

Metadata Catalogs with Semantic Representations

Yolanda Gil, Varun Ratnakar, and Ewa Deelman

USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
gil@isi.edu, varunr@isi.edu, deelman@isi.edu

Abstract.

Metadata catalogs store descriptive information about logical data items. These catalogs can then be queried to retrieve the particular logical data item that matches the criteria. However, the query has to be formulated in terms of the metadata attributes defined for the catalog. Our work explores the concept of virtual metadata, where catalogs can be queried using metadata attributes not originally defined in the catalog. We use semantic web standards, where new metadata attributes can be taken from shared ontologies and can include expressive axioms to define the new terms. We have implemented a virtual metadata catalog as an extension of the Metadata Catalog Service (MCS), using the Web Ontology Language (OWL) and a reasoning engine to map queries of temporal nature in several metadata catalogs.

1. Introduction

An integral part of today's large-scale science is the identification and access of large data sets. To support a scalable solution, many systems distinguish between data cataloging and data storage. Data cataloging is designed for ease of publication of data characteristics (metadata attributes including provenance information) and for ease of querying for data products based on the desired metadata attributes. Having uniquely identified the desired data products (by obtaining an identifier) then enables data access from an appropriate storage location. Metadata attributes and unique identifiers are stored in metadata catalogs, often accessible as services [3,8].

The process of data discovery by querying data sources may be quite complex, because there may be several heterogeneous metadata catalogs that are being published by a community. For example, in the astronomy community and under the umbrella of the National Virtual Observatory project (NVO, www.us-vo.org), astronomers publish data collected by a variety of sky surveys taken from the both ground and space-based telescopes. The surveys span a whole range of spectra from gamma- and X-rays, optical, infrared, through to radio. Each catalog may contain tens of millions of objects. Because the catalogs came online in different period of time, were published by different organizations and deal with different surveys, the metadata attributes are not common across the catalogs. Obviously these differences make it very hard to easily discover desired data products. The problem is that the metadata attributes are not defined or related to one another according to their meaning. Clients must figure out manually the meaning of the attributes, identify what are the relevant ones to query, and formulate queries that include all possibly

relevant metadata attributes resulting in redundancies in the query expression. This poses limitations in terms of the practical usability of these catalogs as well as the potential of existing approaches to scale up to larger and heterogeneous collections of data sources. These problems arise in similar projects in other disciplines, such as the Grid Physics Network (the GriPhyN project, www.griphyn.org) and the Southern California Earthquake Center's Community Modeling Environment (the SCEC-CME project, www.scec.org). There, a central goal is the distributed management of data collections that evolve over time and the consumption of those collections by an entire community with very diverse uses and possibly conceptualizations of the data.

This paper describes an approach that augments the existing metadata catalogs with semantic representations to create *virtual metadata catalogs*. Virtual metadata attributes are mapped to the original attributes that appear in the metadata catalog. This is analogous to the concept of virtual data in GriPhyN [1] or a virtual observatory in NVO [2], where a system can generate the data requested based on its description whether it already exists or it has to be generated from data that already exists. The definitions of the virtual attributes are represented declaratively, as well as any constraints that represent how attributes are interrelated. A query formulated in terms of the virtual metadata can be automatically expanded and translated into the original metadata attributes using the virtual metadata definitions and mappings. This can be done by using a logic reasoner that can handle expressive representations of definitions and relations. With this approach, integrating metadata catalogs can be done through shared ontologies and standard terminologies, decoupling the query formulation from the virtual metadata handling and from the particular metadata attributes that appear in the catalog.

