A platform for collaborative management of semantic grid metadata

Michael Hartung[1], Frank Loebe[2], Heinrich Herre[3], and Erhard Rahm[2]

[1] Interdisciplinary Center for Bioinformatics, University of Leipzig, Germany
hartung@izbi.uni-leipzig.de

[2] Department of Computer Science, University of Leipzig, Germany
(loebe,rahm)@informatik.uni-leipzig.de

[3] Institute for Medical Informatics, Statistics and Epidemiology, University of Leipzig, Germany hherre@imise.uni-leipzig.de

Summary. Grid environments, providing distributed infrastructures, computing resources and data storage, usually show a high degree of heterogeneity in their metadata. We propose a platform for collaborative management and maintenance of common metadata for grids. As the conceptual foundation of this platform, a meta model is presented which distinguishes structured descriptions and classification structures. On this basis, the system allows for the user-friendly creation and editing of grid relevant metadata and provides various search and navigation facilities for grid participants. We applied the platform to the German D-Grid initiative by establishing the D-Grid Ontology (DGO).

1 Introduction

Grid computing offers scientists a distributed infrastructure for collaboration and provides massive amounts of computing, storage, and data resources. Such grid initiatives, e.g., the German D-Grid[4], are highly complex and involve many heterogeneous components. They offer resources of different types (e.g., hardware or software resources). Furthermore, these resources belong to many participating organizations, e.g., universities, research centers or enterprises, which themselves have affiliated persons or take part in different grid sub projects representing individual communities such as medicine or physics.

Metadata at varying levels of detail is needed to describe all these grid resources as well as the participating organizations, projects, and persons. Frequently, grid metadata is managed independently in each participating project, i.e., a project is responsible for its specific metadata. This may be appropriate for the management of project-specific or domain-specific metadata, for example, biomedical grid projects typically use life science ontologies for data annotation. On the other hand, there are common types of metadata which apply to all grid projects. Information about

[4] http://www.d-grid.de

projects, grid resources and organizations can be managed in an integrated form, and should be accessible on-line and directly editable for all authorized participating persons and projects. Furthermore, metadata especially about resources should be offered to grid applications and services, e.g., through metadata service interfaces. Providing an integrated access to grid metadata permits projects to better exchange information about their ongoing work. For example, grid participants can more easily notice related work in other projects, so that cooperation can be improved and duplicate efforts be reduced. It is important that a metadata management system offers simple user interfaces for the extension and change of the metadata (usability aspect), since persons of different domains with diverse technical backgrounds (e.g., computer scientists, physicians, or librarians) meet in a grid's virtual organization. We make the following contributions in this paper:

- We propose a simple yet flexible meta model suitable for management of semantic grid metadata including content types for structured information and ontological categorization for content classification.
- We describe a web-based and wiki-like platform using the defined meta model and supporting the collaborative creation and editing of grid metadata. The platform also addresses usability issues such as powerful search, navigation and visualization capabilities.
- An application of our platform is presented, namely the D-Grid Ontology (DGO) of the German D-Grid initiative available under *http://buell.izbi.uni-leipzig.de/dgo*. In particular, we outline the current organization of the semantic metadata.

The remainder of the paper is organized as follows. In Section 2 we describe models for the collaborative management of grid metadata, with a focus on the meta model level. Section 3 presents the model of DGO, while usability features of the platform are illustrated in Section 4. Implementation details are provided in Section 5. Section 6 discusses related work. We conclude with a summary and an outlook on future work.

2 Models of the platform

We build on a three-layered representation of metadata and data (see Fig. 1) differentiating between the following layers: *meta model*, *models* and *instance data*. The model (or schema) is specific to a particular grid or virtual organization, e.g., D-Grid, and prescribes the structure of possible instances and their semantic annotations. The meta model defines the constructs which can be used for defining the models, in particular for describing the structure of instances (content) and the use of ontologies for semantic annotation of instances. In this section we describe the meta model, whereas Section 3 focuses on the D-Grid Ontology (DGO) with its model and instances.

The meta model consists of two main parts, *content types* and *categories*. Content types are used to define the meta information (structure) for instantiable information or content. Categories, on the other hand, are not directly instantiable but serve for a semantic annotation of content, in particular *content items*. Each content item is associated to a particular content type, i.e., a content item instantiates a specific content type of the model. In the following subsections, we describe content types, categories and related aspects in more detail.

