GES 2007 German e-Science

Available online at http://www.ges2007.de

A Grid Middleware for Ontology Access

Michael Hartung[1] and Erhard Rahm[2]

[1] Interdisciplinary Centre for Bioinformatics, University of Leipzig, Germany
[2] Department of Computer Science, University of Leipzig, Germany
Email: {hartung,rahm}@informatik.uni-leipzig.de
phone: (+49 0341) 97 {32240, 32221}

Abstract

Many advanced grid applications need access to ontologies representing knowledge about a certain application domain. To deal with the high heterogeneity of available ontologies, we propose a general service-oriented middleware for making ontologies accessible to grid applications. Our implementation is integrated in the German D-Grid infrastructure and provides several applications a uniform access to biomedical ontologies such as Gene Ontology, NCI Thesaurus and several OBO ontologies.

1 Introduction

Grid computing provides scientists with a distributed infrastructure for collaboration and massive amounts of computing, storage and data resources. Ontologies increasingly gain importance for grid computing to semantically describe resources and to support an improved interoperability between applications. Especially applications in the biomedical domain make use of the ontology concept. For example, in biology and bioinformatics ontologies have become essential for annotating molecular biological objects or publications and integrating heterogeneous data resources.

Recently, several proposals and recommendations for a semantic grid architecture were made. A reference architecture for semantic grids, S-OGSA, is presented in [1]. The model architecture includes ontology services that enable access to conceptual models and ontologies. While this underlines the high need of a service-based grid middleware for ontology access, the services have not yet been implemented. At a recent workshop, two services were discussed to access RDF-based metadata: OGSA-DAI-RDF [11] and WS-DAIOnt (OntoGrid) [7]. However, both approaches are limited to RDF-based resources while many ontologies, e.g. in the biomedical domain, are stored in relational, XML, OWL, CSV or standardized flat file formats.

We thus propose a generic service-based middleware for ontology access in grid systems which can accommodate ontologies stored in different formats. The goal is to provide grid applications with a uniform and transparent access to different distributed ontologies hiding specifics of their storage location and formats. Our middleware services have already been implemented based on Grid standards and are part of the German D-Grid [2] middleware. First applications using the ontology services have been developed within the MediGRID [12] community project.

In the next section, we give an overview of our service-based middleware for ontology access in grid systems. We show and explain the parts of the middleware and give examples for the used resources. Section 3 presents a detailed interaction scenario between clients and the middleware. Section 4 discusses current applications of our middleware, namely portlets in the

MediGRID portal that use our middleware to access biomedical ontologies in MediGRID. We conclude with a summary and outlook.

2 Architecture of ontology access middleware

2.1 Overview

In order to deal with the various formats of ontologies and to be compatible with the D-Grid infrastructure and its basis software, we decided to reuse and extend OGSA-DAI [10] features in our middleware. The architecture of the ontology access middleware consists of three main parts (Fig. 1): ontology sources, ontology services including the ontology info service and client applications.