In prior work we developed Artemis [3], a query mediator for metadata catalogs that used semantic representations to integrate several metadata catalogs. Artemis uses a centralized approach with a single reasoner that incorporates all the representations and mappings to all the metadata catalogs. The approach we take in this paper is decentralized in that a reasoner is associated with each metadata catalog. We describe our implementation of a virtual metadata query handler that uses semantic web technologies such as the Web Ontology Language (OWL) standard [4] to support virtual metadata queries for the Metadata Catalog Service (MCS) [5]. MCS is a metadata catalog we previously developed to support the publication and query operations on a variety of scientific metadata. It provides an extensible schema and an API that enables easy query and publication capabilities. In future work, we plan to combine this virtual metadata query handler with the query mediator used in Artemis to support the integration of distributed metadata catalogs in a decentralized manner.

The paper begins showing how semantic representations can express declaratively the meaning of metadata attributes, so that automated reasoners can derive relations and infer connections among attributes and data sets. Next it describes briefly existing standards for semantic representations, including RDF schemas and OWL. With this background in hand, the paper then introduces our approach to create virtual metadata attributes and catalog services. The paper ends with a description of an implemented virtual metadata catalog service and discusses important future work.

2. The Need for Semantic Representations of Metadata Attributes

Unless the meaning and interdependencies of metadata attributes are represented declaratively with expressive languages, metadata catalogs have limitations in the way they can be used to query data. An important limitation arises when the meaning of the attributes is not represented explicitly and is instead implicit in the name of the metadata attribute. For example, an attribute named `execution-time` could mean elapsed time or CPU time. The right answer can only be determined manually by looking up the documentation of the catalog or finding out from its developers the consensus meaning of the attribute.

Another problem arises when the interdependencies among attributes are not represented. For example, the duration of an event is related to the start time and the end time of that event, and in this example `execution-time`, `begin-execution-time` and `end-execution-time` are related. If some of the data is missing the `execution-time` it could be derived from its `start-execution-time` and the `end-execution-time`. Otherwise, queries would need to be formulated to include both cases explicitly. The advantage of having these kinds of relationships expressed declaratively as part of the metadata definitions is that queries only have to mention what is needed and disregard what particular metadata attributes are present in each specific case.

All these problems are exacerbated when there is a need to query several metadata catalogs, where the attribute names will be likely to be independent, and any correlation that might exist among attributes in different catalogs is not declared explicitly.

3. Standards for Semantic Representation and Reasoning

In this work we draw upon three semantic web technologies: semantic data representations defined using relevant domain terms, ontologies that attribute meaning to the terms and reasoners that can answer queries about the terms and their relationships.

RDF [6] is a web standard for representing resources on the web. It restricts the description of resources to statements composed of subject, predicate and object triples. It uses XML [7] as the interchange syntax. An RDF Schema (RDFS) [8] defines the terms that will be used in the RDF statements and gives specific meanings to them. The Web Ontology Language (OWL) may also be used to define those terms using more expressive representations. OWL builds on top of RDF. Because these languages are built on web standards, they take advantage of namespaces and URIs to define the scope of the definitions and to import definitions from distributed locations respectively.

We use OWL-DL representations in this work because of its expressive power and because there are efficient reasoners already available for it. We will briefly illustrate here the expressivity of OWL with examples from temporal reasoning, using definitions from the a variant of the OWL-Time ontology [9]. An interval can be defined as:

```
<owl:Class rdf:ID="IntervalThing">
  <rdfs:subClassOf rdf:resource= "#TemporalThing"/>
```

```

    <rdfs:subClassOf><owl:Restriction>
      <owl:onProperty rdf:resource="#from" />
      <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
    </owl:Restriction></rdfs:subClassOf>
    <rdfs:subClassOf><owl:Restriction>
      <owl:onProperty rdf:resource="#to" />
      <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
    </owl:Restriction></rdfs:subClassOf>
  </owl:Class>

