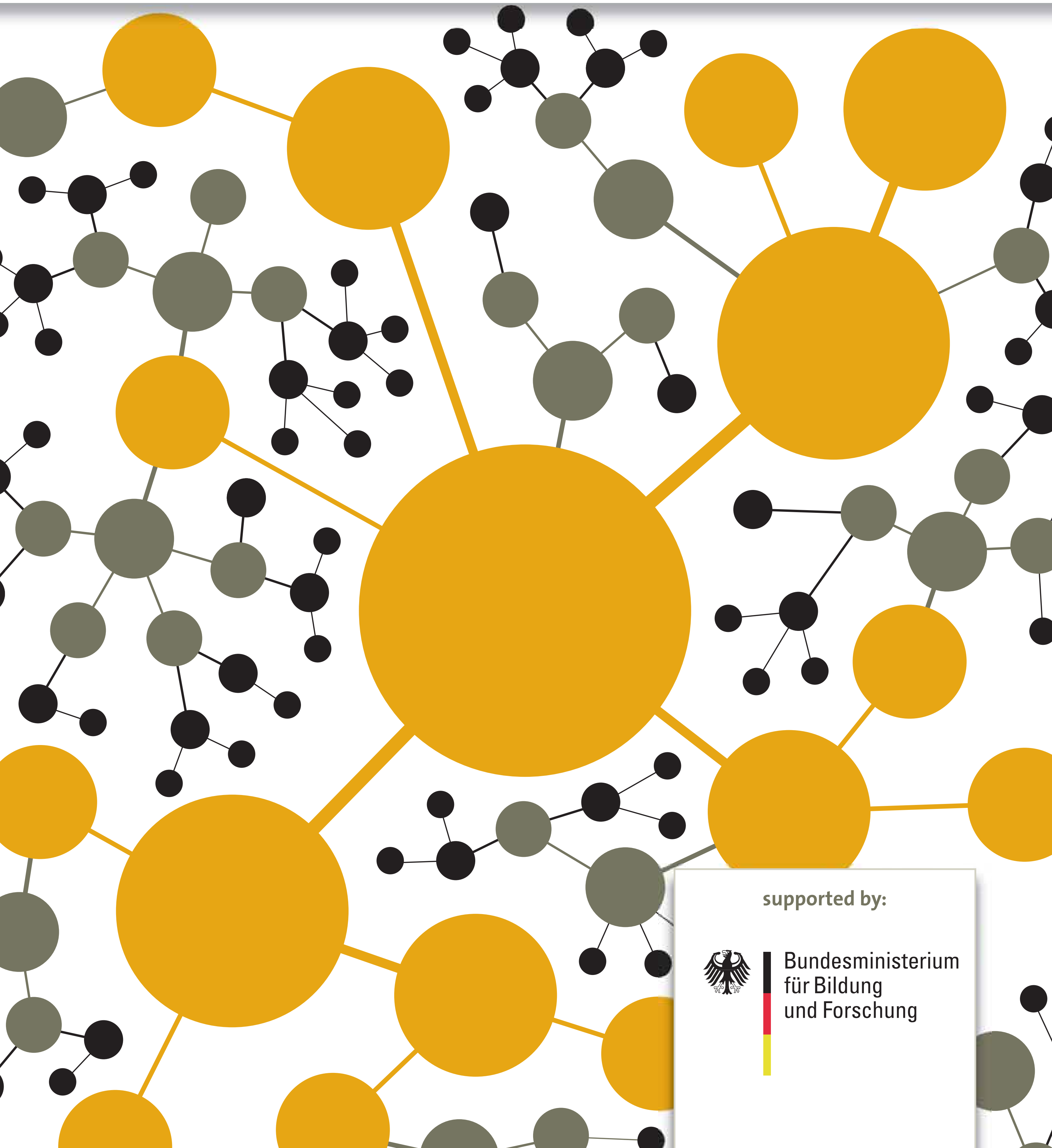
ONTOVERSE

Global Knowledge for Global Challenges



supported by:

Bundesministerium
für Bildung
und Forschung

Foreword

supported by:

Bundesministerium
für Bildung
und Forschung

Prof. Dr.
Arndt von Haeseler

Dear reader,

the booklet you are about to read contains several articles to describe our experiences and our results obtained in the research project Ontoverse, that was a collaboration from different universities and companies. Ontoverse was motivated by the enormous growth of data currently accumulating in life sciences. To cope with such a massive amount of data formal knowledge models – called ontologies – became more and more important especially for data management and retrieval at that time. We realized that tool support for collaborative design and development of ontologies was insufficient. While at the same time new cooperative approaches were emerging, especially in the context of the so-called Web 2.0 that we identified to be a promising tool when applied to ontology development.

Generously funded by the German Federal Ministry of Education and Research (BMBF) during the last three years we investigated the potentials as well as challenges of collaborative knowledge management approaches. As a result of this investigation we designed, implemented and partially tested a novel, web-based ontology development platform. The booklet summarizes the results. The first article "Ontoverse: Collaborative Ontology Engineering in the Life Sciences" introduce the Ontoverse platform. In addition "Architectures for Collaborative Ontology Development", "Collaborative Ontology Developing and Interactive Ontology Merging", "Information Retrieval and Information Extraction in Ontoverse"and "Information-Security Provides Trust Worthiness for the Collaborative Development of Ontologies" describe a variety of technical aspects that had to be solved during the runtime Ontoverse. As collaborative ontology design rests on the active participation of the expert community, we would like to recommend "Collaborative Ontology Engineering needs a Specific Incentive System" and "Marketing the Ontoverse project" especially to all those readers interested in this topic. Last but not least the article"BIO2Me: An Ontology for Bioinformatics" is of particular interest for everyone involved in life sciences, ontology development or ontology-based applications.

In addition to a series of scientific publications, we decided to make our work more accessible to the general readership by publishing this booklet. We do hope that you will find the collection of articles both informative and enjoyable. If you are interested in more informations concerning the Ontoverse project or if you want to try the prototype please visit our project's website www.ontoverse.org.

Finally, I would like to point out that such a huge and collaborative project would not have been successful without the enormous efforts of all the people involved at various developmental stages of Ontoverse. I cannot thank them all, but I would like to give my special thanks to Drs. Dominic Mainz and Ingo Paulsen. They were the driving force to start the project and to finish it successfully.

Yours,

Prof. Dr. Arndt von Haeseler

Content

Ontoverse: Collaborative Ontology Engineering in the Life Sciences

Prof. Dr.
Arndt von Haeseler

Dr. Dominic Mainz

Dr. Ingo Paulsen

Dr. Jochen Kohl

Ontoverse is a research project funded by the German Federal Ministry of Education and Research, that offers an approach within the Ministry's promotional focus on eScience and knowledge management while concentrating on life sciences as the domain of interest.

Introduction

Knowledge networking comprises two different aspects: collaborative knowledge management in communities (human networks) and effective information integration (data networks). Thus, on the one hand techniques are regarded that help to structure and interlink existing knowledge sources effectively with ontologies as the core technique. And on the other hand, in knowledge networking people share their knowledge via social networks.

The Ontoverse project aims at combining these two points of view: establishing a platform, that provides tools for designing ontologies for annotating and interlinking knowledge sources (ontology platform) and that also helps people to build up social scientific networks (web platform). The whole platform is called the Ontoverse platform (web platform + ontology platform). It comprises support for collaborative ontology engineering, an ontology based publication management system and solutions for knowledge exchange in scientific communities.

The project partners within the Ontoverse project were chosen regarding their competences in the fields of life sciences, bioinformatics, knowledge interaction, IT security, computer linguistics, information science, innovation research, project management and marketing. Furthermore, partners with commercial and non-commercial backgrounds are involved in the project to contribute concrete user-specific views on planned applications.

Need for Collaborative Ontology Development

Recently, the optimization of storing, retrieving and integrating data is becoming a popular focus for the WWW in general and a fundamental task for scientific contexts. For the focused domain of interest, the life sciences, the particular problem is the integration of heterogeneous data. For example, bioinformatics data not only consist of customary textual items (scientific publications), but also comprise nucleotide sequences, amino acid sequences, 3D structures of molecules and a manifold of other experimental results. Such diverse biodata need to be stored and structured effectively. Recent progress in the life sciences has already led to the accumulation of biodata that now demand classification, accessibility and visualization.

A shared understanding of a domain is needed as a basis for scientific discussions. If no consensus on a domain of knowledge and its key components exists, definitions have to be mentioned in every single discourse on that topic. Thus, ontologies form the basis for communication within a community as well as for human-machine interaction. It is most desirable to have ontologies produced collaboratively by a large community, to ensure that they do indeed represent a shared view. Opinions and suggestions of a broad community of domain experts should be regarded.

Additionally to the management of the vast amount of biodata there is also an urgency to collect scientific cognitions and make them multidisciplinarily accessible. This addresses a main problem in genetics which is the multiple denotation of similar genes which were found in different organisms. This problem leads back to the rapid increase of knowledge. Scientists often specialize on single research domains, for most of them it is hard to keep track of all new developments even in these limited areas, even harder it is to recognize trends in other fields that might effect ones own research. Because of these advantages a growing interest in ontologies can be noticed especially in the bioscience community, resulting in different ontology projects.