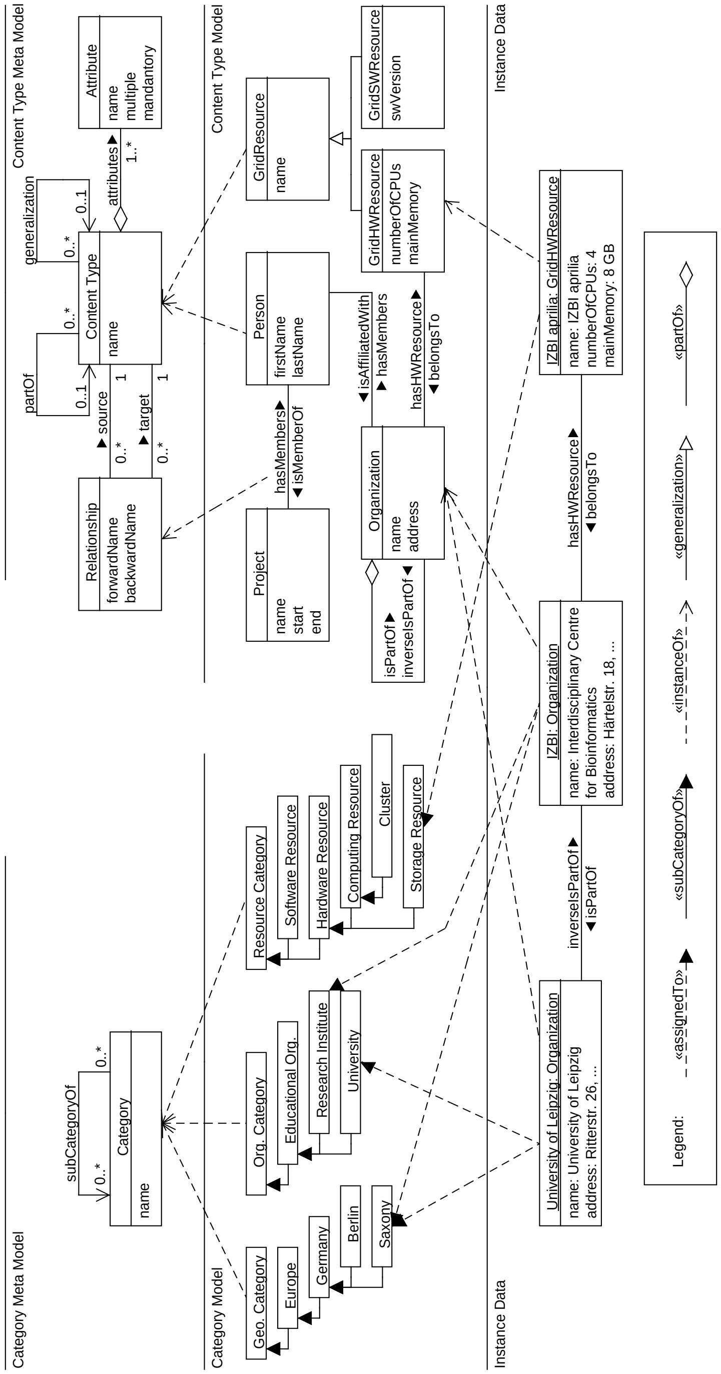


Fig. 1. Three-layered representation of metadata

2.1 Content types

A content type has a *name* and a set of *attributes* describing simple properties for content items. An attribute has a *name*, a data type and a cardinality of one or many. The latter allows for arbitrarily many values of that attribute within a content item. Attributes may also be defined as *mandatory*, i.e., they must be specified during content instantiation (e.g., the first and last name of a person). The attribute's data type restricts the permissible values, e.g., date, URL or string. Furthermore, allowed values can be restricted to a controlled vocabulary to guarantee well-defined terms. We further distinguish between *generic* and *specific* attributes. Generic attributes are predefined and exist for all content types, e.g., the 'ID' and 'Synonym' attributes. Specific attributes describe application-specific properties of content types.

Content types can be interrelated by binary *relationships* of a specified cardinality. Relationships are managed bidirectionally and thus consist of a forward and backward relationship. Hence content items participating in a relationship are accessible from both directions. For instance, assume a content type Person has a relationship with a second content type Organization. When a content item A of Person 'isAssociatedWith' a content item B of Organization (forward relation), we also maintain that B is connected to A through a 'hasMembers' relationship (backward). In order to keep our model simple and flexible, we currently do not use relationship attributes.