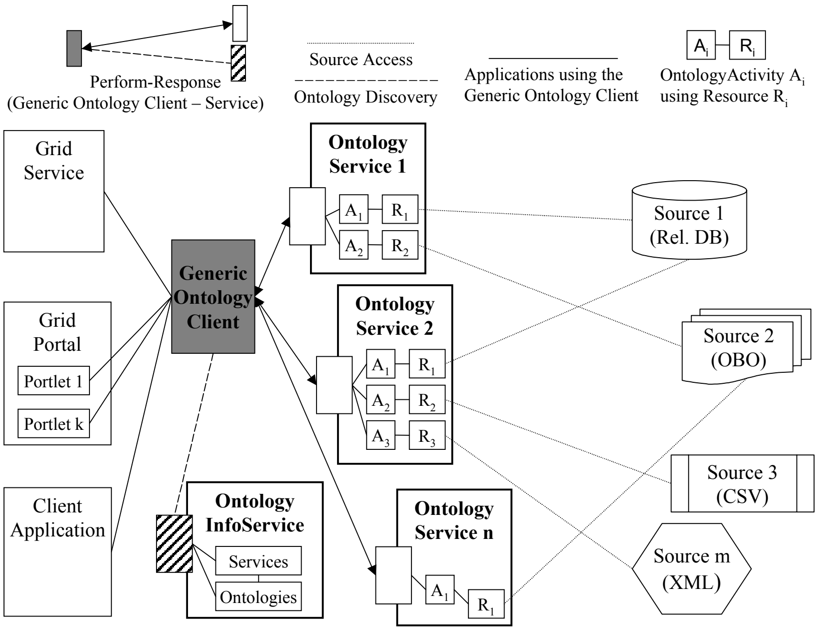


Figure 1: The architecture of the ontology access middleware

2.2 Ontology Sources

We support different source formats for ontologies, in particular relational databases, XML, CSV and standardized flat file formats like OBO (Open Biomedical Ontologies). The approach is generic and extensible because adaptors for other formats can be added and included in the middleware. Ontology sources are independent of our middleware, i.e. administrators can update and manage them through their own specialized APIs. Furthermore the location of an ontology source needs not to be on the same site as an ontology service that accesses it.

Currently we have integrated about 15 ontology sources in the MediGRID project. These include the GeneOntology (GO) [5], the thesaurus of the National Cancer Institute (NCIThesaurus) [16] and several Open Biomedical Ontologies (OBO) [14] like HumanDiseaseOntology, ProteinProteinInteractionOntology or SequenceOntology (SO) [3]. GO is distributed as a relational database and provides a controlled vocabulary to describe gene and gene product attributes of different organisms using three sub ontologies: molecular function, biological process and cellular component. NCIThesaurus is available in CSV and XML formats. It focuses on the medical domain of cancer research and covers vocabularies for clinical care, translational and basic research as well as public information and administrative activities. The OBO ontologies are stored in flat files consisting of attribute – value combinations. This format is supported for freely available biomedical ontologies.

2.3 Ontology Services and the Ontology Information Service

Ontology services are at the heart of the access architecture. An ontology service is a grid service providing a uniform access interface to one or several ontology sources. Each ontology source is connected with at least one ontology service but can also be assigned to several distributed ontology services. Hence, there is a many-to-many relationship between ontology services and ontology sources. This is illustrated in Fig. 1 where ontology source 2 is associated with two ontology services. On the other side, ontology services 1 and 2 provide access to more than one ontology source. This architectural flexibility is important for load balancing and a high reliability so that ontologies remain usable even under high performance demands and in the presence of server failures.

Within our architecture we provide an information structure to support ontology discovery for clients and services. This is achieved by a central grid service called ontology information service. The service enables clients and other services to search and discover the distributed ontologies and their associated ontology services in the grid. Typically, this central service is the starting point for interacting with the ontology middleware.

The ontology services are based on the grid data services of OGSA-DAI and the Web Services Resource Framework (WSRF) [4]. Our implementation of the access middleware uses the OGSA-DAI resources to access the physical ontology sources. An ontology service thus has a *resource* description for each physical ontology source containing information needed for interaction with the ontology. For instance, a resource for accessing a relational ontology keeps information on the JDBC driver, mappings between user credentials and database logins, and database locations. An important advantage is that resources and thus ontologies can dynamically be added to or removed from an ontology service. This makes it easy to support additional ontologies.

Ontology services support an *OntologyActivity API* to allow clients a uniform access to different ontologies. This API contains a variety of methods to access information of ontology concepts, e.g. accession ID, name, definition, synonyms, relations to other concepts and cross references. For each ontology, ontology activities need to be provided implementing the API methods. To this end we extend the currently available activities of the OGSA-DAI framework. As shown in Fig. 2, the general OGSA-DAI activities and the ontology-specific activities represent an inheritance hierarchy. This hierarchy of ontology activities simplifies the integration of new ontologies. In particular, we can reuse existing ontology activities, e.g. AbstractOntologySQLActivity or AbstractOBOActivity, to develop more special ontology activities for additional ontology sources.