Ontoverse Approach

The Ontoverse approach implements an editor platform that supports distributed work on structural (ontological) data as well as informal discussions and annotations (proto-ontological data). Interested users can view and use ontologies, join existing ontology projects or plan and start a project anew. All different phases of ontology development like conceptualization, editing, maintenance and reuse are supported. This also connotes, that Ontoverse is a platform for multiple ontology projects that have to be administered – typically by project administrators (PAs) – diligently and provided in an easily accessible way.

User Community and Collaboration

Ontoverse explicity supports a social network closely combined with a Web-based ontology editor. A focus is placed on the support of a heterogeneous community. Potential users differ in their fields of interest and skills: On the one hand knowledge and expertise is needed from domain experts (DEs). On the other hand ontology languages can only be fully exploited by ontology designers (ODs).

Major innovations in Ontoverse are (see also Figure 1):

- An open collaborative approach that takes into acount all the people who have expert knowledge in a certain knowledge area. The system brings together ODs and DEs and regard their different states of knowledge.
- All different phases of ontology development are supported. The web platform encourages thematic discussions and the adding of unstructured (proto-ontological) data in the actual ontology editing process.
- Ontology debugging, versioning and certification – enabling trustworthy use of ontologies in open and closed professional environments.
- Integrating the publication management system PubDB to store and manage thematic documents and adding information extraction (IE) functionalities. Scientific publications are one kind of knowledge source to gather information about a domain. A publication database is used as the knowledge source for an IE application that supports the community in identifying relevant concepts, instances and relations from texts, which can then be incorporated into the ontologies. In return, the newly developed ontologies themselves help to retrieve relevant documents from publication databases.
- Ontoverse members can also tag arbitrary publications with their own keywords. These tags are mapped whenever possible with the concepts, relations and individuals in the ontologies. Doing this the platform facilitates social search and DE identification.
- Identification of DEs whenever needed by the OD during the editing process is supported. Registered DEs can define expertise in their profiles, which are then used to identify fitting DEs in cases of support requests by ODs.
- A project’s ontology is subject to successional changes. To enable periodical consistent and stable releases the system incorporates a release process. If project members decide that it is time for a new release the current state of the ontology is copied into a debugging branch.

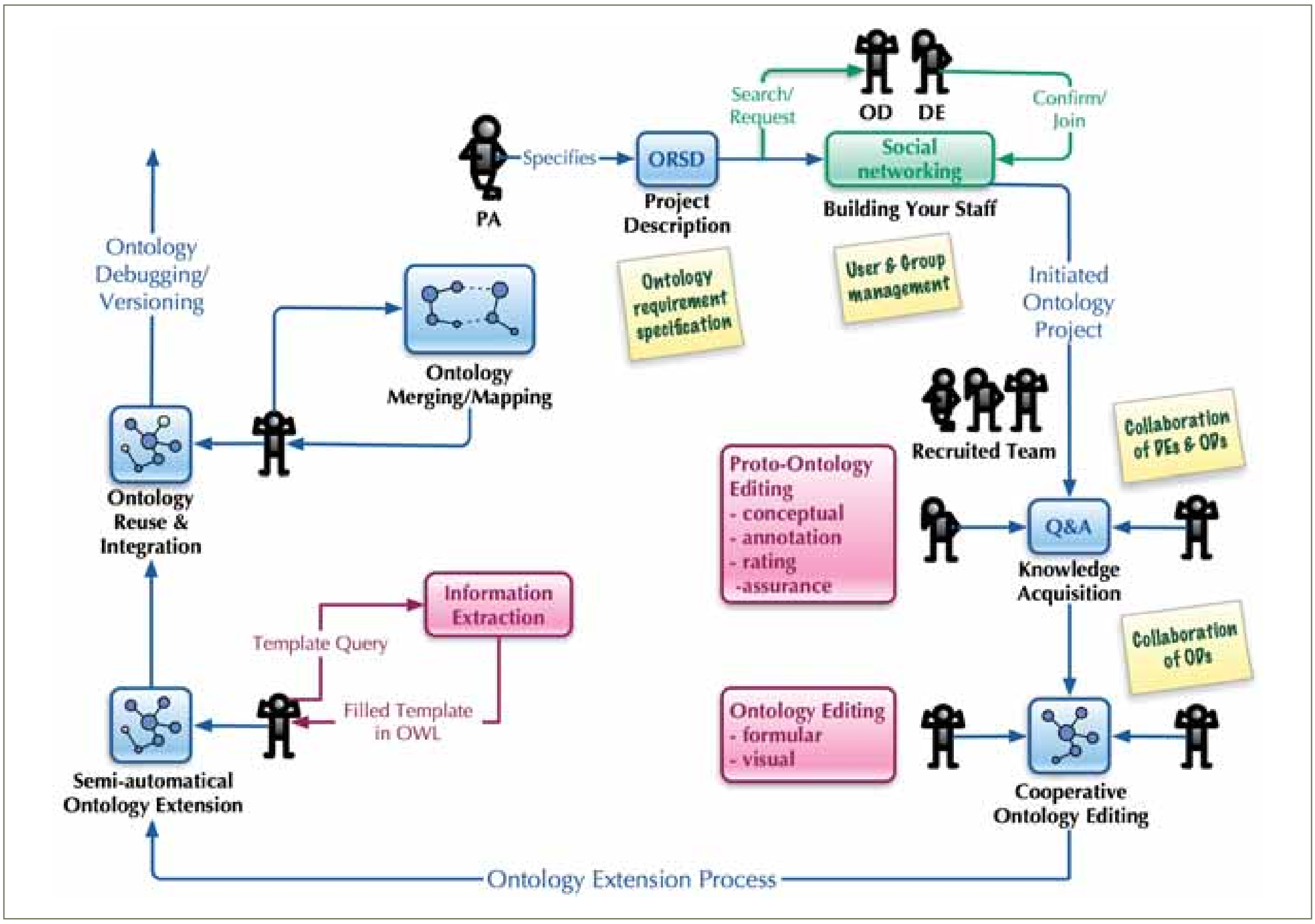


Figure 1: Schematic illustration of the Ontoverse ontology life-cycle process.

Ontoverse Web Platform

In the following all parts and functionalities of the Ontoverse web platform are listed:

- The web platform allows users to create (certified) user accounts and add profiles about themselves: This requires them to log in with a username and a password.
- News/Events blog: This allows editors of the site to create news reports or events and publish them on the front page.
- Discussion forum system: Forum moderators are able to create a number of forums – general or project-specific – in which users can create new topics. Each topic can have any number of posts.
- Blogging engine: This allows users to create their own blogs about their projects and development experiences. It allows users to post blog entries using desktop blogging clients as well as the web.
- Photo gallery for each user of the site: This allows users to upload photos to their profiles or to relevant projects.
- E-mail newsletter: The newsletter can be sent to all members of the site that opt in to receiving e-mails from the site.
- Contact system: This system allow users to add other users to a contacts lists or other types of relationship.
- Tagging and searching support: Projects, user's expertise, and publications are taggable to make it very easy for users to search and browse these objects.
- Geographical maps integration: Opens up the possibilities of embedding interactive, scrollable maps for project members.
- Ontology projects and project's wiki: The building parts and connection to the other architectural parts, especially the wiki as connection to formal ontological data.
- Scientific publication database PubDB: The source for project-specific document collections mainly to extract information.
- Messaging: Adds private messaging between the users on the Ontoverse site, keeping them in-site instead of leaving to check their email.