```

Here we define an IntervalThing as a subClass of a generic TemporalThing that has a beginning (“from”) and an end (“to”). It also defines restrictions on its properties “from” and “to” (declared below) in that they can only have one value (indicated by a cardinality of 1). Note that the prefixes owl, rdfs, and rdf refer to terms from their respective namespaces.

```

<owl:ObjectProperty rdf:ID="from">
  <rdfs:domain rdf:resource= "#TemporalThing"/>
  <rdfs:range rdf:resource="#InstantThing"/>
  <rdf:type rdf:resource= "&owl;FunctionalProperty"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="to">
  <rdfs:domain rdf:resource= "#TemporalThing" />
  <rdfs:range rdf:resource="#InstantThing"/>
  <rdf:type rdf:resource= "&owl;FunctionalProperty"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="duration">
  <rdfs:domain rdf:resource= "#TemporalThing"/>
  <rdfs:range rdf:resource="&xsd;duration"/>
</owl:DatatypeProperty>

```

The above definitions declare the properties “from”, “to”, and “duration” of any TemporalThing. Of note is the range of these properties. The range of a property implies the types of values that the property might have. The range of duration is a native XML Schema datatype called duration. The “&xsd” marker indicates the namespace of the XML Schema. XML Schema provides several built-in datatypes such as Integer, String, dateTime, etc. More details and a thorough introduction of OWL, are available from [\[1\]](#). Many tools are available for OWL including ontology editors, parsers, and reasoners, many surveyed in [\[4\]](#).

It is important to note that OWL does not have the full expressive power of first-order logic, which is the representation of choice in many knowledge representation and reasoning systems. For example, the time ontology OWL-Time was originally specified in first order logic, and the OWL versions of it lack many of the axioms and constraints of the original. To address this issue, rule languages are being developed to complement OWL and to support the representation of more expressive relations and constraints. The following rule expresses how to derive the “to” property from the “from” property and “duration” property:

```

[r1: (?x rdf:type tme:IntervalThing), (?x tme:from ?a), (?x tme:duration ?t2), (?a
tme:at ?t1), sum(?t1, ?t2, ?t3)
makeTemp(?v) -> (?v rdf:type tme:InstantThing) (?v tme:at ?t3) (?x tme:to ?v)]

```

There are no current semantic web standards for rule languages or query languages, although some have been proposed (RuleML[10] and OWL-QL[11]). The rule format shown in the example above is used by Jena [12], the reasoner used in our system. Standard rule and query languages will inevitably emerge soon.

4. Approach

Figure 1 illustrates our approach. We propose to augment metadata catalogs with a semantic layer that supports queries in terms of virtual metadata attributes, resulting in virtual metadata catalog services. These attributes are virtual in that they are not really used in the implementation of the catalog. However, virtual metadata attributes can be used to query the catalog transparently as if they actually were associated with the data. To support this functionality, the virtual metadata attributes need to be mapped to the metadata attributes that are actually contained in the catalog (*actual attributes*).

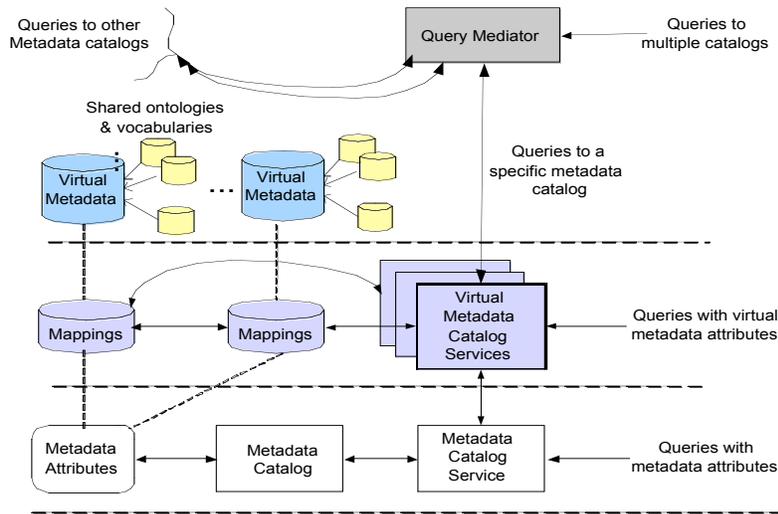


Fig. 1 Approach towards a distributed metadata catalog query.

