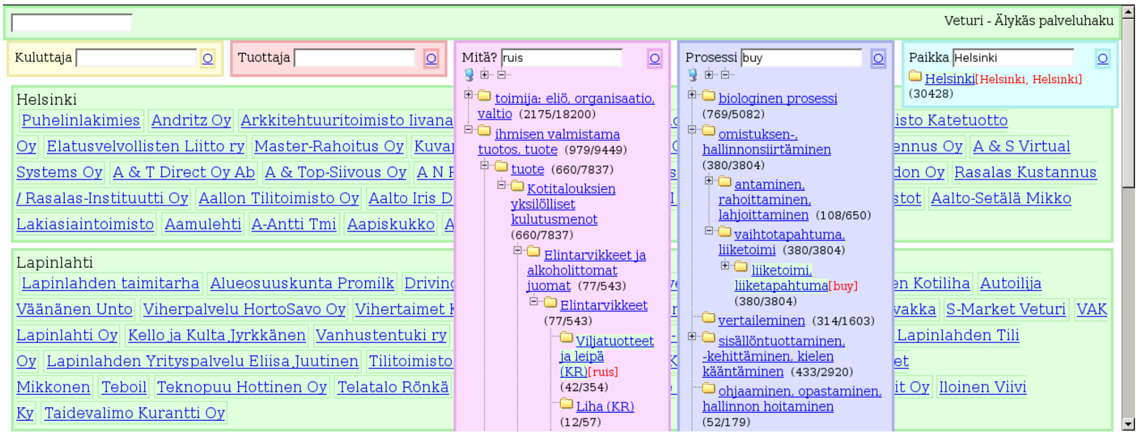

Figure 4.5: The Veturi user interface

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 eventually 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 students[7] 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 project[8] website directory (unpublished), and

8. link library data of the Suomi.fi[9] 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 portal[10], 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/
[10]Available 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 CultureSampo[11], 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,

[11] Portal available at http://www.kulttuurisampo.fi/

the schema for Finnish Museums Online[12], 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: <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 autocompletion 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.

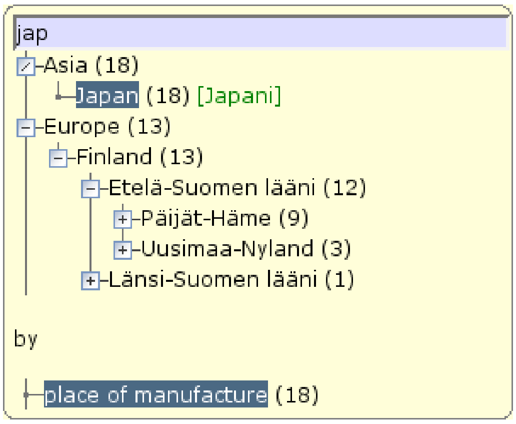

Figure 4.6: Domain Centric View Based Search

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.