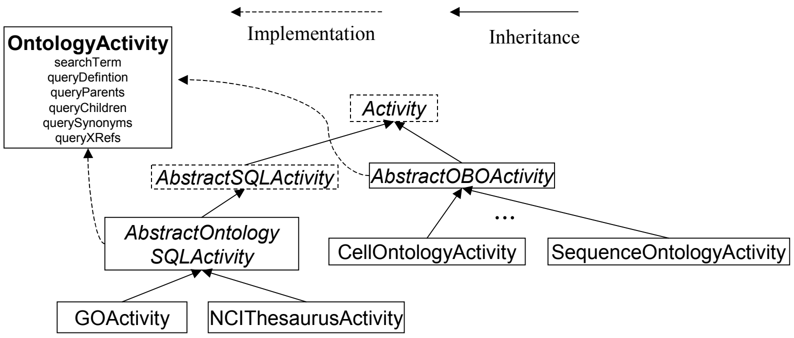


Figure 2: Ontology activities and the OntologyActivity interface (dotted boxes: OGSA-DAI, normal boxes: extensions to OGSA-DAI)

2.4 Clients

A large variety of clients can make use of the provided ontology services, e.g. stand-alone applications, grid services or portlets in grid portals. A generic *ontology client* is used to request and receive information from both the ontology information service and the ontology services and their resources. To access an ontology, a request document is generated and sent to an ontology service registered in the ontology information service and responsible for the desired ontology. An access method and its parameters are specified in the request document. The corresponding ontology activity uses the parameters and information of the request to perform an access to the specified ontology source. The result of an access is returned to the client as a CSV - styled document. The client extracts the returned ontology information from the document for further data processing.

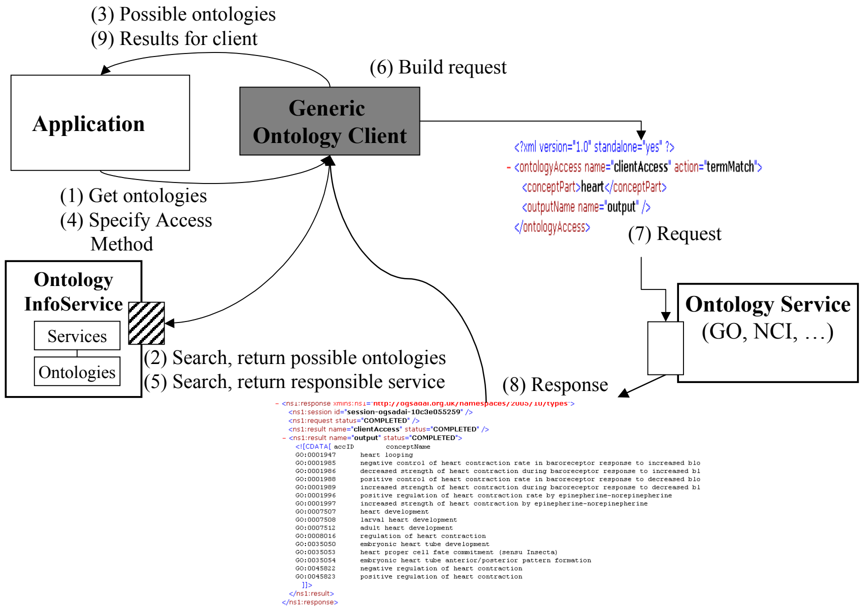
The next section will describe a detailed scenario using the introduced infrastructure for accessing an ontology.

3 A Detailed Access Scenario

To illustrate ontology access using the proposed architecture we explain a typical access scenario which is illustrated in Fig. 3. The various steps involved are explained in the following.

As a first step the application selects the ontology of interest. For this purpose, a request is sent to the ontology client (1) and then to the ontology information service to obtain all registered ontologies in a response document (2). The possible ontologies are returned to the client application (3), where one is selected.

