MUSEUMFINLAND
—Finnish Museums on the Semantic Web

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

Helsinki Institute for Information Technology (HIIT),
University of Helsinki, and Helsinki University of Technology
P.O. Box 5500, 02015 TKK, FINLAND
`FirstName.LastName@cs.Helsinki.FI`
`http://www.cs.helsinki.fi/group/seco/`

Abstract

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

Key words: semantic web, information retrieval, multi-facet search, view-based search, ontology, recommendation system

1   Why Museums on the Semantic Web?

A special characteristic of cultural collection databases is that they contain semantically rich information. Collection items have a history and are related in many ways to our environment, to the society, and to other collection items. For example, a chair may be made of oak and leather, may be of a certain style, was designed by a famous designer, was manufactured by a certain company during a time period, was used in a certain building together with other pieces of furniture, and so on. Other collection items, locations, time periods, designers, companies etc. can be

related to the chair through their properties and implicitly constitute a complicated semantic network of associations. This semantic network is not limited to a single collection but spans over other related collections in other museums. The network of semantic associations can be extended to contents of other types in other organization, as well.

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

Collection interoperability in content Web languages, standards, and ontologies make it possible to make heterogeneous museum collections of different kind mutually interoperable. This enables, e.g., the creation of large inter-museum exhibitions.

Intelligent applications More versatile, user-friendly, and useful applications based on the semantics of the collections can be created.

To realize these ideas in practice, we have developed a semantic web portal called "MUSEUMFINLAND—Finnish Museums on the Semantic Web" [2] . This system contains an inter-museum exhibition of over 4,000 cultural artifacts, such as textiles, pieces of furniture, tools etc. Also metadata concerning some 260 historical sites in Finland were incorporated in the system. The goals for developing the system were the following:

Global view to distributed collections It is possible to use the heterogeneous distributed collections of the museums participating in the system as if the collections were in a single uniform repository.

Content-based information retrieval The system supports intelligent information retrieval based on ontological concepts, not on simple keyword string matching as is customary with current search engines.

Semantically linked contents A most interesting aspect of the collection items to the end-user are the implicit semantic relations that relate collection data with their context and to each other. In MUSEUMFINLAND, such associations are exposed dynamically to the end-user by defining them in terms of logical predicate rules that make use of the underlying ontologies and collection metadata.

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

Museum databases are usually situated at different locations and use different database systems and schemas. This creates a severe obstacle to information retrieval.

---

[1] http://www.w3.org/2001/SW/
[2] http://museosuomi.cs.helsinki.fi/

To address the problem, the web can be used for creating a single interface and access point through which a search query can be sent to distributed local databases and the results combined into a global hit list. This "multi-search" approach is widely applied and there are many cultural collection systems on the web based on it, such as the portals Australian Museums Online [3] and Artefacts Canada [4].

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

This paper presents MUSEUMFINLAND from different viewpoints [15, 13, 19, 18, 25]. The ontologies underlying the system are first discussed. After this we explain how content from the museum databases can be imported into the global RDF(S) [5] [21, 1] repository conforming to the shared ontologies. Next the semantic search and browsing services of MUSEUMFINLAND are explained from the end-user's viewpoint, and adaptation of the system to new data is discussed. Then we get down to the implementation and describe the general architecture underlying the system and its components. The paper concludes by discussing the lessons learned as well as related and future work.

2 Ontologies

MUSEUMFINLAND is based on seven domain ontologies:

(1) The Artifacts ontology (3227 classes) is a subclass taxonomy of tangible collection objects, such as pottery, cloths, weapons, etc. All artifacts in the system belong to some class in this ontology.
(2) The Materials ontology (364 classes) is a subclass taxonomy of the artifact materials, such as steel, silk, tree, etc.

---

[3] http://www.amonline.net.au/
[4] http://www.chin.gc.ca/
[5] http://www.w3.org/RDF/

(3) The Actors ontology (26 classes, 1715 instances) defines classes of agents, such as persons, companies etc., and individuals as instances of these classes.

(4) The Situations ontology (992 classes) is a taxonomy that includes intangible happenings, situations, events, and processes that take place in the society, such as farming, feasts, sports, war, etc.