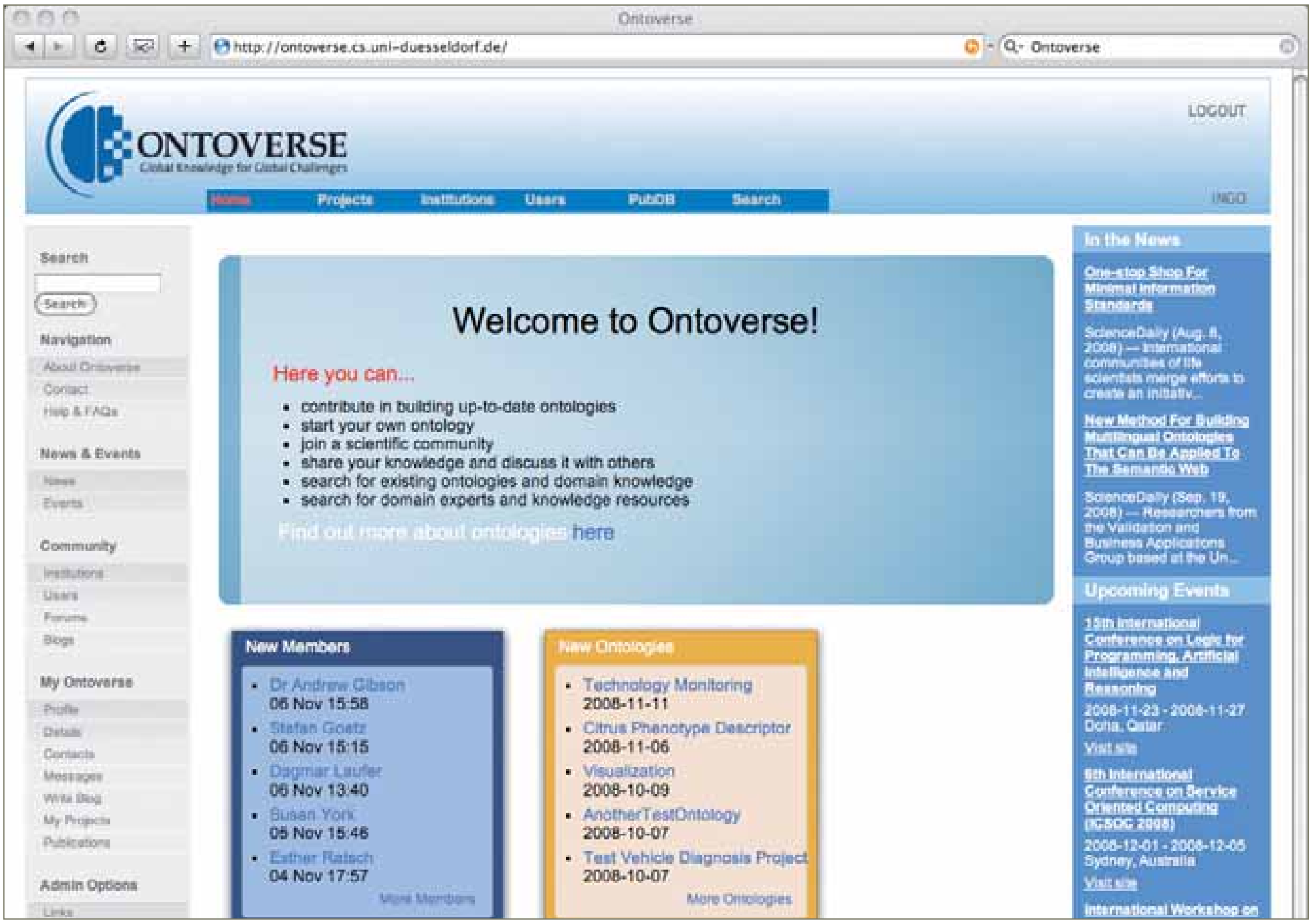


Figure 2: Main page of the Ontoverse web platform.

Conclusion and Outlook

The features described in this article are essential for a collaborative ontology development framework, but the project partners are looking for the necessaty founding to extend and to improve Ontoverse. In particular, further work needs to be spent on sophisticated awareness mechanisms in the user interface to make the ODs and DEs aware of other people working in the same field. Furthermore, the integration of information extraction results into ontologies should be further extended. In order to achieve the common goal, Ontoverse is a platform for collaborators to work and share perspectives, to view common work, and to interactively evaluate and critique each others' contributions.

Authors

Prof. Dr. Arndt von Haeseler (arndt.von.haeseler@univie.ac.at)
acts as the scientific coordinator of the Ontoverse project. Since September 2005 he is the scientific director of the Center for Integrative Bioinformatics Vienna (CIBIV) and has a full professorship for Bioinformatics.

Dr. Dominic Mainz (mainz@cs.uni-duesseldorf.de)
is co-initiator of the research project Ontoverse and works at the Bioinformatics Department, Heinrich-Heine-University Düsseldorf. His research focus is on semantic technologies to process ontological data, ontology editors and description logic programming.

Dr. Ingo Paulsen (paulsen@cs.uni-duesseldorf.de)
is co-initiator of the research project Ontoverse and works at the Bioinformatics Department, Heinrich-Heine-University Düsseldorf. His research focus is on Semantic Web technologies, concurrent programming, and biological databases.

Dr. Jochen Kohl (kohl@cs.uni-duesseldorf.de)
is researcher at the Bioinformatics Department, Heinrich-Heine-University Düsseldorf. He is member of the Ontoverse research project and is particularly engaged in the database-supported management of ontologies

Prof. Dr.
H. Ulrich Hoppe

Dipl.-Inform.
Nils Malzahn

Dipl.-Inform.
Stefan Weinbrenner

Architectures for Collaborative Ontology Development

Introduction

The vision of the Semantic Web introduced by [1] has always been tightly connected to ontologies: By the use of formal, structured and machine-readable knowledge sources in the background it is intended to facilitate intelligent, context-sensitive services, so that these services can act like they insightfully interpret or understand human needs. Although ontologies were the focus of scientific research long before the idea of the Semantic Web emerged, especially in the area of knowledge representation, this movement made them much more popular than in the old days of "Artificial Intelligence".

Therefore also the issues around creating and managing such an ontology are of great interest. The major issue is already addressed by the well-known definition of "ontology" [2] with the later addition of [3], which defines an ontology as a shared conceptualization. The word "shared" indicates, that an ontology typically represents the result of a collaborative process, not the abstract view of an individual. In fact, an ontology usually evolves from a series of brain storming discussions, competitions, proposals, votings and other collaborative activities, which may span over a longer period of time. Additionally automatic knowledge extraction from documents or web sources can provide input, still to be judged by humans. The product should represent the consensus of all participating parties about the targeted domain. However, this necessary collaboration support during the creation and evolution of a common knowledge base represents a major challenge.

Different approaches have been taken to tackle this issue. For instance, there is a collaborative version of Protégé named Collaborative Protégé[4]. This client-server architecture allows clients to access ontologies on the server synchronously. However, Collaborative Protégé does not provide explicit features supporting asynchronous cooperation. Even the editing process itself lacks some features: All editing is done either completely serialized or completely synchronously on one data store. It is not possible, to have a private workspace or even several different versions of one ontology.

Other approaches use wiki systems, as does OntoWiki[5]. As the name suggests their interfaces are similar to a wiki system. It is possible to create, delete and edit ontology entities by using form based web pages. However, if a web-based system has its data stored in a database, it is not automatically "collaborative" in a more specific sense. To support collaborations it is not sufficient to enable two users to access the same data synchronously, but also to support the awareness of other users' actions, and to enable users to discuss changes and to exchange resources and opinions.

A variety of other semantic wikis exist, such as Semantic MediaWiki, Platypus, and Rhizome. These wiki platforms are extended to enrich the (mostly) textual content with semantic metadata. However these approaches are focused on the contribution of content. They apply an ontology, but they do not directly support ontology development.

Aiming at combining advantages of different existing approaches and at providing additional collaborative features, the Ontoverse project provides a platform focusing on the support of communities consisting of domain experts as well as ontology designers to collaboratively design, create and manage ontologies in an easy, convenient and flexible way. The community members may discuss, share and exchange rich information objects. This collaboration comprises functionalities such as user awareness, concurrency control, instant information exchange etc. In order to support such a collaborative working environment, both a flexible architecture embedded into a robust and stable framework connected to an appropriate user interface is needed.

Features of Ontoverse's Backend – SWAT

- Is based on an extensible agent-based architecture
- Provides support for complex ontologies
- Provides a framework for social awareness and content awareness
- Allows for private and shared ontology editing spaces
- Provides means for concurrent versioning of ontologies
- Emulates the Jena standard API
- Comes with interfaces to pellet to build sound ontologies
- Supports a programming platform independent client protocol with clients for Java, Ruby, and SWI Prolog already available
- Integrates a trust center connection that can be used to protect intellectual property.

Architecture and Backend

The basic architectural decision of the Ontoverse backend was to use a blackboard approach. From a distributed systems point of view, the basic idea of the blackboard approach consists in replacing point-to-point communication by publishing messages onto the blackboard (broadcast). To make the public information easily shareable without detailed protocol definitions it is important to rely on simply structured data formats. Blackboard architectures can be implemented using the TupleSpaces approach. A TupleSpaces system consists of one server (the blackboard), that holds all data fragments as tuples, i.e. as ordered lists of primitive

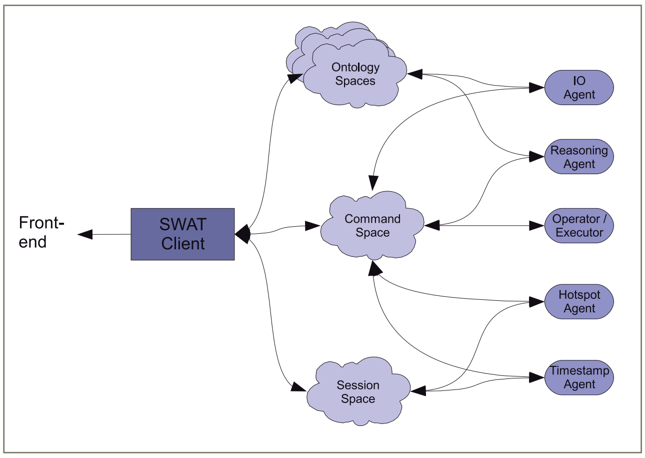


Figure 1: Sketch of SWAT's agent-based architecture

data, which can be accessed using associative query templates. We chose to use an implementation called SQLSpaces[6], which has been developed by the Collide Research Group at the University of Duisburg Essen, Germany. The SQLSpaces have advantages over other existing implementations such as JavaSpaces from Sun Microsystems or TSpaces from IBM, e.g. in terms of enhanced persistence (due to the underlying relational database, which may be MySQL, HSQLDB or PostgreSQL), versioning support and multi-language support. The whole blackboard can be separated into different subspaces, if needed.

Based on this SQLSpaces system we developed an agent system called SWAT (Semantic Web Application Toolkit). The specific division into spaces of SWAT is shown in figure 1. In the center of the diagram the three spaces can be seen. In the ontology spaces (for each ontology one space) the actual ontological data is stored in form of RDF triples. The session space contains all process-related data like login/logout events, creation/modification/deletion events, lock events, etc. This space is also used to map every action in the system to the acting user. Together with the Trust Center and timestamp service it ensures that every user of the system may legally proof his or her claim on particular contributions to the community.