In addition to such application-specific relationships we support two general kinds of relationships with predefined semantics: *generalization* and *partOf*. Firstly, content types can be part of generalization hierarchies supporting inheritance. Hence, derived content types reuse the metadata of their predecessors in the generalization hierarchy and may define additional attributes or relationships. The topmost (root) nodes of the generalization relation are called *base content types*. For instance, a base content type 'GridResource' may inherit its attributes and relationships to more specific content types such as 'GridHardwareResource' or 'GridSoftwareResource'. Secondly, the partOf relationship interrelates content types to construct aggregation hierarchies. For example, we use a recursive partOf relationship between organizations. Such partOf hierarchies are used in our platform to support navigation and to specify the context of content items. For instance, we may have several items called 'Department of Computer Science'. Their meaning only becomes clear by considering their predecessors within the organizational partOf hierarchy, e.g., to differentiate between 'University of Leipzig' / 'Department of Computer Science' and 'TU Munich' / 'Department of Computer Science'.

2.2 Categories

Categories have a *name* and are hierarchically organized within *subCategoryOf* relationships. These relationships are assumed to form directed acyclic graphs (DAGs) of categories. Moreover the subCategoryOf relationship involves different semantics depending on what categories are interrelated, e.g., 'Germany' is part of 'Europe' or a 'University' is an 'Educational Organization'. *Roots* are special categories without predecessor for the subCategoryOf relationship and therefore act as entry points of a category structure.

We build on this simple yet flexible category model to broadly support semantic annotations, i.e., the ontological structuring and classification of content items (instance data). Categories can be used to manage content items of different content types *independently* of the content structure. In particular, content items can be categorized along multiple categories. Notably, the associations between content items and categories exhibit the character of annotations (see 'assignedTo' associations in Fig. 1). Such associations may be used in many cases, e.g., to instantiate categories or to associate objects to a geographical category. For example, the content item 'University of Leipzig' may be associated to a 'University' category and a 'Saxony' category.

Categories can be used to improve the navigation within the platform (along the lines of faceted classification) and to support semantic queries. For instance, if somebody is interested in all universities participating in a grid, one navigates through the organization category structure to the university category to see all associated university organizations.

3 Sample application – the D-Grid Ontology

D-Grid started in 2005 as a Germany-wide grid initiative. Its aim is to provide a common grid infrastructure for e-Science projects in Germany and to prove the viability and advantages of grid usage in different scientific domains. D-Grid entails many community projects, e.g., for medical and physics applications, and a common integration project (DGI).

Currently, metadata about D-Grid and its structures is highly heterogeneous and distributed across many websites and project-specific repositories, e.g., information about projects, persons, or available hardware and software resources. Furthermore, there are almost no relations or explicit semantic links between these independently maintained information objects. The goal of our metadata platform is to integrate and semantically categorize this heterogeneous information in a common system and to offer it to all D-Grid participants, applications and interested users. New participants in D-Grid can thus quickly inform themselves about ongoing work in D-Grid projects and the organizations and persons involved. Further, resource providers, i.e., institutes providing hardware or software to the grid, can specify parameters about their resources which may be useful for scheduling and distribution of grid applications. Our platform semantically categorizes its content within a so-called D-Grid Ontology (DGO). It simplifies the manual creation and maintenance of metadata using a collaborative, wiki-like platform. Through the use of the meta model including content types and ontological annotations a high data consistency and quality is pursued.

On the basis of our meta model described in Sec. 2, we use four basic grid content types in the DGO model, namely *Person*, *Project*, *Organization* and *GridResource* (see content type model in Fig. 1). As an example, the content type Person uses attributes such as first name, last name, email or phone number for the registration of personal information. Furthermore, relationships to content items of other content types show a person's semantic neighborhood, e.g., the projects a person is working in ('isMemberOf') or the organization to which a person is affiliated ('isAffiliated-With'). Furthermore, DGO exploits recursive partOf relationships for projects and organizations. In particular, 'D-Grid' is the topmost project of DGO and contains a number of sub projects such as 'MediGRID', 'HEP-Grid' or the 'Integration Project

(DGI)', which themselves include further sub projects. Furthermore, DGO uses several category hierarchies for ontological classification of content items (see category model in Fig. 1). Every content item of DGO is assigned to a minimum of one category. For instance, a community project such as 'MediGRID' is assigned to the category 'Community Project' (in terms of project type) and 'D-Grid I' (funding aspect) since it was funded as one of the starting projects of the D-Grid initiative.