Mapping queries given in terms of virtual metadata attributes into queries in terms of the actual metadata attributes can be very complex. As we motivated in the examples of the previous section, expressive semantic representations and reasoners are needed to do these mappings automatically. We use standards for semantic representation and reasoning when they exist. Additional standards and tools are under development that will support increasingly more expressive query languages and mappings.

The figure also illustrates that in different contexts users may have different preferences or standards for querying the data in a catalog, resulting in alternative virtual metadata catalog services that can be built on top of the same underlying metadata catalog. For example, a catalog of historical weather data could be used by a climatologist to test a weather prediction model, or by an oceanographer to correlate underwater vegetation with weather conditions. Some of these virtual metadata attributes could be drawn from shared ontologies or standard vocabularies, which are becoming commonplace in many scientific communities [13-17]. Many scientific ontologies are already undergoing conversion to semantic web standards [4, 6-8] and others will soon follow as the benefits of these expressive languages are shown. Users may also define their own virtual metadata attributes by creating customized

ontologies, effectively creating personalized metadata catalogs.

The approach is modular and decentralized in that each virtual metadata catalog service reasons about its own virtual metadata within its own reasoners. This is in contrast with our work on Artemis, where a centralized reasoner resolved all the mappings to all the metadata catalogs. The advantages of this decentralized approach is that it will be more robust to failures. In addition, the reasoning tasks will be more manageable and will scale better as more metadata catalogs are added.

5. Virtual Metadata Catalogs

Figure 2 illustrates the architecture of our implementation of a virtual metadata catalog developed using MCS. We use OWL in combination with rules to express the query, the shared domain ontologies, and the virtual metadata attributes and mappings. The original query is provided as an OWL document that includes references to the domain ontologies from where the virtual metadata attributes in the query are drawn. The query may also reference terms from a generic catalog ontology that we have created. The purpose of this ontology is to define terms such as “files”, “views”, “collections”, that are used in typical queries to MCS.

The central component of the architecture is the Query Mapping module. It takes the OWL query and turns it into an MCS query that uses the metadata attributes that actually appear in the catalog. The MCS query is then submitted to the MCS, which returns all the references to data stored in it that satisfy the query. We will use an example to explain in detail how the Query Mapping module works.

Consider a query for data within a temporal interval starting on 10th October 2004 at 10am and a duration of 30 seconds. Suppose the user wishes to query using the virtual metadata attributes “from” and “duration”, both taken from the OWL Time ontology. Assume that the metadata attributes present in the MCS are “startDate” and “endDate”. The core of the original OWL query is:

```
<tme:IntervalThing rdf:ID="Interval1">
  <tme:from rdf:resource="#T1"/>
  <tme:duration rdf:datatype = "&xsd;duration"> PT30S </tme:duration>
</tme:IntervalThing>
<tme:InstantThing rdf:ID="T1">
  <tme:at rdf:datatype="&xsd;dateTime">2004-01-01T10:00:00</tme:at>
</tme:InstantThing>
```

The Virtual Metadata Attributes and Mappings express that the MCS “startDate” attribute is equivalent to the “from” virtual metadata attribute, and the MCS attribute “endDate” is equivalent to the “to” virtual metadata attribute. Here is how these mappings are specified:

```
<owl:ObjectProperty rdf:ID="startDate">
  <owl:equivalentProperty rdf:resource = "&tme;from"/>
  <terms:hasMCSAttribute> startDate </terms:hasMCSAttribute>
  <terms:pathToData>->at</terms:pathToData>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="endDate">
  <owl:equivalentProperty rdf:resource = "&tme;to"/>
  <terms:hasMCSAttribute> endDate </terms:hasMCSAttribute>
  <terms:pathToData>->at</terms:pathToData>
</owl:ObjectProperty>
```