The third space is the command space, which acts as a coordination channel for all participating agents. Figure 1 shows the currently implemented agents. Besides the the IO agent, which is responsible for importing and exporting ontologies to and from OWL XML files, there are other agents that are responsible to keep the system running. They take care that the user service agents (like the Hot Spot agent or the reasoning agent) are restarted if they may fail. So the platform is self-maintaining in a way and since the agents are working independently from each other the failure of one agent does not influence the services of others. As already mentioned there are programming interfaces to several programming languages (currently Java, C#, Ruby, PHP, Prolog), so it is also possible to implement agents in different languages, so that each agent is implemented in the language that fits best the needs of the task (and possibly the programmer).

The following agents are implemented and working:

Hotspot Agent Calculates the hotspot values of all nodes of an ontology. It listens to actions in the session space and writes or updates these hotspot tuples into the session space.

Reasoning Agent Uses an external reasoner (Pellet) to analyse the consistency and other characteristics of the particular ontology.

TimeStamp Agent Adds timestamps to all logged actions on the ontology.

IO Agent Imports and exports OWL files. It reads commands from the command spaces, reads or writes in the ontology spaces and writes the result back to the command space.

Operator & Executor Agents System agents maintaining the overall running of the platform.

Finally on the left side of the diagram the programming interface to the front-end is shown. The SWAT Client encapsulates all features of SWAT in an easy accessible Java class. This includes not only the agents' features and their management (starting, stopping, changing properties), but also e.g. the translation of the RDF triples into easy manageable OWL related Java objects. It is a kind of proxy to make the whole SWAT platform as transparent as possible to the front-end system that is accessing it.

Collaborative Ontology Editing based on SWAT

If a group of ontology editors wants to work together over some period in time and distance in space, they will encounter the problem of modifying the same ontology resource at the same time from different places. How to control concurrency is always a key problem for collaborative work.

Locking an ontology resource before other users start to edit it will prevent such problems. In Ontoverse we use a locking mechanism to control concurrency and prevent the generation of conflicts in synchronous (i.e. if the users participate in the same session at the same time) editing sessions. When a user starts to edit an ontology resource, a lock will be set on the resource. A lock event will be triggered in the same time and will be spread to each client which is in the same session. Once a lock is set to the resource, all other users cannot access any edit function, such as adding a new super class or editing a comment. There are two different ways to set a lock on a specific ontology resource. One is so-called 'Automatic Setting'. In this case a lock will be set automatically if the user modifies anything on a resource. This lock will be released, if the user acts on another resource. Another option is the so-called 'Manual Setting'. In this case users set a lock on one or several ontology resources manually. These locks are held until the user releases them manually or the user's client shuts down. Manual locking is useful if a user wants to make a 'cluster' modification on a set of ontology resources.

Group awareness keeps the team members informed of each other's working status. So some of the potential conflicts can be avoided. The Ontoverse backend offers multiple awareness features:

Once a user modifies an ontology resource, all other users who are in the same session will automatically be notified immediately. So the team notices what has been edited by whom. There is also the opportunity to highlight an ontology resource e.g. to hint at a specific concept during a discussion in a chat. Last but not least the open architecture of Ontoverse offers the capabilites for a simple chat application to exchange short text messages during synchronous editing sessions.

While these capabilities offer an adequate support for synchronous sessions there are also functionalities to support asynchronous cooperative work. The SWAT-Framework with its underlying SQLSpaces provide a concurrent versioning system. Thus there can be multiple different versions of an ontology. The users may work on their own version of the ontology either to experiment with it or to propose another way of structuring the field. Furthermore each version may be shared with particularly invited users or with the whole community (mostly for releases). In addition to this a simple merging facility is provided. If two versions of the same ontology (both created on the Ontoverse platform) are selected, they may be merged. This is done by trying to combine the different user actions in a way that there are no conflicting actions. If there are conflicting actions, they are identified and presented to the user who tries to merge these two versions.

In addion to this event awareness the SWAT framework provides the Hot Spot agent. In communities there is often the notion of a current topic that is interesting for the majority of the community at the moment. In online communities it is often not easy to tell which of the currently discussed issues is the most important one (measured by activity!). While forum based communities may apply quite simple and well-known means to get a feeling for the current subject-matter quickly, e.g. the length of a discussion thread or the amount of users participating in this thread are simple indicators for the „hotness" of a topic.

However, in the case of communities that create ontologies collaboratively and asynchronously being aware of the community's current preferences is not that easy. As stated above in the case of a forum the hotspots can be found by taking the number of replies per thread into consideration. This can be mapped to ontologies, where the actions per ontology component are simply counted. These actions comprise creation, modification and deletion of classes, instances and properties. However, the comparison between a forum and an ontology is missing the important fact, that forums usually have either only one or at least a quite small, fixed number of hierarchical levels, where the threads always form the bottom level. In ontologies the hierarchy is on the one hand often much deeper, much more flexible (i.e. not all leaves of the tree are on the same level, if it is a tree at all) and also hierarchical crosslinks and loops are possible. Another problem related to this is the problem, that actions not only belong to leaves, but also to nodes in between.

So the first question is, what actually can be the focus of a hotspot? We think that the main interest in finding hotspots is the class hierarchy, since this is mostly the actual "space" where the user usually starts working. In the Ontoverse system the hotspot calculation is done by an agent that frequently updates the hotspot value for each ontology node. So whenever a user looks at the class hierarchy, the current hotspot values are presented to the user.

References

1) Berners-Lee, T., Hendler, J., Lasilla, O.: The Semantic Web. Scientific American (May 2001)
2) Gruber, T.R.: A translation approach to portable ontology specifications. Knowl. Acquis. 5(2) (1993) 199–220
3) Borst, W.N.: Construction of Engineering Ontologies for Knowledge Sharing and Reuse. PhD thesis, University of Enschede (1997)
4) Noy, N.F., Sintek, M., Decker, S., Crubezy, M., Fergerson, R.W., Musen, M.A.: Creating Semantic Web Contents with Protégé 2000. IEEE Intelligent Systems 16(2) (2001) 60–71
5) Auer, S., Dietzold, S., Riechert, T.: Ontowiki - A tool for Social, Semantic Collaboration. In Cruz, I.F., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L., eds.: International Semantic Web Conference. Volume 4273 of Lecture Notes in Computer Science., Springer (2006) 736–749
6) Giemza, A., Weinbrenner, S., Engler, J., Hoppe, H.U.: Tuple Spaces as flexible integration platform for distributed learning enviroments. In: Proceedings of ICCE 2007. (November 2007) 313–320

Authors

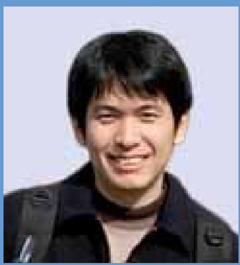
Prof. Dr. H. Ulrich Hoppe (hoppe@collide.info)
is head of the Research Group for Collaborative Learning in Intelligent Distributed Environments (Collide) at University Duisburg Essen, Department of Computer Scinece and Applied Cognitive Science. His research area encompasses distributed collaborative environments and intelligent support in educational and community oriented systems.

Dipl.-Inform. Nils Malzahn (malzahn@collide.info)
is member of the Collide Research Group at the University Duisburg Essen. He has worked on the architecture of Ontoverse. His current research focus is on intelligent community support and heterogeneous system architectures.

Dipl.-Inform. Stefan Weinbrenner (weinbrenner@collide.info)
is member of the Collide Research Group at the University Duisburg Essen. He is the main developer of the Ontoverse backend. His current research focus is on collaborative process support in general and on integration of heterogeneous applications and data.

Collaborative Ontology Developing and Interactive Ontology

Prof. Dr.
Jürgen Ziegler



Fan Bai

Zoulfa El Jerroundi

Introduction

As a formal, structured and machine readable knowledge sources the ontology makes it possible to provide intelligent, context-sensitive services to the human users, as if computers use their own mind to take care of the human problems. Although ontologies were the focus of scientific research long before, the movement in the area of knowledge representation made them become much more popular in the last years than they have ever been in computer sciences.

Ontology developing and managing usually evolves a series of brain storming discussions, competitions, proposals, voting and other collaborative activities. An efficient ontology development environment should support several collaborative activities. There are some obvious and important facts to be met, including: synchronous and asynchronous information exchange, concurrency control and group awareness etc. Beside these collaborative supporting, a good ontology editor should support a mechanism to map and merge different ontologies or different versions of an ontology. With the assistances of the system users could easily and intuitively compare and merge ontologies. In the Ontoverse project, we developed an ontology editor, which supports collaborative process of ontology development and offers functionalities to support interactive ontology comparing and merging. It uses Java Applet technique and use SWAT as its backend. We will discuss visualization and collaborative functionalities in the second section, and discuss interactive ontology mapping and merging in the third section

Collaborative Ontology Development

The editor uses SWAT as its backend, which enables users to construct, edit and manage ontologies in a collaborative working environment. The editor is embedded in the Web page so it does not need to be installed in the local machine. In developing this editor there are several visualization and collaboration problems we have to face, such as how to express an overview of an ontology's hierarchical structure with enough detail information, how to control concurrency, how to support group awareness and exchange synchronous/asynchronous messages between users, etc. For this purpose we developed an extension component of Swing's JTree, called SmartTree to express some detail information of an ontology concept in the hierarchical structure and define a locking mechanism to support concurrency. Furthermore our applet also supports group awareness by immediately showing other users actions in the interface. We will discuss these functionalities separately.