(5) The Locations ontology (33 classes, 864 instances) represents areas and places on the Earth. It contains classes such as Continent, Country, County, City, Farm etc. The main content in the ontology is its individual location instances (e.g., Helsinki or Finland) and their mutual meronymy relations (e.g., Helsinki is a part of Finland).

(6) The Times ontology (57 classes) is a meronymy of various predefined historical periods. First, there are categories representing special eras of interest such as the Middle Ages and the time of the World War II. Second, there is a linear breakdown hierarchy of centuries and decennia. The properties of time concepts are a human readable label of period and the beginning and end year of the time interval.

(7) The Collections ontology (22 classes, 24 instances) is a taxonomy that classifies the collections included in the portal under the museums hosting them. The properties of the taxonomy indicate the name and the hosting museum of the collection.

In Finland, the most notable and widely used thesaurus for cultural content in Finnish is MASA [23] maintained by the National Board of Antiquities [6]. MASA consists of some 6000 terms and employs the usual thesaurus relations [5], such as NT (narrower term), BT (broader term), and RT (related term). In our work, MASA thesaurus was transformed into an RDF Schema ontology called MAO by first creating an RDF Schema structure from the MASA database. This initial ontology was then enhanced and edited as a Protégé-2000 project by hand. In this way, three domain ontologies, Artifacts, Materials, and Situations emerged as sub-ontologies of MAO. These ontologies were later on extended based on collection item data from the collections of the National Museum [7], Espoo City Museum [8], and Lahti City Museum [9].

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

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

- From the end-user's view-point, it enables semantic link generation and semantic browsing. This feature will be discussed in detail in the coming sections.
- From the cataloger's view-point, it makes the cataloging process simpler because many additional annotations can be automatically created. For example, if we know that the artifact is a doctor's hat, then there is no need to tell that it is related to academic ceremonies, because this inference can be drawn by a simple rule.

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

A few simple explicit association types of form *X isRelatedTo Y* were identified. Firstly, we envisioned that the events taking place in the society, i.e., classes in the Situations ontology, are of central importance for creating useful semantic linkage. Therefore additional association triples of form $(artifact, is-related-to-event, situation)$ were created. These relations were defined by a museum curator with the user-friendly N3-notation [10] . For example:

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

Secondly, artifacts are related to each other, which can be represented by the triple $(artifact_1, is-related-to-artifact, artifact_2)$. For example, sailing ships are related to sails, screw drivers to screws, etc. Thirdly, there are association between artifacts and materials. Altogether, 301 different associations between ontology classes were created in this way.

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

---

[10] http://www.w3.org/DesignIssues/Notation3.html

3 Content creation process

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

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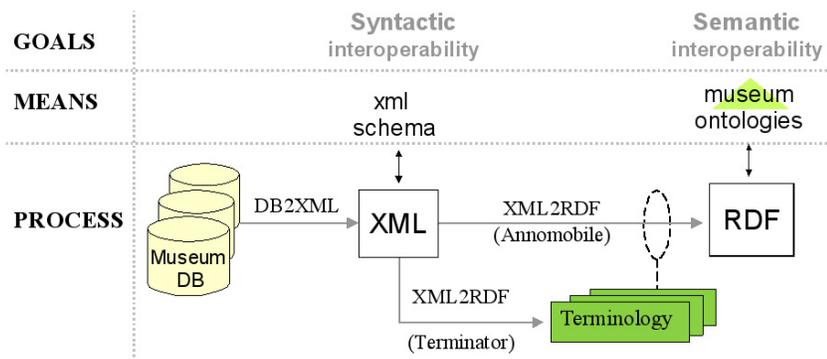

Fig. 1. The content creation process in MUSEUMFINLAND.

The first step in combining the heterogeneous databases is to gain syntactic inter-

operability by transforming database contents into a shared XML format. Based on the schema, each collection item has an XML description of its own, the XML card. For example, the XML card representing a calendar is presented below [11] :

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

An XML card presents the main features of a collection object by sub-elements. The values of the features, such as the string "Espoo City Museum" in the sub-element `<museum>`, are read from the underlying database tables.

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

For example, the XML card presented in page **??** would translate into the RDF card below:

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

---

[11] The example is translated and slightly simplified from the original version in Finnish.