The Query Mapping component accepts the OWL Query document and configures

the semantic reasoner by loading the OWL ontologies and rules referenced in it. The query mapping process is depicted in Figure 3, and is composed of three major steps. First, the basic query constituents are created by running the reasoner and generating attribute/value pairs based on the rules defined. In our case, the rule that defines “to” in terms of interval duration would be used to generate the following attribute/value pairs:

```
{from=[2004-01-01T10:00:00 ^http://www.w3.org/2001/XMLSchema#dateTime],
  to=[2004-01-01T10:00:30Z ^http://www.w3.org/2001/XMLSchema#dateTime]}
```

The suffix starting with ^^ in RDF signifies the datatype of the value that it is appended to.

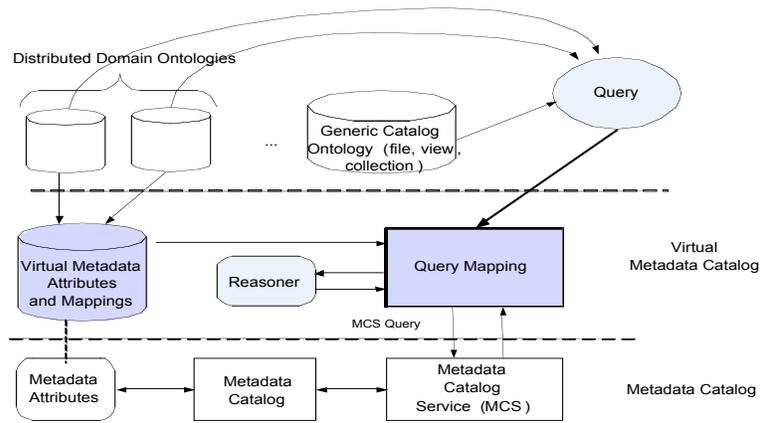


Fig. 2 Architecture of a Virtual Metadata.

These virtual metadata attribute value pairs are then converted into MCS attribute value pairs by selecting the relevant subset of the triples and applying the relevant mappings. Therefore, “from” is replaced by “startDate”, and “to” is replaced by “endDate”. Another mapping performed in this step is the conversion of the values from the XML Schema Datatypes to the ones that are expected by the database. In our case, the dateTime formats of the OWL Query need to be converted to the Date types that are expected for the startDate and endDate. Finally, the MCS query is constructed by adding the operators to construct the appropriate query formula.

In our implementation we used data from three different domains: climate modeling, earthquake science, and workflow execution tracking. The climate modeling catalog contains such information as longitude, latitude, temperature and date and time. The earthquake science catalog collects data about simulation results that show seismic wave propagation over time. This catalog has more than one hundred metadata attributes. Finally, the workflow tracking catalog includes data about the names of the workflow tasks, their execution duration, the execution location (resource used), success or failure and others. Although these various domains deal with different types of data, they all make use of temporal concepts.

To add new virtual metadata attributes, a user would only have to define their mappings into MCS attributes using similar definitions to the ones shown above to map "from" and "to" to "startTime" and "endTime" respectively.

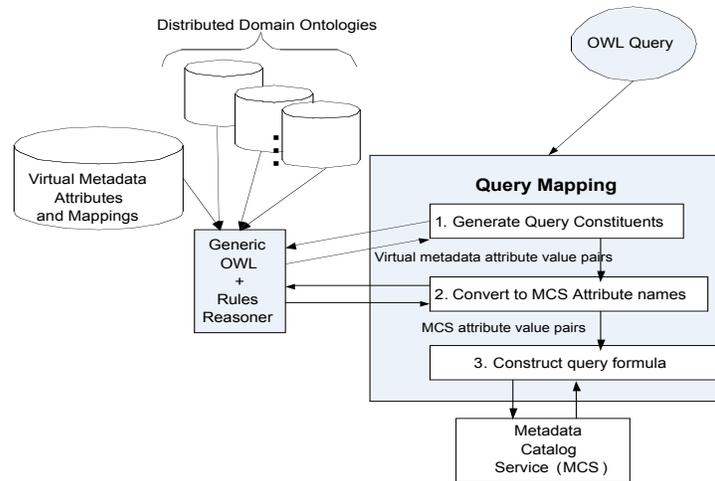


Fig. 3 A more detailed overview of the query mapping process.