SMART Tree [1], as shown in figure 1, is an extension component of Java's JTree component. In the SmartTree component there are some new features to express hierarchical structures more isolationistic and clear. Usually ontology will be represented as a concept hierarchy such as the ontology class or the property hierarchy. However this representation only gives users the overview of the parent-children relationship. For an ontology class, other relations such as its properties and owl:equivalentWith cannot be seen in this structure. Furthermore those relative concepts cannot be expressed

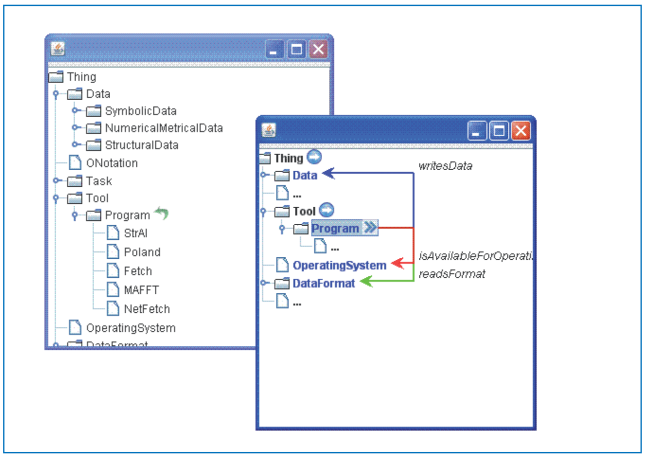


Figure 1: SmartTree shows relations and highlight relative classes.

and users do not get an overview impression on which other concepts are related with the currently selected concept. For example in ' Bioinformatics Ontology for Tools and Methods.owl' the `Program` class has a property `writesData` whose range is `Data`. In SmartTree these non-hierarchical relations can be expressed and users can get a more detailed overview of these concepts. SmartTree also offers a functionality which enables users to easily keep an eye on concepts they are interested in regardless of the amount of other concepts in the component by condensing or `shrinking` other concepts from the tree structure. For example users can only trace those classes, which have relations with `Program`. Some main features of `SmartTree` are: Express non-hierarchical relations, shrink/spread tree nodes and focus nodes by relations.

Locking an ontology resource before users start to edit will prevent other users from editing the same resource at the same time. How to control concurrency is always a key problem for collaborative work. In Ontoverse we use a locking mechanism to control concurrency and prevent the generation of conflicts. When users start to edit an ontology resource, a lock will be set on the resource. A lock event will be triggered in the same time and will be spread to each client which is in the same session, then we will display an icon in the GUI to notify other users of the lock events. Once a lock is set to the resource, all other users cannot access any edit function, such as adding a new super class or editing comment. All these functions are disabled in the application's user interface. There are two different ways to set a lock on a specific ontology resource. One is so-called `Automatic Setting`. In this case a lock will be set automatically if users modify something on a resource when there is no lock currently on it. This lock will be held until users change to another resource by selecting another resource in the tree hierarchy and the lock will be released automatically. Another one is so-called `Manual Setting`. In this case users set a lock on one or several ontology resources by clicking a lock-setting button in the GUI. These locks are obtained by certain users and will be held until the user clicks the lock setting button again to release them or the user left the application. This manual locking is useful for a user who wants to make a `cluster` modification on a set of ontology resources and does not want anyone else to modify them before he or she finishes these bunches of modifications.

Group awareness supports team members being informed of each other's working status and is critical to successful collaboration. Our application offers several ways to keep team workers tied together closely. Figure 2 shows how two users are notified about other users action. Once a user modifies an ontology resource, all other users who are in the same session will be notified immediately. For example when a user created a resource, other users will see this new resource in the tree taxonomy in their own application at once. An icon will be displayed on the resource in the tree taxonomy to indicate that someone has modified this resource. This icon will be deleted if the user selects this resource to see detailed information. This awareness mechanism gives an `after the event` way for users to notice what other team members edit the ontology. The application also supports a `before the event` way. We call this `highlighting` an ontology resource. Users highlight an ontology resource by selecting it and clicking on the `highlight` button. In the other team members' applications this ontology resource is shown immediately and a flashing exclamation mark will be displayed next to the resource in the tree taxonomy to catch the user's attention. If the resource is not visible in the tree taxonomy since either the resource is collapsed or the position is outside the currently visible area, the application will expand its parent and scroll the visible area to make sure that other users can see the resource. After the attention is caught by the `highlighting`, usually we need a conversation to talk about it. Therefore the application also offers an embedded instant messaging tool. Using it users can realize which other users are working with them now and chat with each other to exchange their opinion on some resource.

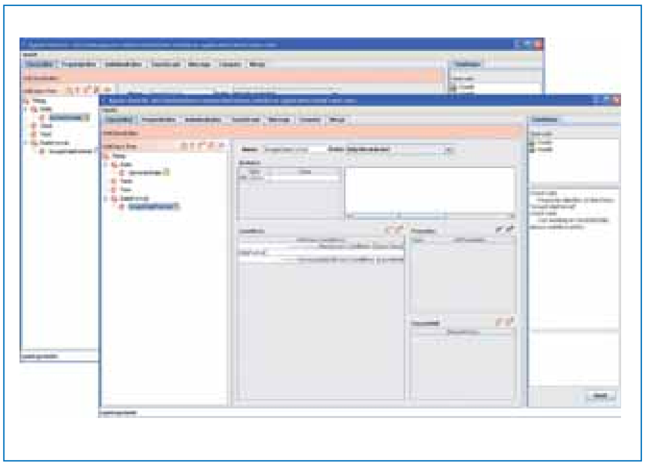


Figure2: shows how group awareness is working in the application

Interactive Ontology Mapping and Merging

The areas of ontology mapping and ontology merging have largely relied on automatic and semi-automatic methods in the past, where user control and interaction is limited and results are typically only presented to the user at the end of some complex computational process. The user could not trust in automatic results, where he does not know where and for which reason concepts are merged. In our editor we present an approach, which helps to enable users to explore the ontology and to compare results in an intuitive and efficient manner. In this paper we present an approach, which helps to enable users to explore the ontology and to compare results in an intuitive and effcient manner. We aim to support the analytic comparing and merging process providing tightly linked and integrated techniques and views for visualizing and exploring the raw ontologies and derived merging results.

The SmartTree-View
extends the conventional tree widget with a number of mechanisms facilitating ontology exploration and development.

The Matrix-View
is suited for comparing two ontologies and determining where most of the mappings between ontologies occur. In the Matrix-View the ontologies to be compared are confronted on the both axes of the matrix. High agreement in the ontologies is signified with green symbols at junctures, parts which are different are signalized with red symbols. A plus symbol in the matrix indicates that there are similar concepts hidden in the substructure. For the comparing process the user can select different algorithms. Based on the results of comparison the ontologies can be merged.

The InteractiveMERGE-View
supports merging of two ontologies step by step and with leverage the users knowledge and expertise. For supporting this task, both ontologies are highlighted in different colors, so the user can register from which parts the changes comes from. The differences are shown first in the original ontology to the user and after that the consequences of the merging step are shown visually in the target ontology. The domain expert can accept, change or even reject this step. The alternative views ease the ontology designers to comprehend the consequences of their work. For determining the similarity between concepts, instances and properties the similarity measure is calculated in this work with linguistic, structural and semantic approaches:

- The linguistic approach exploits text-based properties of the ontologies, such as name and description. With the method EditDistance [9] string similarity is computed from the number of edit operations (insertions, deletions, substitutions of single characters) necessary to transform one string into another one. As an alternative the method N-Gram [10] can be used. Here, strings are compared according to their set of n-grams, i.e., sequences of n characters.
- The structural approach exploits relationships between concepts that appear together in a structure. Usually, concepts and their relations are represented in a graph so that dierent kinds of structural related elements can be identied for matching. To estimate the similarity between two concepts, we can compare different kinds of their neighbor elements, such as the parents, children, or the leaves subsumed by them.
- The semantic approach estimates the similarity between concepts based on their terminological relationships, such as synonymy, hypernymy, hyponymy. This approach requires the use of auxiliary sources, such as documents or annotations, in which the semantic relationship is captured. This method takes as input two ontologies and a set of documents which are linked with the concepts. We assume that, if documents annotated with concept a (of O1) are similar to the documents annotated with concept b (of O2), then the concepts a and b are similar.

Each matcher determines a match result consisting of a similarity value between 0 (strong dissimilarity) and 1 (strong similarity) for each combination of O1 and O2 concepts. To combine the single similarities to the global similarity, we use a weighted average by assigning weights to all involved single similarities.