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

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

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

4 End-user's perspective

MUSEUMFINLAND provides the end-user with two services:

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

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

Each category is related to a set of search objects that we will call its *projection*. The *extension E* of a category is the union of its projection *P* and the extensions of its subcategories S_i: $E = P \cup S_1 \cup S_2 \cup ... \cup S_n$. A search query in multi-facet view-based search is formulated by selecting categories of interest from the different facets, typically one selection from a facet. The answer to the query is simply the intersection of the extensions E_i of the selected categories: $A = \cap\{E_i\}$. For example, by selecting the category "Chairs" from the Artifact facet, and "Helsinki" from the Place of Manufacturing facet, the user can express the query for retrieving all chairs (of any subtype) manufactured in Helsinki (or in any of its suburbs and other locations within Helsinki).

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

Facets can be used for helping the user in information retrieval in many ways. First, the facet hierarchies give the user an overview of what kind of information there is in the repository. Second, the hierarchies can guide the user in formulating the query in terms of appropriate concepts. Third, the hierarchies do not suffer from the problems of homonymous query terms. Fourth, the facets can be used as a navigational aid when browsing the database content [9]. Fifth, the number of hits in every category that can be selected next can be computed *beforehand* and be shown to the user [28]. In this way, the user can be hindered from making a selection leading to an empty result set—a recurring problem in information retrieval systems—and is guided toward selections that are likely to constrain (or relax) the search appropriately.

Figure 2 shows the search interface of MUSEUMFINLAND. The nine facets are shown on the left (in Finnish), such as Artifact type ("Esinetyyppi") and Material ("Materiaali"). For each facet, the next level of sub-categories is shown as a set of links. A query is formulated by selecting a category by clicking on its name. When the user selects a category c in a facet f, the system constrains the search by leaving in the result set only such objects that are annotated in facet f with some sub-category of c or c itself. Here the user have selected the sub-category

Tools ("työvälineet") from the Artifact type facet ("Esinetyyppi"), and the result set is seen on the right grouped by the sub-categories of Tools, such as Textile making tools ("tekstiilityövälineet") and Tools of folk medicine ("kansanlääkinnän työvälineet"). Hits in different categories are separated by horizontal bars and can be viewed page by page independently in each category. The number of hits shown in each sub-category is determined from the number of sub-categories in the result set in order to maximize useful information on the limited screen space. In this case, all subcategories do not fit on the screen, and only a single line of hits is shown for each subcategory.

Fig. 2. MUSEUMFINLAND search interface after selecting the category link Tools ("työvälineet").

The user can refine the query further by selecting another category on the left. For example, assume that the user selects category Farming and cattle tending ("maatalous ja karjanhoito") in the view Situation of usage ("Käyttötilanne") in figure 2. Three things happen when answering the query. First, the result set on the right is refined to the intersection of previous selections; here the result is tools used in farming and cattle tending. Second, the selected view is changed to expose the sub-

10

categories of the selected category. Third, the size n of the result set resulting from the selection of any category link seen on the screen is recomputed proactively, and a number (n) is shown to the user after each category name. This number tells that if the category is selected next, then there will be n hits in the result set. For example, in figure 2, the number 193 in the Collection facet ("Kokoelma") on the bottom tells that there are 193 tools in the collections of the National Museum ("Kansallismuseon kokoelmat"). A selection leading to an empty result set ($n = 0$) is removed from the facet (or alternatively disabled and shown grayed out, depending on the user's preference). In this way, the user is hindered from making a selection leading to an empty result set, and is guided toward selections that are likely to constrain the search appropriately. The query can be relaxed by making a new selection on a higher level in the facet or by dismissing the facet totally from the query.

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

View-based search is not a panacea for information retrieval. Google-like keyword search interface is usually preferred [4] if the user is capable of expressing her information need terms of accurate keywords. MUSEUMFINLAND seamlessly integrates this functionality with view-based search in the following way: First, the search keywords are matched against category names in the facets in addition to text fields in the metadata. A new dynamic facet is created in the user interface. This facet contains all facet categories whose name (or other property values) matches the keyword. Intuitively these facet categories tell the different interpretations of the keyword, and by selecting one of them next the right choice can be made. This approach also solves the search problem of finding relevant categories in facets that contain thousands of categories. Second, a result set of object hits is shown. This result set contains all objects contained in any of the categories matched in addition to all objects whose metadata directly contains the keyword. The hits are grouped by the categories found.