6. Discussion

Our work illustrates how semantic representations can be used to support virtual metadata attributes and how reasoners can be used to resolve queries that use them, opening the way for virtual metadata catalog services. As proof of concept, the system implemented so far can answer queries but that is only part of the functionality of a metadata catalog service. We plan to extend its functionality to the fullest in the future and include useful functions such as returning lists of available metadata attributes, publish data, group data into hierarchical collections, and set authorization information on objects.

An important issue that needs to be addressed more fully in the future is handling alternative data formats. In our examples, time points and durations were represented using the XML schema data types, but other metadata attributes may be defined using non-standard formats. A point in time can be expressed in different syntactic formats such as 2004/01/30-23:34:48 or as 30/1/04.11:34pm, and a query on time points for November 30 of 2004 should retrieve both. Addressing this issue requires representing mappings between alternative formats, which could be done using the same approach we have used to define mappings among different terms in our system. The same issue applies to transport formats, for example the year 2004 could be rendered as a string or a number. At this point it is not clear how far reaching the transformations need to be. One can imagine a whole spectrum from simple date and time transformations to more complicated coordinate transformations as is sometimes necessary in astronomy. Astronomers for example use a variety of different coordinate formats to point to specific locations in the sky. Sometimes they also use a stellar object name to denote a location. We also plan to investigate how to support the integration of multiple catalog services using query mediators as was done in our previous work on Artemis. When catalogs are mapped to identical shared ontologies, their integration to a query mediator should be simplified.

Another important issue to investigate is how this architecture scales up to large amounts of attributes and multiple metadata catalog services. Semantic web technologies are being developed very quickly to reason efficiently as the amount of data and definitions grow. Existing systems handle millions of RDF triples, a basic unit to render RDF and OWL descriptions, however evaluating systems that rely on these technologies in real deployments.

7. Related Work

This work bridges several research areas: metadata management, query mediation and semantic web technologies. In this section we mention the most relevant work in these areas. In terms of metadata management, besides the Metadata Catalog Service mentioned in Section 1, the Storage Resource Broker (SRB) from the San Diego Supercomputing Center [18] and its associated MCAT Metadata Catalog [19] provide metadata and data management services. SRB supports a logical name space that is independent of physical name space. The logical objects, logical files in the case of SRB, can also be aggregated into collections. SRB provides various authentication mechanisms to access metadata and data within SRB. Unlike the work presented here, SRB is based on a centralized metadata catalog and does not provide semantic information about the catalog content.

In [20], the authors describe a mediator-based system that utilizes the semantics of the data exported by the data sources to integrate the data. A key assumption in the [20] paper is that the data sources export the semantics of the data. This work is complimentary as it provides a means for the semantics to be added to the data sources and thus available to existing mediator systems. The myGrid project [21] is developing and exploiting semantic web technology to describe and integrate a wide range of services in a grid environment. Data sources are modeled as semantic web services, and are integrated through web service composition languages. The result is a workflow that may include not only steps to access to data sources, but also as simulation or other data processing steps. A key difference between myGrid and the work presented here is that myGrid relies on the use of standard ontologies from the bioinformatics domain and thus the problem of semantic information representation is greatly simplified. Unlike other domains, scientists in bioinformatics have made great strides in design common semantic representations.