The current version of DGO (as of April 2008) categorizes and interrelates about 40 projects, 150 organizations, 300 persons, and 75 grid resources. There are about 950 bidirectional relationships between content items.

4 Usability features

In the following, we describe some of the features of our platform to illustrate its usability. In particular, we firstly illustrate how semantic metadata is displayed within the platform. Furthermore, we present navigation and search capabilities as well as options for creation, classification and editing of content. For a hands-on experience the interested reader may directly use the system (after registration) under *http://buell.izbi.uni-leipzig.de/dgo*.

4.1 Content visualization

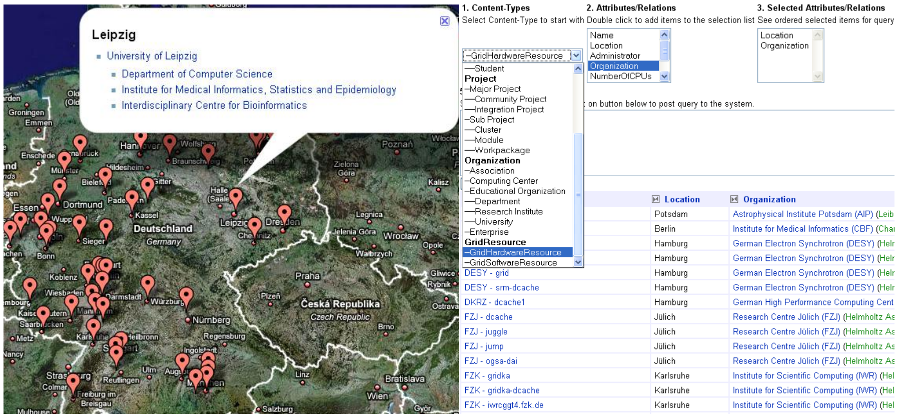
Each content item is shown on its own article page, providing information about its name, basic attributes, relationships, category classifications, explanations (free text), images and versioning. Relationships to other content items are presented as hyperlinks allowing the user to traverse to the content page of the referenced item. Specific tabs allow the direct change of content pages, in particular editing, renaming or category assignment.

Our platform exploits Web 2.0 techniques, such as maps and navigable trees, to display semantic metadata in different forms. In particular, we use Google Maps[5] to geographically locate content items such as organizations or D-Grid hardware resources on a map. For example, users are able to notice what organizations in their local environment also participate in the same grid project and hence regional cooperation is improved or duplicate work can be reduced. Furthermore, we employ partOf relationships between content types to generate trees representing hierarchical structures such as organization or project structures.

The sample map in Fig. 2 (left) includes all organizations currently participating in D-Grid. When selecting a location, e.g., Leipzig, all organizations in this place participating in D-Grid are listed and may be further explored. In order to generate these maps, we utilize location attributes of a content type as well as partOf relationships between content items. Currently, the location attributes represent the city, e.g., of an organization. The geographical coordinates (latitude/longitude) of a city needed for the map visualization is obtained from a publicly available web service[6]. For each location on the map, we use the partOf structure among content items to aggregate all corresponding items for display.

[5] http://maps.google.com

[6] http://www.geonames.org



1. Content-Types
Select Content-Type to start with

–GridHardwareResource

—Student
Project
–Major Project
—Community Project
—Integration Project
–Sub Project
—Cluster
—Module
—Workpackage
Organization
–Association
–Computing Center
–Educational Organization
—Department
—Research Institute
—University
–Enterprise
GridResource
–GridHardwareResource
–GridSoftwareResource

2. Attributes/Relations
Double click to add items to the selection list

Name
Location
Administrator
Organization
NumberOfCPUs

3. Selected Attributes/Relations
See ordered selected items for query

Location
Organization

on button below to post query to the system.