At any point during multi-facet search the user can select any hit found by clicking on its image. The corresponding data object is then shown as a web page, such as the one in figure 3. The example depicts a special part, distaff ("rukinlapa" in Finnish) used in a spinning wheel. The page contains the following information and links:

(1) On top, there are links to directly navigate in the groups and results of the current query.
(2) The image(s) of the object is (are) depicted on the left.

(3) The metadata of the object is shown in the middle on top.
(4) All facet categories that the object is annotated with are listed in the middle bottom as hierarchical link paths. A new search can be started by selecting any category there.
(5) A set of semantic links to related artifacts is shown on the right.

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

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

5 Adapting services for new content
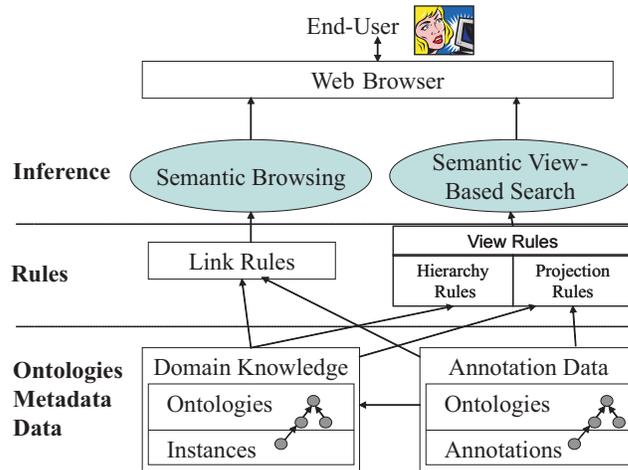
Fig. 4. Architecture of MUSEUMFINLAND on the server side.

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

A technical innovation of MUSEUMFINLAND is to introduce an intermediate mapping layer of logical rules between the content and semantic services: Link Rules for the browsing service and View Rules for the search engine. By using the rules the generic service engines can be separated from domain and annotation specific details and be adapted to contents of different kind by changing the rules only. The rules are defined declaratively in terms of Prolog predicates operating on RDF triples as in [12].

In the following, the idea of View Rules and Link Rules is described in more detail by using examples. We use SWI-Prolog [12] as the inference engine and SWI-Prolog syntax in the examples [13].

A view is a hierarchical index-like decomposition of category resources where each category is associated with a set of subcategories and a set of directly related data items. A view is defined in terms of ontologies by specifying a view rule predicate called `ontodella_view`. It contains the following information: 1) the root resource URI, 2) a *hierarchy rule* defined by a binary subcategory relation predicate, 3) a

binary *projection rule* predicate that maps search objects onto the view categories, and 4) a label for the view. An example of a view rule predicate is given below:

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

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

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

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

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

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

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

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

Hierarchy rules tell how the views are projected logically. A separate question is

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

Link Rules (cf. figure 4) are used for creating semantic recommendation links, such as those in figure 3. Such links can be created in various ways [29]. In our work, we employed the idea of *rule-based recommendations* where the domain specialist explicitly describes the notion of "interesting related resource" with generic logic rules. The system then applies the rules to the underlying knowledge base in order to find interesting resources related to the selected one.

This method has several strengths. Firstly, the rule can be associated with a label, such as "Other artifacts used in event x", that can be used as the explanation for the recommendations found. It is possible to deduce the explanation label as a side effect of applying the rule. Secondly, since semantic linking rules are described by the domain specialist, the rules and explanations are explicitly defined and are not based on heuristic measures, which could be difficult to understand and motivate. The specialist knows the domain and may promote the most important relations between the resources. However, this could also be a weakness if the user's goals and the specialists thoughts about what is important do not match, and the user is not interested in the recommendations. Thirdly, the rule-based recommendations do not exclude the possibility of using other recommendation methods but provides an infrastructure for applying any rules. For example, the recommendation rules could perhaps be learned or tuned by observing the users actions.