In the next step the client application specifies the request method and its parameters (4). Currently we provide basic and more complex methods for retrieving information of an ontology. Methods are search facilities for concepts, getter methods for ID, name, definition and synonyms of a concept, and complex methods like retrieval of relations between concepts (e.g. 'is_a' and 'part_of') or cross references to other ontologies or associated databases. Furthermore, an overall method is provided collecting all information about an existing concept. Each method requires specific parameters for its work. E.g. the search method needs search terms to look for, ID or the name of a concept are needed to access concrete information of a concept.



```
<?xml version="1.0" standalone="yes" ?>
- <ontologyAccess name="clientAccess" action="termMatch">
    <conceptPart>heart</conceptPart>
    <outputName name="output" />
  </ontologyAccess>
```

```
- <ns1:response xmlns:ns1="http://ogsadai.org.uk/namespaces/2005/10/types">
    <ns1:session id="session-ogsadai-10c3e055259" />
    <ns1:request status="COMPLETED" />
    <ns1:result name="clientAccess" status="COMPLETED" />
  - <ns1:result name="output" status="COMPLETED">
      <![CDATA[ accID        conceptName
      GO:0001947      heart looping
      GO:0001985      negative control of heart contraction rate in baroreceptor response to increased blo
      GO:0001986      decreased strength of heart contraction during baroreceptor response to increased bl
      GO:0001988      positive control of heart contraction rate in baroreceptor response to decreased blo
      GO:0001989      increased strength of heart contraction during baroreceptor response to decreased bl
      GO:0001996      positive regulation of heart contraction rate by epinepherine-norepinepherine
      GO:0001997      increased strength of heart contraction by epinepherine-norepinepherine
      GO:0007507      heart development
      GO:0007508      larval heart development
      GO:0007512      adult heart development
      GO:0008016      regulation of heart contraction
      GO:0035050      embryonic heart tube development
      GO:0035053      heart proper cell fate commitment (sensu Insecta)
      GO:0035054      embryonic heart tube anterior/posterior pattern formation
      GO:0045822      negative regulation of heart contraction
      GO:0045823      positive regulation of heart contraction
        ]]>
    </ns1:result>
  </ns1:response>
```

Figure 3: Scenario of an interaction with the ontology access middleware

If the client application has specified the ontology, method and parameters, the ontology client contacts the central information service to look for possible grid resources (ontology services) that can handle the request (5). Thereby the ontology information service checks which ontology services are currently available and provide access to the specified ontology. The list of matched resources is used to randomly select one ontology service to be used for interaction. The location and name of the selected ontology service are returned to the ontology client, where the next step is initiated.

Based on the specified method and its parameters a request document (SOAP message) is generated automatically with the help of the ontology client and OGSA-DAI toolkit facilities (6). This request document is sent to the ontology service that was determined by the ontology information service in step (5).

When an ontology service receives a request document (7), it parses the specified method and parameters of the request. With the help of this information the ontology service selects and instantiates a corresponding internal method that is used to obtain information of the desired ontology. These internal methods are part of the ontology activities (Sec. 2.3) and contain special access tasks, e.g. a query on a relational database, a flat file reader or a parser for a XML document. The results of an access are converted into a CSV - styled result schema. This schema is used to build a response document that is generated automatically by the OGSA-DAI infrastructure. Finally the response document is sent back to the calling client (8).

In the final step (9) the generic ontology client de-serializes the received response document and extracts the CSV - styled information of the ontology access. These results are now used by the ontology client to build up an easy to handle representation of the data for client applications. For example,. the search results can simply be converted into a list of concepts. Moreover, objects representing the information of an ontology concept are built and used by client applications for further processing. These objects store the desired concept information like ID, name, definition, comments, synonyms or relations to other concepts of the ontology. Supplementary functions of these objects allow the generation of more complex representation formats, e.g. images showing the relations of a concept to other concepts or the hierarchy between the root concept and the stored one. For instance, these representations are used by applications in the MediGRID portal.