Similarity measures for ontological structures have been widely researched. In our Editor following important requirements have been considered:

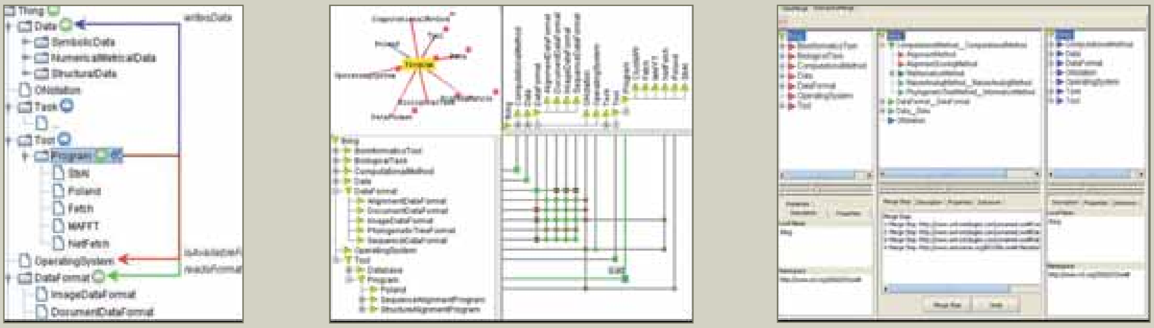


Figure3. Different views of the ontology and the result of comparison.

- Support ontology exploration and manual creation of mappings
- Provide a visual representation of the source and target ontology
- Provide a method for the user to accept/reject a suggested mapping
- Provide access to full definitions of ontology terms
- Show the context of a term when a user is inspecting a suggested mapping
- Provide interactive access to source and target ontologies
- Support interactive navigation and allow the user to accept/reject mappings
- Provide progress feedback on the overall mapping process

Only the automatic verification of the supposed mapping is not implemented by the proposed editor. The Editor proposes some mappings but does not consider possible conflicts that may occur if the concepts are merged.

References:

1) Huesken, P., Ziegler, J.: Degree-of-interest visualization for ontology exploration. In Baranauskas, M.C.C., Palanque, P.A., Abascal, J., Barbosa, S.D.J., eds.: INTERACT (1). Volume 4662 of Lecture Notes in Computer Science., Springer (2007) 116-119
2) Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Technical Report 8 (1966)
3) Damashek, M.: Gauging similarity with n-grams: Language-independent categorization of text. Science 267(5199) (February 1995) 843{848

Authors

Prof. Dr. Ziegler (ziegler@interactivesystems)
is the scientific director of the academic research group interactive systems, University Duisburg- Essen. He is particularly engaged in human-computer interaktion, engineering for interactive systems and context adaptive web applikation

Fan Bai (bai@interactivesystems.info)
is a researcher of Duisburg interactive system group. He is response for the designing and implementation of the Ontoverse collaborative ontology editor.

Zoulfa El Jerroudi (eljerroudi@interactivesystems.info)
is researcher at the interactive systems group, University Duisburg- Essen. She is working at novel visual analytics techniques which help the user in the process of interactive ontology mapping and merging

BIO2Me: An Ontology for Bioinformatics

Within the Ontoverse project, we have built an ontology for the domain of bioinformatics and computational biology: the BioInformatics Ontology for Tools and Methods (BIO2Me). This ontology currently contains about 580 classes and individuals (together representing 7.397 facts) and has thus become a considerable knowledge base for information about tools in bioinformatics. This article provides a short insight to our motivation for developing this ontology and reports our experiences from the ontology engineering process.

Prof. Dr.
Gerhard Steger

Dr. Indra Mainz

Katrin Weller, M.A.

A Short Introduction to Ontologies and Ontology Engineering

Ontologies are data constructs to store information in a structured form. For this purpose, information is decomposed into small units: the relevant abstract concepts of a domain of interest, single individual objects (mainly corresponding to real-world individuals like persons, cities or – as in our case – bioinformatics program versions), plus information about how these are interrelated. In ontologies, concepts can be collected, annotated with definitions, put into hierarchical order, interlinked with various cross-references and defined by formal rules. One may for example capture the fact that a program is a kind of tool. So the concept „Program“ is modeled as subconcept (subclass) of concept „Tool“. Furthermore, one may specify that every program has certain characteristics, e.g. that it needs a certain operating system or has a certain release date. Such types of information can be represented in form of concept properties. The overall result is a domain specific knowledge base, consisting of the core vocabulary for the domain, precise definitions and rules for interpretation.

Ontologies are typically stored in specific data formats based on ontology languages like OWL or RDF. They enable to represent ontologies in a machine-interpretable way so that certain computer programs may precisely retrieve the information stored in an ontology and also derive information that are only implicitly included in the ontology. Information captured in form of ontologies can easily be exchanged among different persons (e.g. in a working group or community) or may become the basis for automatic information systems.

Ontology engineering is the process of building an ontology. To create ontologies on the basis of formal ontology languages, so-called ontology editors are available (tools with graphical user interfaces that store information in form of ontologies). For the creation of high-value ontologies one will typically need domain experts for the topic that should be covered by the ontology as well as experts in knowledge formalization who can make use of ontology editors. Only if these different experts work together it is possible to gather relevant knowledge and to turn it into a well-structured ontology. One challenge in ontology engineering thus is to enable a community to collaboratively create, discuss and maintain ontologies.

Motivation for Building BIO2Me

The overall aim of the Ontoverse project was to provide such a space, where people can jointly collect their expert knowledge and formalize it in form of ontologies.

Thus one important aspect during the project was also to build an ontology to gather experimental data and to test the new working space. By creating our ontology, we gained a lot of experiences in collaborative ontology engineering, exposed advantages and disadvantages of available ontology editors and collected necessary features of the support of the whole ontology engineering process. Moreover, the ontology served as a well-known ontology for evaluating and testing purposes on the evolving ontology platform.

Apart from this general motivation, a more special need encouraged us to build an ontology particularly for the domain of bioinformatics. Members of our research project had been engaged in earlier research on specific tools used in bioinformatics procedures. The studies by Mainz and Wilm and colleagues both compare several alignment tools with respect to the composition of the input data. This research, amongst others, yielded that the most popular and mostly employed program of this field, ClustalW , provided not necessarily the best results for each input data set. Obviously, many researchers are using this tool although it is not the one that matches their requirements best.

This pointed out a big challenge in biology and bioinformatics in particular: There is a variety of programs, packages, databases etc. dealing with various problems like the efficient processing of experimental data, sequence analysis and structure prediction and visualization. The major problem is that these various tools can currently not be surveyed with reasonable effort. Even for experts in a specific domain of bioinformatics it is hard to decide which tool fits the given requirements best. Moreover, there are lots of programs dealing with the same problems but using miscellaneous computational, mathematical and biological approaches.

The domain of bioinformatics tools and methods is characterized by an enormous amount of available information (about the single tools) which are not at all structured and can thus hardly be accessed precisely. It is too difficult and to time-consuming to gather information about all available options and there is no platform providing the relevant information in one place. Thus, our approach is to capture these complex information about existing tools and methods for bioinformatics within an ontology: the BioInformatics Ontology for Tools and Methods (BIO2Me).

Our Approach

In BIO2Me detailed information about bioinformatics tools and methods is collected in a structured way. Bioinformatics tools are classified and described according to their characteristics and features; computational methods are examined according to their application ranges.

The resulting ontology serves as basis for an information system that enables searching for tools that meet the users' needs. Such a system may answer questions like „Which tools perform sequence analysis?“ or „Which data output do these tools provide?“. Thus, a comprehensive ontology for the domain of bioinformatics tools and methods as well as a system which uses this structured information can assist researchers in finding precise information about bioinformatics tools and methods.

The capability of the information system depends to an enormous degree on the quality of the underlying ontology as its knowledge base. A search within the system can only yield helpful results if the ontology is substantial and is constantly enlarged and updated. This cannot be done by one single person without any support of either domain experts who can contribute their knowledge about new tools and methods or the computer aided extraction of information from scientific literature.

Furthermore, the ontology has to be checked periodically for its consistency, accuracy and correctness. Thus, our solution was to establish the basic structure of BIO2Me and to put it in a place where the community of experts in this domain can contribute single pieces of information that help to keep the ontology up-to-date and where modifications and enrichments of the ontology file can be performed directly: the Ontoverse platform.

Designing BIO2Me

The practical aim to make bioinformatics tools easily accessible highly influences the actual structure of the ontology. The whole ontology conceptualization was focused on this practical applicability (sometimes dominating over strictly logical representations). Tools are categorized according to their application ranges (with bioinformatics perspective), supported biological tasks, utilized computational methods, processed data formats and information about the tools. Figure 1 shows extracts of the ontology structure.

The basic challenge in creating the ontology was to define its fundamental structure. This is where the highest quality control is needed, because it is very difficult to change fundamental structures at a later point in time. It is also the part of the work which requires the most discussion and planning. Less fundamental aspects, like adding new individuals to existing concepts, can however easily and freely be handled by a large community.

To identify the most relevant characteristics of bioinformatics tools, we used a competency questionnaire, i.e. we collected those questions that the final information system should be able to answer (like „Which tools compute sequence alignments and which data output formats do they provide?“). From these questions, we could extract many of the most important concepts for the domain of interest. Based on these initial concepts, the fundamental structure of the ontology was created by a small research team and approved by experts from the domain of bioinformatics tools.