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

6 Architecture and implementation

MUSEUMFINLAND has been implemented by using a tool called ONTO-
VIEWS [14] [25]. This tool was developed during the project and has later been ap-
plied to creation of other semantic portals as well [22, 25]. ONTOVIEWS consists
of the three major components shown in figure 5:

(1) The logic server ONTODELLA provides the system with reasoning services,
    such as category view projection and dynamic semantic link recommenda-
    tions.
(2) The search engine ONTOGATOR is a generic view-based RDF search engine,
    responsible for the multi-facet search functionality of the system.
(3) The third component ONTOVIEWS-C binds the services of ONTOGATOR and
    ONTODELLA together, and provides the user interfaces.
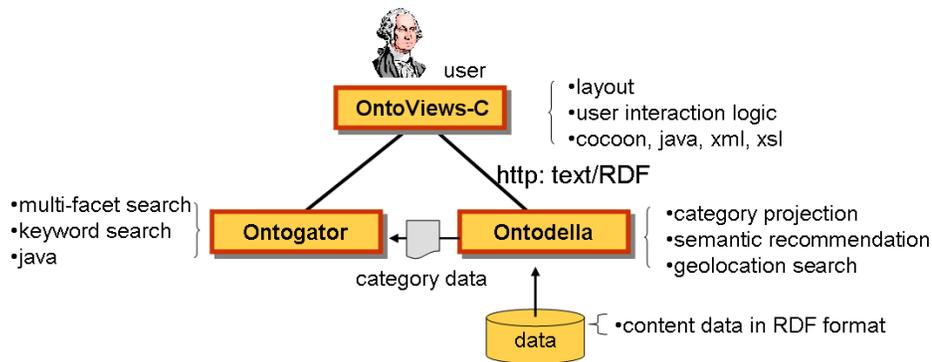
Fig. 5. The components of ONTOVIEWS.

ONTODELLA is a logic engine for defining and executing the View and Link Rules
of figure 4. ONTODELLA is a multi-threaded web server which provides remote
access to execute the rules in the framework. The web server and the rule execu-
tion framework are written using SWI Prolog [15] and its readily available HTTP
libraries. For the mobile user interface, ONTODELLA has been extended to provide
simple point-of-interest search based on geo-coordinates available from the mobile
phone.

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

---

[14] The software is available at http://www.cs.helsinki.fi/group/seco/museums/dist/ in open
source.

[15] http://www.swi-prolog.org

View creation service provides necessary hooks for executing the hierarchy and projection predicates. The view creation algorithm traverses the ontologies by using the given predicates dynamically in a depth-first search. The resulting view structure is serialized in RDF/XML according to a model derived from the Annotea Bookmark Schema [16]. This structure is used by ONTOGATOR as the basis for the view-based search.

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

$$predicate(SubjectURI, TargetURI, Explanation)$$

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

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

```
related_by_event( Subject, Target, Explanation ) :-

ItemTypeProperty =
  'http://www.cs.helsinki.fi/seco/ns/2004/03/artifacts#item_type',
ItemTypeToEventRelatingProperty =
  'http://www.cs.helsinki.fi/seco/ns/2004/03/mapping#related_to_event',

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

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

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

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

% don't make uninteresting links between items of the same type
```