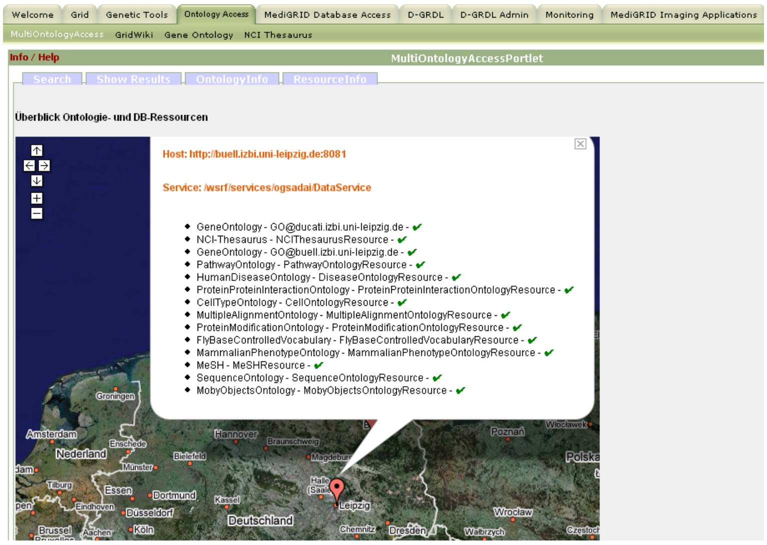
4 Applications using the ontology access middleware

Current grid technologies are often difficult to learn and use. Therefore, developing an easy-to-use portal interface on top of these grid technologies can make it easier for users to utilize grid technologies and underlying resources such as our middleware for ontology access. In MediGRID, a portal based on the GridSphere Portal Framework [13] and its Grid Portlets extension [15] is used to interact with community-specific grid applications. Thus, we integrated our generic ontology client in portlets to enable access to biomedical ontologies for MediGRID portal applications.

Currently, three MediGRID applications, namely the Ontology Look Up Service, the gene predication tool AUGUSTUS [17] and the SNPSelection Service [8] make use of the proposed ontology access middleware to interlink application specific data with available ontologies. The next sections will describe these applications and their usage of ontologies.

4.1 Ontology Look Up Service

The Ontology Look Up Service is designed as a portlet that enables look up and information features for all integrated ontologies in MediGRID. The portlet uses the generic ontology client of our middleware to search in ontologies and to present ontology information via the portal user interface. The portlet itself consists of three sub applications named Search, Result and Monitor.



Host: http://buell.izbi.uni-leipzig.de:8081

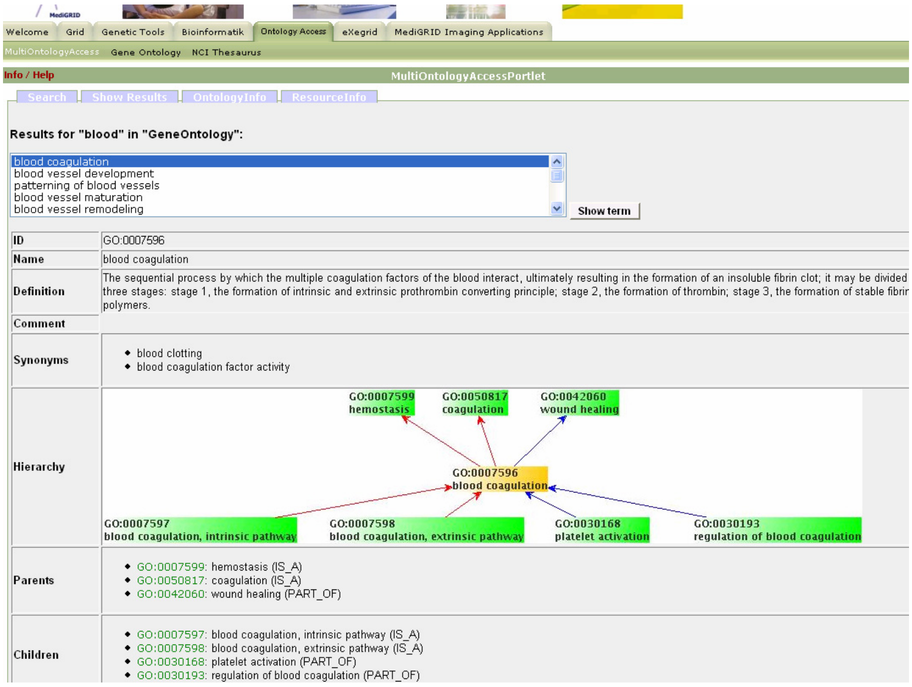
Service: /wsrf/services/ogsadai/DataService