	Location	Organization
	Potsdam	Astrophysical Institute Potsdam (AIP) (Leib
	Berlin	Institute for Medical Informatics (CBF) (Char
	Hamburg	German Electron Synchrotron (DESY) (Helr
	Hamburg	German Electron Synchrotron (DESY) (Helr
DESY - grid	Hamburg	German Electron Synchrotron (DESY) (Helr
DESY - srm-dcache	Hamburg	German Electron Synchrotron (DESY) (Helr
DKRZ - dcache1	Hamburg	German High Performance Computing Cent
FZJ - dcache	Jülich	Research Centre Jülich (FZJ) (Helmholtz As
FZJ - juggle	Jülich	Research Centre Jülich (FZJ) (Helmholtz As
FZJ - jump	Jülich	Research Centre Jülich (FZJ) (Helmholtz As
FZJ - ogsa-dai	Jülich	Research Centre Jülich (FZJ) (Helmholtz As
FZK - gridka	Karlsruhe	Institute for Scientific Computing (IWR) (Hel
FZK - gridka-dcache	Karlsruhe	Institute for Scientific Computing (IWR) (Hel
FZK - iwrcggt4.fzk.de	Karlsruhe	Institute for Scientific Computing (IWR) (Hel

Fig. 2. Organizations of D-Grid on a map (left) and query generator (right).

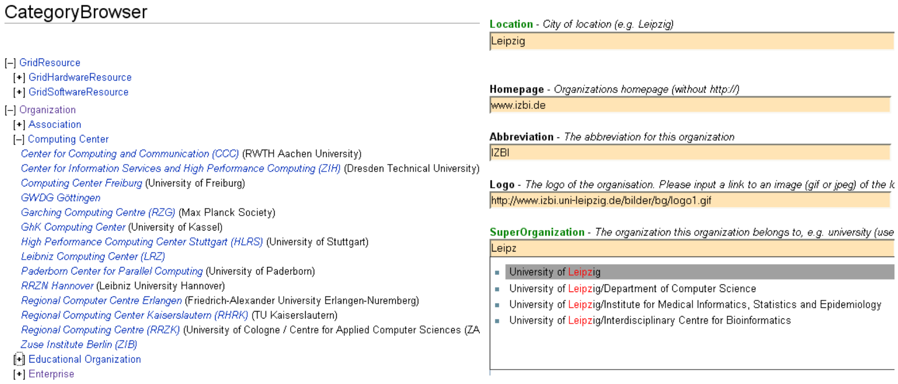
4.2 Search and navigation facilities

The platform provides different search and navigation facilities. A simple text search supports keyword-based search over all attributes of content items. Furthermore, semantic query capabilities on content types and categories are provided. In particular, a query generator (Fig. 2 right) for interactive specification of semantic queries is available so that users can pose powerful queries without having to learn a complex query syntax or query rules. Users choose a specific content type and their attributes or relationships they are interested in. For instance, a query to determine the email and names of all persons working in D-Grid can be generated within a few seconds. The results are presented in tables which can be interactively sorted on different attributes or relationships, e.g., person name or the affiliated organization.

Besides search, the platform provides extensive navigation capabilities for content retrieval. A category browser (Fig. 3 left) enables simple and fast navigation to content of interest. It dynamically generates a navigation tree representing categories and content items in an integrated form, by attaching content items as leaves to their most specific categories. For instance, with some clicks a user can navigate from the top category 'Person' to 'Researcher' or 'Professor' to see all associated content items. All nodes of the tree are linked, i.e., a click on a category displays the corresponding category page with all assigned content items, and a click on a content item shows the article of the content item, respectively.

4.3 Creation and editing of content

For every content type the system provides an interactive input form to create new content items. These forms are dynamically created from the current meta information (attributes, relationships, category associations) of a content type. To change existing content items, the current content attribute values, relationships and category associations are presented for editing within UI forms analogous to the ones for creating new content items (Fig. 3 right).



CategoryBrowser

- [–] GridResource
  - [+] GridHardwareResource
  - [+] GridSoftwareResource
- [–] Organization
  - [+] Association
  - [–] Computing Center
    - *Center for Computing and Communication (CCC)* (RWTH Aachen University)
    - *Center for Information Services and High Performance Computing (ZIH)* (Dresden Technical University)
    - *Computing Center Freiburg* (University of Freiburg)
    - *GWDG Göttingen*
    - *Garching Computing Centre (RZG)* (Max Planck Society)
    - *GhK Computing Center* (University of Kassel)
    - *High Performance Computing Center Stuttgart (HLRS)* (University of Stuttgart)
    - *Leibniz Computing Center (LRZ)*
    - *Paderborn Center for Parallel Computing* (University of Paderborn)
    - *RRZN Hannover* (Leibniz University Hannover)
    - *Regional Computer Centre Erlangen* (Friedrich-Alexander University Erlangen-Nuremberg)
    - *Regional Computing Center Kaiserslautern (RHRK)* (TU Kaiserslautern)
    - *Regional Computing Centre (RRZK)* (University of Cologne / Centre for Applied Computer Sciences (ZA
    - *Zuse Institute Berlin (ZIB)*
  - [+] Educational Organization
  - [+] Enterprise