For the highly detailed information about single bioinformatics tools, we were depending on feedback from the community who uses these tools. Reports on personal experiences thus became highly valuable knowledge resources. During the

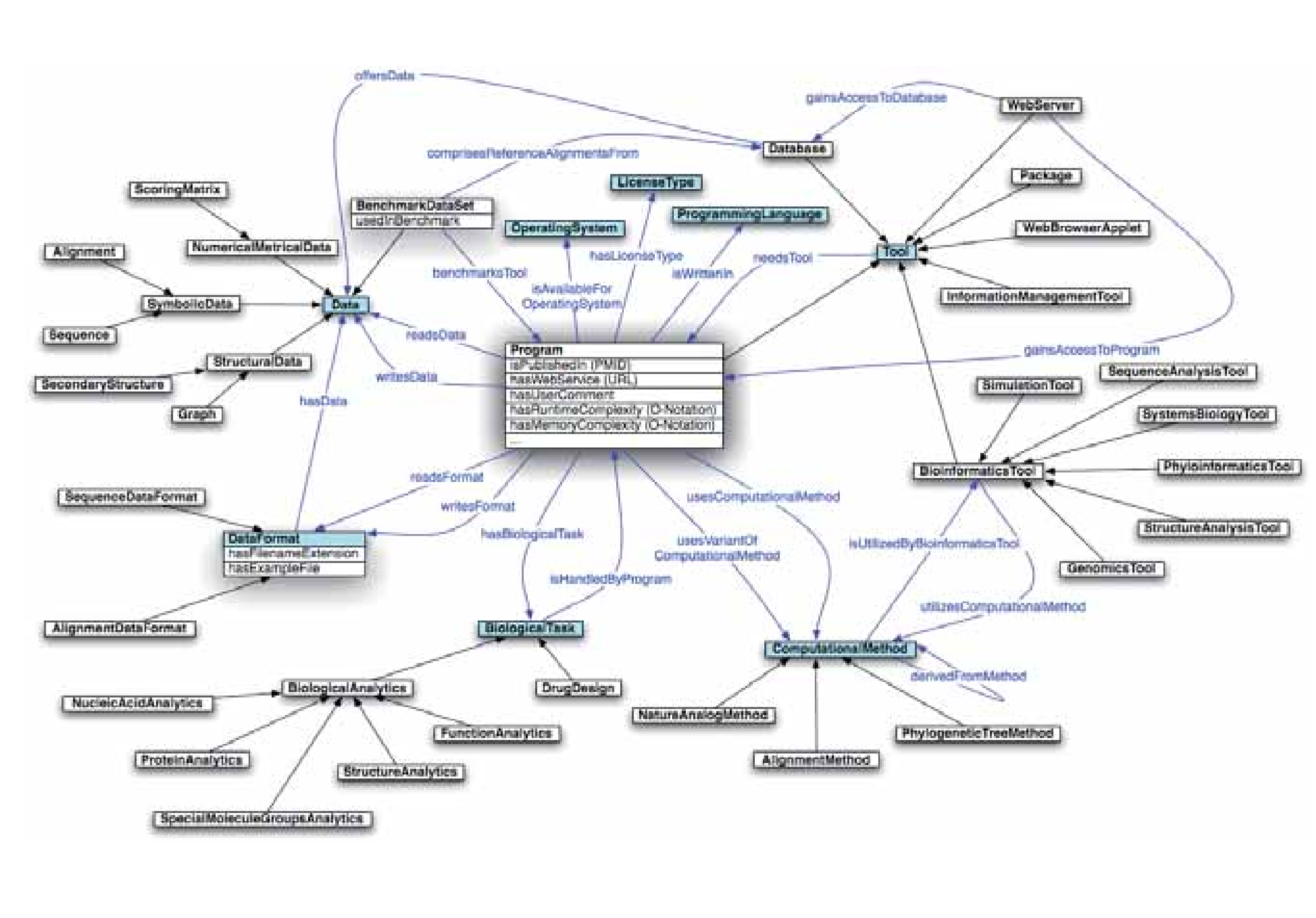


Figure 1: Overview of BIO2Me. Black arrows illustrate subclass relationships and blue pointers object properties. Some datatype properties are inserted in the concept boxes.

ontology design process we determined deficiencies of traditional ontology editors in the support of collaborative ontology engineering. Within the Ontoverse platform, we consequently paid much attention to communication channels and possibilities to add preliminary unstructured pieces of information. To exemplify the characterization of a program Fig. 2 demonstrates the classification of StrAl version 0.5.4 in the ontology network.

Application

There are many motivations for a scientist to use the BIO2Me ontology. A BIO2Me search can be helpful for a fast familiarization with a new research task in the field of bioinformatics and will help newcomers in the field to find relevant references quickly. They can get information about tools and about how other scientists addressed a certain problem. On the other hand, a lot of tools developed in diploma or bachelor/master theses are currently not published although they provide good approaches which are worthwhile to pursue. An application is needed that allows to simply add such unpublished work to the ontology and thus make unpublished methods accessible to the scientific community. Furthermore, experimental biologists can use BIO2Me to find an adequate tool for their data analyses or for the planning phase of experiments. Even for experienced bioinformaticians it is useful to get a review of available tools and to have a quick reference to differences between versions and tools, to publications and additional features. The information about input and output formats of a tool facilitates the pipeline of tools.

Lessons Learned from the Construction of BIO2Me

During the construction of BIO2Me we experienced some of the major challenges in collaborative ontology engineering – ranging from general design problems in ontology formalization to particular requirements for tools supporting the collaborative ontology engineering process.

First of all, the domain of BIO2Me eminently points out the benefits of collaborative ontology engineering. To represent bioinformatics tools with their applications, the whole bioinformatics research field and biology itself have to be displayed adequately in a structured way. Different fields of expert knowledge are needed to characterize different functions and application areas of bioinformatics tools. This is far too

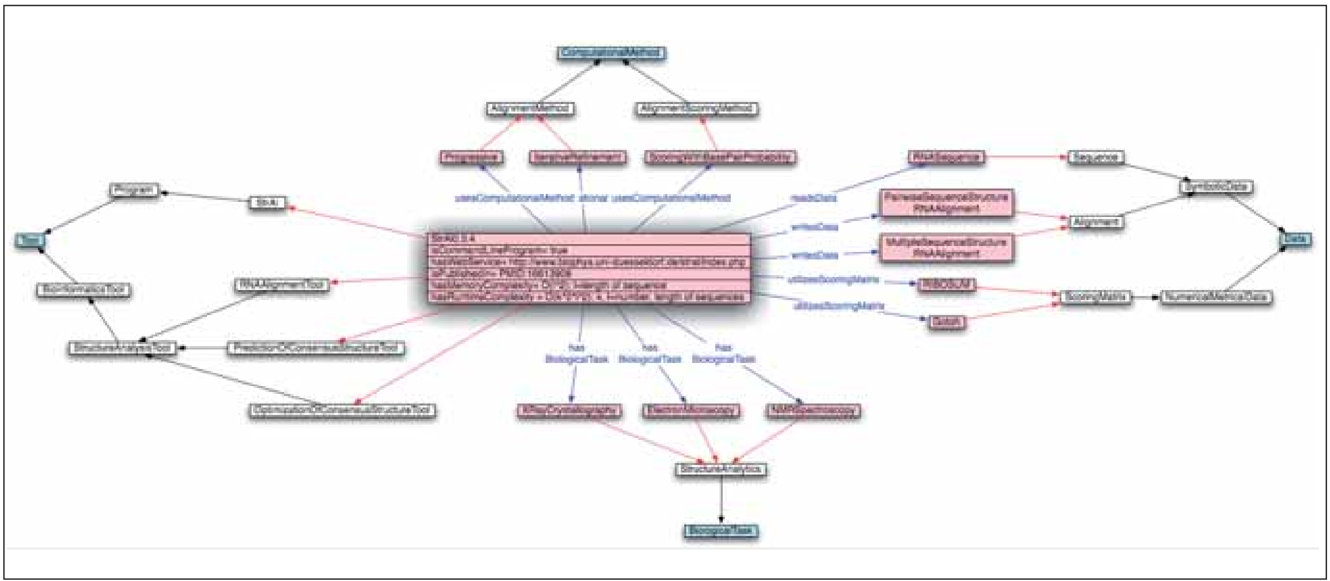


Figure 2: The schema visualizes a cutout of BIO2Me that exemplifies the filing of a program on the basis of StrAl (version 0.5.4). Red arrows indicate "instance of" relationships and pink boxes are individuals.

be handled by single persons and requires the involvement of domain experts and/or a techniques for the semi-automated extraction of information.

Yet, the incorporation of user communities into the information collection and ontology engineering process is not easy. Lots of motivation is needed to encourage users to contribute their own knowledge because it still costs some time of effort to get familiar with the idea of ontologies and the Ontoverse platform.

As we realized the problems with recruiting domain experts who are willing to contribute, we have also developed an alternative way of collecting relevant background knowledge for extending the ontology: semi-automatic information extraction from scientific publications and examination of keywords added to publications within the Ontoverse platform.

Outlook: The Future of BIO2Me

So far, we have established the core structure of BIO2Me and have filled it with 354 instances. Thus, BIO2Me comprises detailed information about more than 100 bioinformatics tools and methods. In long term, the major challenge with BIO2Me will be to keep it up-to-date. It will be necessary to keep track of new developments in bioinformatics, e. g. as new tools or new versions of existing tools may be published.

Finally, the simultaneously developed information system which is based on the ontology will have to be published and made accessible for practical usage. Anyone interested in contributing to BIO2Me may register for the Ontoverse Platform and join the Bioinformatics Ontology for Tools and Methods project!