8. Conclusions

We have designed and implemented a virtual metadata catalog that provides rich semantic information about the catalog content in a variety of semantics views. The views are customized to a particular global ontology. This system provides an easy way for users to publish and discover data using metadata attributes that are appropriate for them. In this work we drew upon data from three different disciplines: atmospheric sciences, performance databases and earthquake science. We also put the Virtual Metadata Catalog in a broader context of a distributed system, where multiple such catalogs would exist and query mediation technologies such as those based on our previous system (Artemis) would be used to query across the multiple catalogs.

In future work, we would like to extend the query mapping process to make it more robust and better integrated with OWL reasoners as well as the MCS back end.

We would like to formalize the mappings of different query expressions in a comprehensive framework. This will be facilitated as standard OWL query languages emerge. We also plan to use the virtual metadata catalog in some of our ongoing projects.

Acknowledgements

We gratefully acknowledge the support from the NSF's Shared Cyberinfrastructure program under grant SCI-0455361 and the NSF SCEC-CME project with grant number EAR-0122464.

References

- [1] E. Deelman, et al. Representing Virtual Data: A Catalog Architecture for Location and Materialization Transparency," TR GriPhyN-2001-14, 2001.
- [2] P. Messina and A. Szalay, "Building the Framework for the National Virtual Observatory," <http://www.us-vo.org/docs/nvo-proj.doc>, 2002.
- [3] R. Tuchinda, S. Thakkar, Y. Gil, and E. Deelman., "Artemis: Integrating Scientific Data on the Grid," Proceedings of IAAI, San Jose, California, 2004 .
- [4] "Web Ontology Language (OWL)," <http://www.w3.org/2001/sw/WebOnt/>
- [5] E. Deelman, et al. "Grid-Based Metadata Services," Proceedings of Statistical and Scientific Database Management (SSDBM), Santorini, Greece, 2004.
- [6] "RDF," <http://www.w3.org/TR/REC-rdf-syntax/>,
- [7] "XML Schema," <http://www.w3.org/XML/Schema>
- [8] "RDF Schema," <http://www.w3.org/TR/rdf-schema/>
- [9] J. R. Hobbs and F. Pan, "An Ontology of Time for the Semantic Web," ACM TALIP: Special issue on Temporal Information Processing, (3,) pp. 66-85, 2004.
- [10] "RuleML," <http://www.ruleml.org/>
- [11] R. Fikes, P. Hayes, and I. Horrocks, "OWL-QL: A Language for Deductive Query Answering on the Semantic Web," KSL Technical Report 03-14 2003.
- [12] "Jena," <http://jena.sourceforge.net>
- [13] C. J. Wroe et al., "A Methodology to Migrate the Gene Ontology to a Description Logic Environment Using DAML+OIL," Proc. of Pacific Symposium on Biocomputing 2003.
- [14] U. Hahn and S. Schulz, "Building a Very Large Ontology from Medical Thesauri," in Handbook on Ontologies, R. S. S. Staab, Ed.: Springer, 2004.
- [15] "GO," <http://www.geneontology.org/>
- [16] "UMLS," <http://www.nlm.nih.gov/research/umls/>
- [17] J. Golbeck, et al. "The National Cancer Institute's Thesaurus and Ontology," Journal of Web Semantics, vol. 1, 2003.
- [18] C. Baru et al, "The SDSC Storage Resource Broker," Proceedings of CASCON'98, 1998.
- [19] "MCAT (Version 1.1)," <http://www.npaci.edu/DICE/SRB/mcat.html>
- [20] B. Ludäscher, et al, "A Model-Based Mediator System for Scientific Data Management," in Bioinformatics:Managing Scientific Data, Critchlow, Lacroix, Eds.: MK, 2003.
- [21] C. Wroe, et al, "A Suite of DAML+OIL Ontologies to Describe Bioinformatics Web Services and Data," Int. J. of Cooperative Information Systems, 2003.