---

[16] http://www.w3.org/2003/07/Annotea/BookmarkSchema-20030707

```
SuperClassOfTargetItemType \= SubjectItemType,
not(rdfs_transitive_subClassOf(SuperClassOfTargetItemType,
SubjectItemType)), not(rdfs_transitive_subClassOf(SubjectItemType,
SuperClassOfTargetItemType)),

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

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

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

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

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

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

ONTOGATOR defines and implements an RDF-based query interface that is used to separate view-based search logic from the user interface. The interface is defined as an OWL [17] ontology [18], and is based on selectors that can be used to query for both view category hierarchies and the projection resources of their categories based on various criteria, such as category, keyword, and geolocation-based constraints. The

[17] http://www.w3.org/OWL/
[18] http://www.cs.helsinki.fi/group/seco/ns/2004/03/ontogator#

query is represented in XML/RDF form.

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

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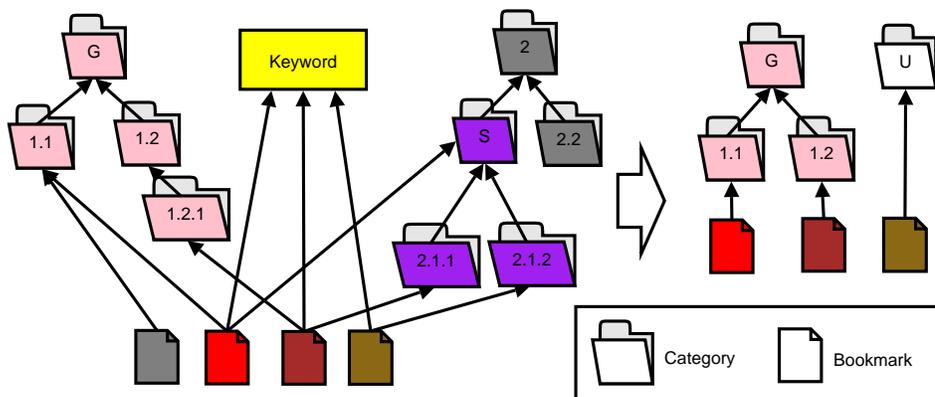
Fig. 6. A keyword plus category selector search with results grouped into an independant, partially cut hierarchy

ONTOVIEWS-C is the user interface, interaction and control component of ONTO-VIEWS (cf. figure 5). It is built on top of the Apache Cocoon framework [19] . Cocoon is a framework based wholly on XML and the concept of pipelines constructed from different types of components. A pipeline always begins with a generator, that generates an XML-document. Then follow zero or more transformers that take an XML-document as input and output a document of their own. The pipeline always ends in a serializer that serializes its input into the final result, such as an HTML-page, a PDF-file, or an image. It is also possible for the output of partial pipelines to be combined via aggregation into a single XML-document for further processing. Execution of these pipelines can be tied to different criteria, e.g, to a combination of the request URI and requesting user-agent.

---

[19] http://cocoon.apache.org/

7   Discussion

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

Exact definitions By using ontologies, the museums can define the concepts used in cataloging in a precise, machine understandable way.

Terminological interoperability The terms used in different institutions can be made mutually interoperable by mapping them onto common shared ontologies. The ontologies are not used as a norm for telling the museums what terms to use, but rather to make it possible to tolerate terminological variance as far as the terminology mapping from the local term conventions to the global ontology is provided.

Ontology sharing Ontologies provide means for making exact references to the external world. For example, in MUSEUMFINLAND, the location ontology (villages, cities, countries, etc.) and the actor ontology (persons, companies, etc.) is shared by the museums in order to make the right and interoperable references. For example, two persons who happen to have the same name should be disambiguated by different URIs, and a person whose name can be written in many ways, should be identified by a single URI to which the alternative terms refer.

Automatic content enrichment Ontological class and individual definitions, cultural and common sense rules, view projection rules, semantic linking rules, and consolidated metadata enrich collection data semantically.

Intelligent services Ontologies can be used as a basis for intelligent services to the end-user. In MUSEUMFINLAND, the view-based multi-facet search engine is based on the underlying ontological structures and the semantic link recommendation system reveals to the end-user the underlying semantical context of the collection items and their mutual relations.

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

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

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

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

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