- GeneOntology - GO@ducati.izbi.uni-leipzig.de - ✓
- NCI-Thesaurus - NCIThesaurusResource - ✓
- GeneOntology - GO@buell.izbi.uni-leipzig.de - ✓
- PathwayOntology - PathwayOntologyResource - ✓
- HumanDiseaseOntology - DiseaseOntologyResource - ✓
- ProteinProteinInteractionOntology - ProteinProteinInteractionOntologyResource - ✓
- CellTypeOntology - CellOntologyResource - ✓
- MultipleAlignmentOntology - MultipleAlignmentOntologyResource - ✓
- ProteinModificationOntology - ProteinModificationOntologyResource - ✓
- FlyBaseControlledVocabulary - FlyBaseControlledVocabularyResource - ✓
- MammalianPhenotypeOntology - MammalianPhenotypeOntologyResource - ✓
- MeSH - MeSHResource - ✓
- SequenceOntology - SequenceOntologyResource - ✓
- MobyObjectsOntology - MobyObjectsOntologyResource - ✓

Figure 4: Monitoring of ontology services in the MediGRID portal

With the help of monitoring users can obtain information about available ontologies and ontology services in the grid. An integrated map displays all registered ontologies and their services with information about their availability status (Fig. 4). In the Search panel users find a list of available ontologies and a text field for keyword search actions. The user chooses an ontology and submits keyword(s) of interest to the system. After a lookup via an ontology service a list of matched concepts is displayed. By selecting one of these concepts a user can request more detailed information.

In the Result panel (Fig. 5) detailed information of a concept is presented with the help of tables and graphs. Simple information like name, ID, definition, comments or synonyms is printed in tabular form, while relationships between concepts are displayed in a graph. With the help of these relationships users can navigate through the existing ontology to get information about more general or more specific concepts compared to the selected one. The underlying middleware allows MediGRID users to utilize these functions uniformly across different ontologies. New ontologies can easily be added and seamlessly be accessed.



Results for "blood" in "GeneOntology":

blood coagulation
blood vessel development
patterning of blood vessels
blood vessel maturation
blood vessel remodeling

ID	GO:0007596
Name	blood coagulation
Definition	The sequential process by which the multiple coagulation factors of the blood interact, ultimately resulting in the formation of an insoluble fibrin clot; it may be divided three stages: stage 1, the formation of intrinsic and extrinsic prothrombin converting principle; stage 2, the formation of thrombin; stage 3, the formation of stable fibrir polymers.
Comment	
Synonyms	• blood clotting • blood coagulation factor activity
Hierarchy	
Parents	• GO:0007599: hemostasis (IS_A) • GO:0050817: coagulation (IS_A) • GO:0042060: wound healing (PART_OF)
Children	• GO:0007597: blood coagulation, intrinsic pathway (IS_A) • GO:0007598: blood coagulation, extrinsic pathway (IS_A) • GO:0030168: platelet activation (PART_OF) • GO:0030193: regulation of blood coagulation (PART_OF)