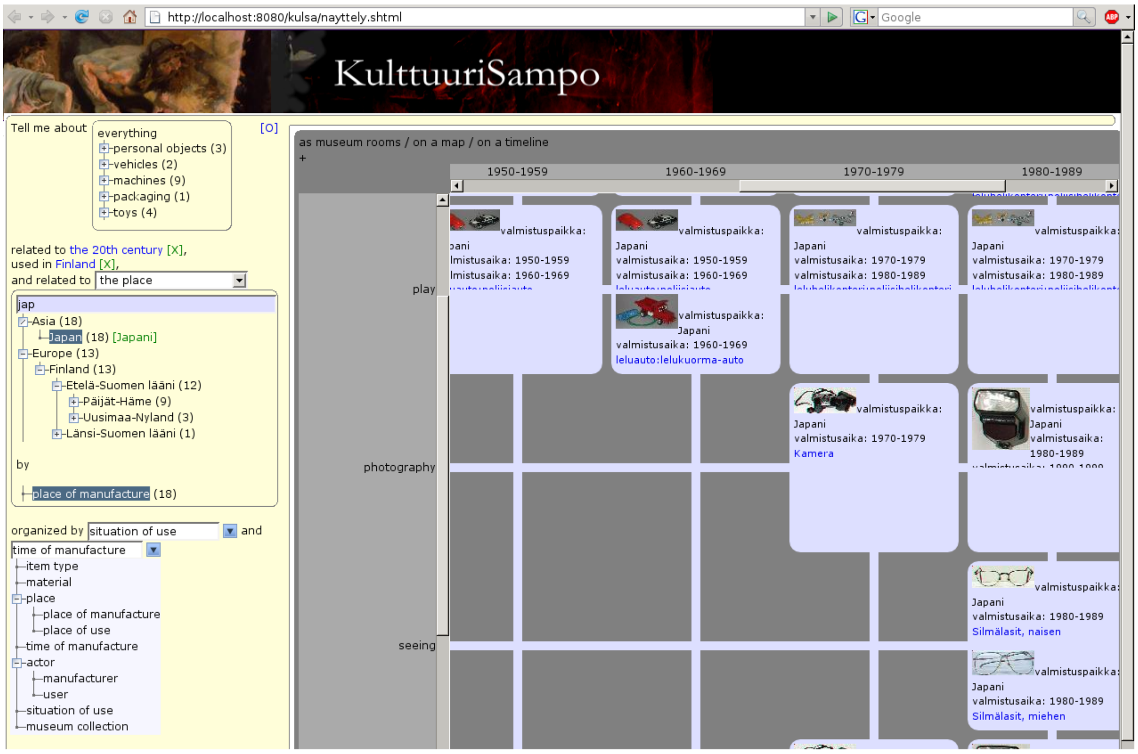

Figure 4.7: Exhibition room visualization in CultureSampo

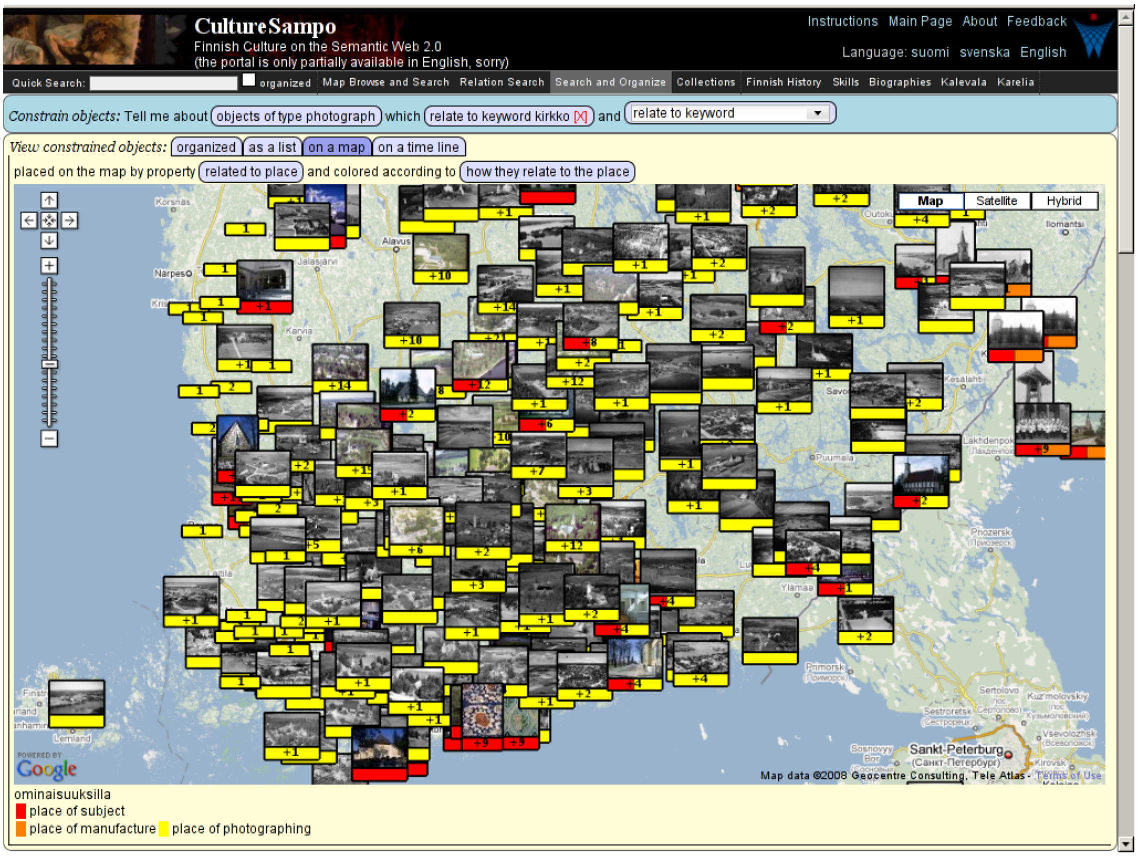

Figure 4.8: Map visualization in CultureSampo

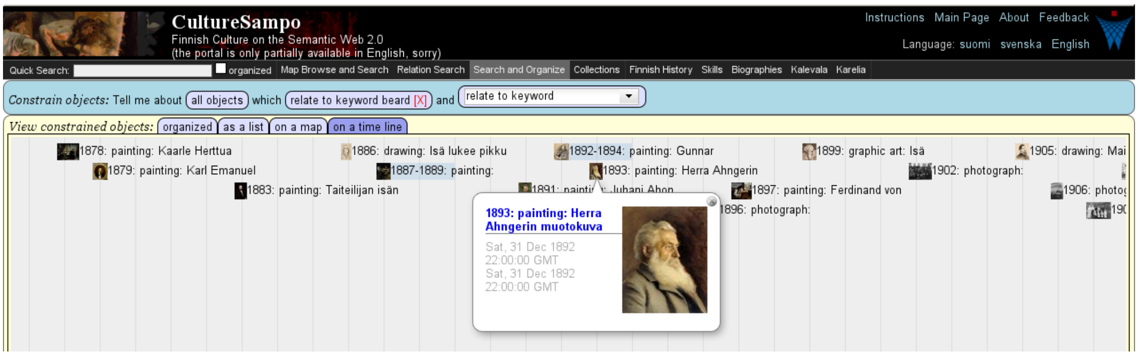

Figure 4.9: Timeline visualization in CultureSampo

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 CultureSampo 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 OntoViews[13] 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-based[14] 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

[13] The 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 Project[16] 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 browser[17] 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 Navigator[18] 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 Cocoon[19], 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 11^{th} 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, Massachusetss. 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. Informationg 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 Eureopean 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: IIiX: 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 Salminen, 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 Känsälä, 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 moninäkymähaun 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`.

[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 Känsälä 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 multimodal 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-goverment 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 Känsälä, 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.

ISBN 978-952-60-3446-1
ISBN 978-952-60-3447-8 (PDF)
ISSN 1795-2239
ISSN 1795-4584 (PDF)