---

[20] http://www.dmoz.org/

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

The idea of linking collection items with semantic associations was inspired by Topic Maps [26]. However, in our case the links are not given by a map but are determined by logical inference using the underlying RDF(S) ontology and RDF metadata. Another application of this idea to generating semantically linked static HTML sites from RDF(S) repositories is presented in [12]. Logic and dynamic link creation on the semantic web has been discussed, e.g. in the work on Open Hypermedia [6, 3], and in the Promootori system [17]. In the HyperMuseum [32], collection items are also semantically linked with each other. Here linking is based on shared words in the metadata and their linguistic relations, such as synonymy and antonymy. In contrast, our system is not based on words but on ontological references in the underlying RDF(S) knowledge base and the links can be defined freely in terms of logical rules. The idea of annotating cultural artifacts with ontologies has been explored, e.g., in [11]. Other ontology-related approaches used for indexing cultural content include Iconclass [21] [33] and the Art and Architecture Thesaurus [22] [27].

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

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

---

[21] http://www.inconclass.nl/
[22] http://www.getty.edu/research/conducting_research/vocabularies/aat/

Acknowledgments

References

[1] D. Brickley and R. V. Guha. *Resource Description Framework (RDF) Schema Specification 1.0, W3C Candidate Recommendation 2000-03-27*, February 2000. http://www.w3.org/TR/2000/CR-rdf-schema-20000327/.

[2] Stefan Decker, Michael Erdmann, Dieter Fensel, and Rudi Studer. Ontobroker: Ontology based access to distributed and semi-structured unformation. In *DS-8*, pages 351–369, 1999. citeseer.nj.nec.com/article/decker98ontobroker.html.

[3] P. Dolong, N. Henze, and W. Neijdl. Logic-based open hypermedia for the semantic web. In *Proceedings of the Int. Workshop on Hypermedia and the Semantic Web, Hypertext 2003 Conference, Nottinghan, UK*, 2003.

[4] J. English, M. Hearst, R. Sinha, K. Swearingen, and K.-P. Lee. Flexible search and navigation using faceted metadata. Technical report, University of Berkeley, School of Information Management and Systems, 2003. Submitted for publication.

[5] D. J. Foskett. Thesaurus. In *Encyclopaedia of Library and Information Science, Volume 30*, pages 416–462. Marcel Dekker, New York, 1980.

[6] C. Goble, S. Bechhofer, L. Carr, D. De Roure, and W. Hall. Conceptual open hypermedia = the semantic web? In *Proceedings of the WWW2001, Semantic Web Workshop, Hongkong*, 2001.

[7] S. Handschuh, S. Staab, and F. Ciravegna. S-cream - semi-automatic creation of metadata. In *Proceedings of EKAW 2002, LNCS*, pages 358–372, 2002.

[8] S. Handschuh, S. Staab, and R. Volz. On deep annotation. In *Proceedings of International World Wide Web Conference*, pages 431–438, 2003.

[9] M. Hearst, A. Elliott, J. English, R. Sinha, K. Swearingen, and K.-P. Lee. Finding the flow in web site search. *CACM*, 45(9):42–49, 2002.

[10] J. Hefflin, J. Hendler, and S. Luke. Shoe: A knowledge representation language for internet applications. Technical report, Dept. of Computer Science, University of Maryland at College Park, 1999.

[11] L. Hollink, A. Th. Schereiber, J. Wielemaker, and B.J. Wielinga. Semantic annotations of image collections. In *Proceedings KCAP'03, Florida*, 2003.

[12] E. Hyvönen, M. Holi, and K. Viljanen. Designing and creating a web site based on RDF content. In *Proceedings of WWW2004 Workshop on Application Design, Development, and Implementation Issues in the Semantic Web,*

New York, USA. CEUR Workshop Proceedings, Vol-105, 2004. http://ceur-ws.org.

[13] E. Hyvönen, M. Junnila, S. Kettula, E. Mäkelä, S. Saarela, M. Salminen, A. Syreeni, A. Valo, and K. Viljanen. Finnish Museums on the Semantic Web. User's perspective on MuseumFinland. In *Proceedings of Museums and the Web 2004 (MW2004), Seleted Papers, Arlington, Virginia, USA*, 2004. http://www.archimuse.com/mw2004/papers/hyvonen/hyvonen.html.