Figure 5: Representation of ontologies in the Look Up Service

4.2 AUGUSTUS gene prediction

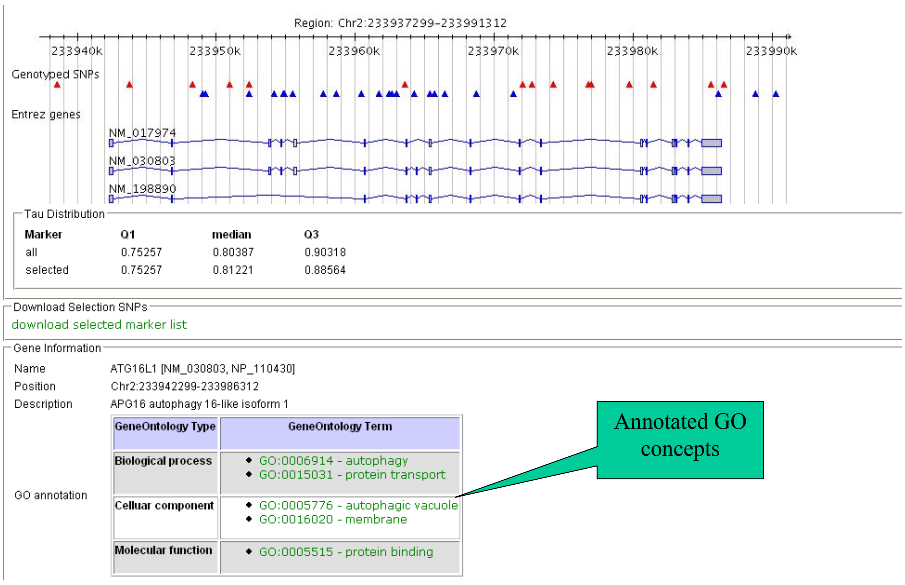
AUGUSTUS is a tool that predicts genes in eukaryotic genomic sequences. Sequences of different species (e.g. human, mouse, fly) are being compared, a result file describing gene locations and further information is produced. The result file is based on the standardized General Feature Format (GFF) [6], which is easy to parse and process by a variety of applications. The format is record-based, i.e. each feature of the result is described on a single line.

Particularly, AUGUSTUS produces a GFF file that describes gene locations and other features annotated with names of the Sequence Ontology (SO). We interlink these feature names with entries in the SO by utilizing the ontology client to search for concepts in the SO. The connection is displayed as a link on the result file in the AUGUSTUS portlet. A click on the link leads the user to a special ontology page displaying detailed ontology information about the used feature.

4.3 SNPSelection

The SNPSelection application uses direct regions (e.g. chromosome positions) or keywords (e.g. gene name or RefSeq ID) and population information to compute an optimal genetic marker set for given constraints. The data for the calculation is located in publicly available databases, namely HapMap release 20 [9] and UCSC (hg17).

We use the parameters of SNPSelection to build a mapping between input IDs and GeneOntology (GO) concepts to annotate the result with semantic information about biological processes, molecular functions and cellular components. For this purpose we integrated the Ensembl data source (www.ensembl.org) for Homo Sapiens as a OGSA-DAI data service in MediGRID. Ensembl supports users with protein data and several associations to other biological databases, in our case RefSeq IDs (ID as registered in the RefSeq database) and GeneOntology concepts. The mapping between RefSeq IDs and GO concepts is computed via the OGSA-DAI service, the result is integrated in the portlet application of SNPSelection (Fig. 6). For further information about the annotated GO concepts, we again interlink the concepts with our ontology access middleware. If a user is interested in detailed information about a concept of GO, the generic ontology client of the ontology access middleware is used to serve further information like a concepts definition or relationship graph. Comparable to AUGUSTUS, a link leads the user to a special page representing the further knowledge.