Location - *City of location (e.g. Leipzig)*
Leipzig

Homepage - *Organizations homepage (without http://)*
www.izbi.de

Abbreviation - *The abbreviation for this organization*
IZBI

Logo - *The logo of the organisation. Please input a link to an image (gif or jpeg) of the lo*
http://www.izbi.uni-leipzig.de/bilder/bg/logo1.gif

SuperOrganization - *The organization this organization belongs to, e.g. university (use*
Leipz

- University of Leipzig
- University of Leipzig/Department of Computer Science
- University of Leipzig/Institute for Medical Informatics, Statistics and Epidemiology
- University of Leipzig/Interdisciplinary Centre for Bioinformatics

Fig. 3. Category browser (left) and editing of content (right).

A UI form for creation or editing of content consists of different kinds of form fields, in particular mandatory fields, autocomplete-aware fields, single- / multivalued fields, category association fields and free text. Mandatory fields reflect mandatory attributes, i.e., they need to be filled out in order to create a new content item, e.g., a person's name. In order to simplify user input and to avoid duplicate entries, autocompletion is utilized in the following way. As soon as a user clicks on an autocomplete field or types some letters into it, value suggestions are offered for selection. For example, an input field capturing a relationship to the content type 'Organization' (e.g., a person's affiliation) suggests organization items matching the input. Furthermore, if an attribute is restricted to a controlled vocabulary, we suggest values matching current entries of such a vocabulary. In order to enter multiple values for an attribute or relationship we utilize multivalued fields with a common separator to separate multiple values. The category association field provides the possibility to assign the current content item to different categories. Here, we again make use of autocompletion to simplify categorization and to guarantee correct category associations. Finally, a free text field allows for entering content not covered by attributes, relationships or category association. The different fields just described are marked with different background colors and labels to improve user interaction and the input dialog.

5 Implementation

The presented platform builds upon a widely used semantic wiki implementation, the Semantic Media Wiki (SMW) [5]. SMW, in turn, extends the MediaWiki[7] implementation, which is also used by Wikipedia. MediaWiki provides a powerful infrastructure for collaborative management of text-based articles. It is also aware of categories and sub categories, but links between articles in MediaWiki are untyped (have no semantics) and search capabilities are limited to simple text searches. SMW introduces semantic properties for wiki articles and thus supports a semantic annotation and enhanced querying of wiki contents.

[7] http://www.mediawiki.org

We extended MediaWiki and SMW in several directions. Firstly, we introduce content types (based on the template feature of MediaWiki) to capture semantic metadata in the form of structured content. Secondly, we introduce bidirectional relationships (on the basis of SMW semantic properties) between content types to automatically maintain referential integrity and to provide better navigation capabilities. Thirdly, we support the use of controlled vocabularies and user-friendly UIs for content creation and change, e.g., autocompletion to avoid duplicates. Finally, we utilize Web 2.0 techniques for novel visualization and interaction options, e.g., dynamic generation of maps for content items and interactive specification of semantic queries.

6 Related Work

Our approach builds upon established wiki technology [6] and its combination with semantic technology, cf. [8, 10]. The initially visible distinction between semantic wikis originating from 'classical' wikis, e.g., the Semantic MediaWiki [5], and editors for knowledge bases or ontologies with wiki-like, collaborative features, e.g., IkeWiki [9] or OntoWiki [1], is currently diminishing [3].