[14] E. Hyvönen, M. Junnila, S. Kettula, S. Saarela, M. Salminen, A. Syreeni, A. Valo, and K. Viljanen. Publishing collections in the Finnish Museums on the Semantic Web portal – first results. In *Proceedings of the XML Finland 2003 conference. Kuopio, Finland*, 2003. http://www.cs.helsinki.fi/u/eahyvone/publications/ xmlfinland2003/FMSOverview.pdf.

[15] E. Hyvönen, S. Kettula, V. Raatikka, S. Saarela, and Kim Viljanen. Semantic interoperability on the web. Case Finnish Museums Online. In Hyvönen and Klemettinen [16], pages 41–53. http://www.hiit.fi/publications/.

[16] E. Hyvönen and M. Klemettinen, editors. *Towards the semantic web and web services. Proceedings of the XML Finland 2002 conference. Helsinki, Finland*, number 2002-03 in HIIT Publications. Helsinki Institute for Information Technology (HIIT), Helsinki, Finland, 2002. http://www.hiit.fi/publications/.

[17] E. Hyvönen, S. Saarela, and K. Viljanen. Application of ontology-based techniques to view-based semantic search and browsing. In *The semantic web: research and applications. First European Semantic Web Symposium, ESWS 2004, Heraklion, Greece*. Springer–Verlag, Berlin, May 2004. 92–106.

[18] E. Hyvönen, M. Salminen, and M. Junnila. Annotation of heterogeneous database content for the semantic web. In *Proceedings of the 4th International Workshop on Knowledge Markup and Semantic Annotation (SemAnnot 2004), Hiroshima, Japan*, November 2004.

[19] E. Hyvönen, M. Salminen, S. Kettula, and M. Junnila. A content creation process for the Semantic Web. In *Proceeding of OntoLex 2004: Ontologies and Lexical Resources in Distributed Environments, May 29, Lisbon, Portugal*, 2004.

[20] E. Hyvönen, A. Styrman, and S. Saarela. Ontology-based image retrieval. In Hyvönen and Klemettinen [16], pages 15–27. http://www.hiit.fi/publications/.

[21] O. Lassila and R. R. Swick (editors). Resource description framework (RDF): Model and syntax specification. Technical report, W3C, February 1999. W3C Recommendation 1999-02-22, http://www.w3.org/TR/REC-rdf-syntax/.

[22] M. Laukkanen, K. Viljanen, M. Apiola, P. Lindgren, and E. Hyvönen. Towards ontology-based yellow page services. In *Proceedings of WWW2004 Workshop on Application Design, Development, and Implementation Issues in the Semantic Web, New York, USA*. CEUR Workshop Proceedings, Vol-105, 2004. http://ceur-ws.org.

[23] R. L. Leskinen, editor. *Museoalan asiasanasto*. Museovirasto, Helsinki, Finland, 1997.

[24] A. Maedche, S. Staab, N. Stojanovic, R. Studer, and Y. Sure. Semantic por-

tal — the SEAL approach. Technical report, Institute AIFB, University of Karlsruhe, Germany, 2001.

[25] E. Mäkelä, E. Hyvönen, S. Saarela, and K. Viljanen. Ontoviews—a tool for creating semantic web portals. In *Proceedings of the 3rd International Semantic Web Conference (ISWC 2004), Hiroshima, Japan*, pages 797–811. Springer–Verlag, Berlin, November 2004.

[26] Steve Pepper. The TAO of Topic Maps. In *Proceedings of XML Europe 2000, Paris, France*, 2000. http://www.ontopia.net/topicmaps/materials/rdf.html.

[27] T. Peterson. Introduction to the Art and Architechure thesaurus, 1994. http://shiva.pub.getty.edu.

[28] A. S. Pollitt. The key role of classification and indexing in view-based searching. Technical report, University of Huddersfield, UK, 1998. http://www.ifla.org/IV/ifla63/63polst.pdf.

[29] J. Ben Schafer, Joseph A. Konstan, and John Riedl. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5(1/2):115–153, 2001.

[30] D. Soergel, B. Lauser, A. Liang, F. Fisseha, J. Keizer, and S. Katz. Reengineering thesauri for new applications: the agrovoc example. *Journal of Digital Information*, (4), 2004.

[31] L. Stojanovic, N. Stojanovic, and R. Volz. Migrating data-intensive web sites into the semantic web. In *Proceedings of the ACM Symposium on Applied Computing SAC-02, Madrid, 2002*, pages 1100–1107, 2002.

[32] Peter Stuer, Robert Meersman, and Steven De Bruyne. The HyperMuseum theme generator system: Ontology-based internet support for active use of digital museum data for teaching and presentations. In D. Bearman and J. Trant, editors, *Museums and the Web 2001: Selected Papers*. Archieves & Museum Informatics, 2001. http://www.archimuse.com/mw2001/papers/stuer/ stuer.html.

[33] J. van den Berg. Subject retrieval in pictorial information systems. In *Proceedings of the 18th international congress of historical sciences, Montreal, Canada*, pages 21–29, 1995. http://www.iconclass.nl/texts/history05.html.