Tau Distribution

Marker	Q1	median	Q3
all	0.75257	0.80387	0.90318
selected	0.75257	0.81221	0.88564

Download Selection SNPs

download selected marker list

Gene Information

Name	ATG16L1 [NM_030803, NP_110430]
Position	Chr2:233942299-233986312
Description	APG16 autophagy 16-like isoform 1

GO annotation

GeneOntology Type	GeneOntology Term
Biological process	◆ GO:0006914 - autophagy ◆ GO:0015031 - protein transport
Celluar component	◆ GO:0005776 - autophagic vacuole ◆ GO:0016020 - membrane
Molecular function	◆ GO:0005515 - protein binding

Annotated GO concepts

Figure 6: GO annotation in the SNPSelection portlet

5 Summary and outlook

We described a generic, service-based middleware to access ontologies in grid systems. Our ontology services provide grid applications a uniform and transparent access to ontologies in grids. The implementation is based on OGSA-DAI and compatible with the D-Grid infrastructure. New ontologies can easily be added. The usage of several ontology services supports good performance (load balancing) and availability. First applications in the MediGRID community project are using these services, in particular portlets to search and browse biomedical ontologies, the gene prediction tool AUGUSTUS and the SNPSelection portlet.

In the future, we will investigate more complex grid services for data integration and ontologies especially in the life sciences domain. Furthermore, we plan to provide a wiki-like system for collaborative editing and development of domain-specific ontologies.

Acknowledgements

This work is supported by BMBF grant 01AK803E "MediGRID – Networked Computing Resources For Biomedical Research".

References

1. O. Corcho, P. Alper, I. Kotsiopoulos, P. Missier, S. Bechhofer, C. Goble: An overview of S-OGSA: a Reference Architecture for the Semantic Grid. Journal of Web Semantics Vol. 4. 102 –115. 2006.
2. D-Grid: http://www.d-grid.de
3. K. Eilbeck, S.E. Lewis, C.J. Mungall, M. Yandell, L. Stein, R. Durbin, M. Ashburner: The Sequence Ontology: A tool for the unification of genome annotations. Genome Biology 6:R44. 2005.
4. I. Foster, J. Frey, S. Graham, S. Tuecke, K. Czajkowski, D. Ferguson, F. Leymann, M. Nally, T. Storey, S. Weerawaranna: Modeling Stateful Resources with Web Services. Globus Alliance. 2004.
5. The Gene Ontology Consortium: The Gene Ontology (GO) database and informatics resource. Nucleic Acids Research, 32. D258-D261. 2004.
6. Generic Feature Format (GFF): http://www.sequenceontology.org/gff3.shtml
7. M. E. Gutiérrez, A. Gómez-Pérez, O.M. García, B.V. Terrazas: WS-DAIOnt: Ontology Access Provisioning in Grid Environments. GGF 16 Semantic Grid Workshop. 2006.
8. J. Hampe, S. Schreiber and M. Krawczak: Entropy-based SNP selection for genetic association studies. Human Genetics, 114. 36 –43. 2003.
9. The International HapMap Consortium: A haplotype map of the human genome. Nature, 437. 1299 – 1320. 2005.
10. K. Karasavas, M. Antonioletti, M.P. Atkinson, N.P. Chue Hong, T. Sugden, A.C. Hume, M. Jackson, A. Krause, C. Palansuriya: Introduction to OGSA-DAI Services. LNCS Vol. 3458. 1 – 12. 2005.
11. I. Kojima: Design and Implementation of OGSA-DAI-RDF. GGF 16 Semantic Grid Workshop. 2006.
12. MediGRID. Medical Grid Computing. http://www.medigrid.de
13. J. Novotny, M. Russell and O. Wehrens: GridSphere: A Portal Framework For Building Collaborations. The 1st Int. Workshop on Middleware for Grid Computing. 2003.
14. Open Biomedical Ontologies: http://obo.sourceforge.org
15. M. Russell, J. Novotny and O. Wehrens: The Grid Portlets Web Application: A Grid Portal Framework. http://www.gridsphere.org.
16. N. Sioutos, S. de Coronado, M.W. Haber, F.W. Hartel, W.-L. Shaiu, L.W. Wright: NCI Thesaurus: A semantic model integrating cancer-related clinical and molecular information. Journal of Biomedical Informatics Vol. 40. 30 –43. 2007.
17. M. Stanke, O. Keller, I. Gunduz, A. Hayes, S. Waack, B. Morgenstern: AUGUSTUS: ab initio predication of alternative transcripts. Nucleic Acids Research, 34. W435-W439. 2006.