In general, the platform presented herein aims at the collaborative and user-friendly collection and maintenance of structured data. A major difference to other systems concerns our meta model. The meta models of many semantic wikis are based on Semantic Web standards, most often RDF (e.g., WikSAR [2], SweetWiki [3], etc.) and sometimes OWL [7] (e.g., IkeWiki, OntoWiki). In contrast, our meta model supports both a database-oriented and an ontological part. The first comprises multiple content types, relationships and attributes for expressing structured contents. The ontological part provides multiple hierarchies of categories for the classification of content items. These aspects result in a clearly structured system configuration and facilitate a user-friendly access and maintenance of grid metadata. In contrast, the sole use of RDF and OWL models often result in complex graph structures and reduced user friendliness. Another feature of our platform is the bidirectionality of the relationships. This can be considered as a simple form of reasoning which still allows for efficient system behavior. Many semantic wikis avoid the use of Semantic Web reasoning for efficiency reasons (cf. [3, p. 87]; exceptions are e.g. IkeWiki and BOWiki [4]).

As already discussed in the previous section, the presented system utilizes the features of the meta model (content types, bidirectional relationships, categories, controlled vocabularies) for improved consistency and usability, e.g., semantic queries and powerful navigation, visualization and editing (e.g., autocompletion). This is a clear improvement over approaches in which editing of information is only possible in terms of wiki syntax as used for free text editing and markup.

7 Summary and Future Work

We presented a meta model and a platform for the collaborative management of semantic metadata in grids. The platform provides grid participants of large-scale grid initiatives such as D-Grid with a collaborative, web-based and user-friendly way of creating, editing and using grid metadata, e.g., on grid resources, projects, and participating organizations and persons. We applied the platform within the German D-Grid initiative in order to build a semantic metadata repository for D-Grid and to improve the collaboration between participating projects. The platform is currently running under *http://buell.izbi.uni-leipzig.de/dgo* and is actively used by D-Grid members.

In the future, we will extend the platform based on new requirements from the D-Grid communities. We further investigate automatic support of the evolution of the domain model, i.e., changes in the content types and categories (instances with respect to the meta model level).

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

References

1. S. Auer, S. Dietzold, and T. Riechert. Ontowiki – a tool for social, semantic collaboration. In *The Semantic Web – ISWC 2006: Proc. of the 5th International Semantic Web Conference, Athens, Georgia, USA, Nov 5-9*, volume 4273 of *LNCS*, pages 736–749. Springer, Berlin, 2006.
2. D. Aumüller and S. Auer. Towards a semantic wiki experience – desktop integration and interactivity in WikSAR. In *Proc. of the ISWC 2005 Workshop on The Semantic Desktop: Next Generation Information Management & Collaboration Infrastructure, Galway, Ireland, Nov 6*, volume 175 of *CEUR Workshop Proceedings*, pages 212–217. CEUR-WS.org, Aachen, Germany, 2005.
3. M. Buffa, F. L. Gandon, G. Ereteo, P. Sander, and C. Faron. SweetWiki: A semantic wiki. *Journal of Web Semantics*, 6(1):84–97, 2008.
4. R. Hoehndorf, K. Prüfer, M. Backhaus, H. Herre, J. Kelso, F. Loebe, and J. Visagie. A proposal for a gene functions wiki. In *OTM 2006 Workshops [Part I], Workshop Knowledge Systems in Bioinformatics, KSinBIT*, volume 4277 of *LNCS*, pages 669–678. Springer, Berlin, 2006.
5. M. Krötzsch, D. Vrandečić, M. Völkel, H. Haller, and R. Studer. Semantic Wikipedia. *Journal of Web Semantics*, 5(4):251–261, 2007.
6. B. Leuf and W. Cunningham. *The Wiki Way: Collaboration and Sharing on the Internet.* Addison-Wesley Professional, Reading, Massachusetts, 2001.
7. D. L. McGuinness and F. van Harmelen. OWL Web Ontology Language overview. W3C Recommendation, World Wide Web Consortium (W3C), Cambridge, Massachusetts, 2004.
8. D. Riehle and J. Noble, editors. *Proc. of the 2006 International Symposium on Wikis.* ACM, New York, 2006.
9. S. Schaffert. IkeWiki: A semantic wiki for collaborative knowledge management. In *Proc. of the 15th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, WETICE 2006, Manchester, UK, Jun 26-28*, pages 388–396. IEEE Computer Society, Los Alamitos, Calif., 2006.
10. M. Völkel and S. Schaffert, editors. *SemWiki2006 – From Wiki to Semantics: Proc. of the First Workshop on Semantic Wikis, Budva, Montenegro, Jun 12*, volume 206 of *CEUR Workshop Proceedings.* CEUR-WS.org, Aachen, Germany, 2006.