References

1) W3C specification: http://www.w3.org/2004/OWL/
2) W3C specification: http://www.w3.org/RDF/
3) Mainz, Indra (2006). Development of a prototype ontology for bioinformatics tools. (in german). Bachelor thesis, Heinrich-Heine-Universität Düsseldorf.
4) Wilm, Andreas, Mainz, Indra & Steger, Gerhard (2006). An enhanced rna alignment benchmark for sequence alignment programs. Algorithms Mol Biol, 1, 19.
5) Thompson, J.D., Higgins, D.G. & Gibson, T.J. (1994). CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucl. Acids Res., 22, 4673-4680.

Authors

Prof. Dr. Gerhard Steger (steger@biophys.uni-duesseldorf.de)
is head of the Biocomputing and RNA Research Group of the Institute of Biophysics at the Heinrich-Heine-University Düsseldorf. His research concerns bioinformatics as well as experimental methods in RNA structure and function determination.

Dr. Indra Mainz, B. Sc. (mainzi@biophys.uni-duesseldorf.de)
is responsible for the design of the a prototype ontology in Ontoverse and works in the Biocomputing Group of the Institute of Biophysics, Heinrich-Heine-University Düsseldorf. Her research interests lay in ontology engineering techniques and bioinformatics knowledge management.

Katrin Weller, M.A. (weller@uni-duesseldorf.de)
is researcher and lecturer at the Department of Information Science, Heinrich-Heine-University Düsseldorf. Her main research topics are knowledge organization systems, collaborative ontology engineering and the Social Semantic Web.

Information Retrieval and Information Extraction in Ontoverse

The rapidly growing amount of information in electronic form in all fields of human activity makes it possible and necessary to use computers to filter the relevant information for us. Information Retrieval (IR) and Information Extraction (IE) are two slightly different approaches to this challenge. IR is about documents - it automatically filters for us the documents that are relevant to our query, while IE is about content – it takes unstructured natural language texts as input and outputs structured or semi-structured, fixed-format, unambiguous representations that can be used, e.g., to populate a database or an ontology.

Our Information Retrieval and Information Extraction module has two versions – one that is integrated in the Ontoverse platform for cooperative knowledge management in the life science network, and a standalone version. All figures in this document illustrate the standalone-version GUI since it presents the IR and IE functions in a more compact way.

Information Retrieval Functions

IR functions include simple keyword search and ontology-supported expanded query, document clustering based on tf-idf, indexing and annotation, and visualization tools, such as keyword-snippet extraction and color output in browser of annotated articles. The keyword-search functions deliver a list of file-names with the number of occurrences of the keyword, sorted in descending numerical order (i.e., the file that contains the highest number of occurrences appears on the top of the list). One can choose an ontology-supported expanded query that will include not only the occurrences of the targeted keyword but also all its hyponyms that appear in the Bio2Me Ontology. The keyword-search functions deliver absolute frequencies.

The tf-idf function operates with relative (weighted) frequencies and is better suited for document clustering. The tf–idf (term frequency – inverse document frequency) weight is a statistical measure that shows how important a term is for a document in a corpus. The 'term frequency' relates to the number of occurrences of the term in the document, while the 'inverse document frequency' is a relation between the number of all documents in the corpus and the number of the documents that contain the term. Tf–idf weighting is often used by search engines to determine a document's relevance to the user query.

Figure 1 illustrates the document-clustering function. In the 'Results' window one can see the scores, the publication-database IDs and the titles of the relevant documents. By clicking a line one can see the full title and the abstract of the relevant document displayed in the text fields to the right. Machinese Metadata from Connexor (www.connexor.com) is a tool for term extraction and named entities (NEs) recognition. It supports custom lexicons and can be used for indexing and

Prof. Dr. James Kilbury

Dr. Katina Bontcheva

Christof Rumpf

Simone Kirstein

Nico Kimm

Matthäus Zloch

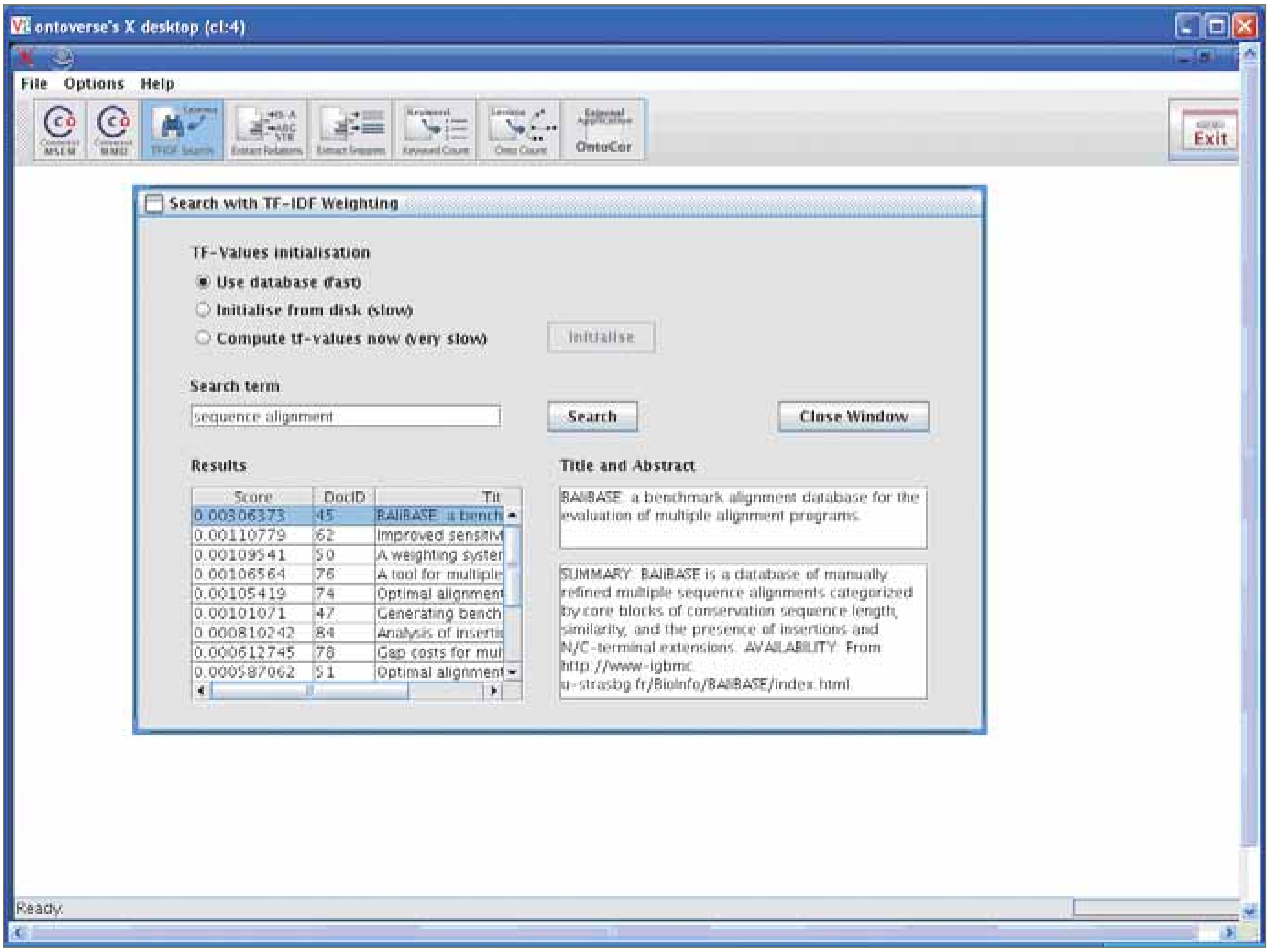


Figure 1 Document clustering with tf-idf

annotation. The annotation function uses a Machinese Metadata custom lexicon that mirrors the Bio2Me ontology.

The visualization tools are designed to help the domain experts and allow them to see the context of the extracted NEs and relations. Figure 2 (see next page) illustrates the snippet-extraction function.

Information Extraction Functions

The main IE functions are relation extraction and co-reference resolution.

Our purpose in this project was to extract instances of bioinformatics tools and methods to maintain the prototype ontology. The ontology was constructed by domain experts who defined its structure, the main classes and the relevant relations between the concepts. Thus, we had the task of extracting instances (e.g., 'ClustalW 1.83') of the predefined concepts (e.g., 'sequence alignment tool') and predefined relations between them (e.g., 'ClustalW 1.83' 'utilizesScoreMatrix' 'BLOSUM').

We concentrated on relation extraction. Mere extraction of named entities or terms is not an option – to maintain the ontology we need to extract taxonomic and other predefined relations. However, NE extraction appears to be a by-product of relation extraction.

Our main goal was to extract taxonomic relations, precise or underspecified. We achieve this by, e.g., applying XQuery scripts to the output of Machinese Semantics to extract the relevant arguments of the verb 'be' and appositions. The parser itself performs sentence splitting, tokenization, lemmatization, chunking, semantic roles and other dependency relations assignment, determines the polarity of the sentence, and provides other information that is not directly relevant for this task. With appropriate ontology-supported passage retrieval, the precision is very high while the recall remains relatively low. The method discussed so far can lead to extraction not only of instances, but also of concepts that are not yet in the ontology. However, we always present the extracted relations to the domain experts to decide whether they are precise or underspecified.

Similarly, our main method of extracting other relations is to take the relevant arguments and/or adjuncts of relevant verbs from main and subordinated clauses. For example, from the sen-

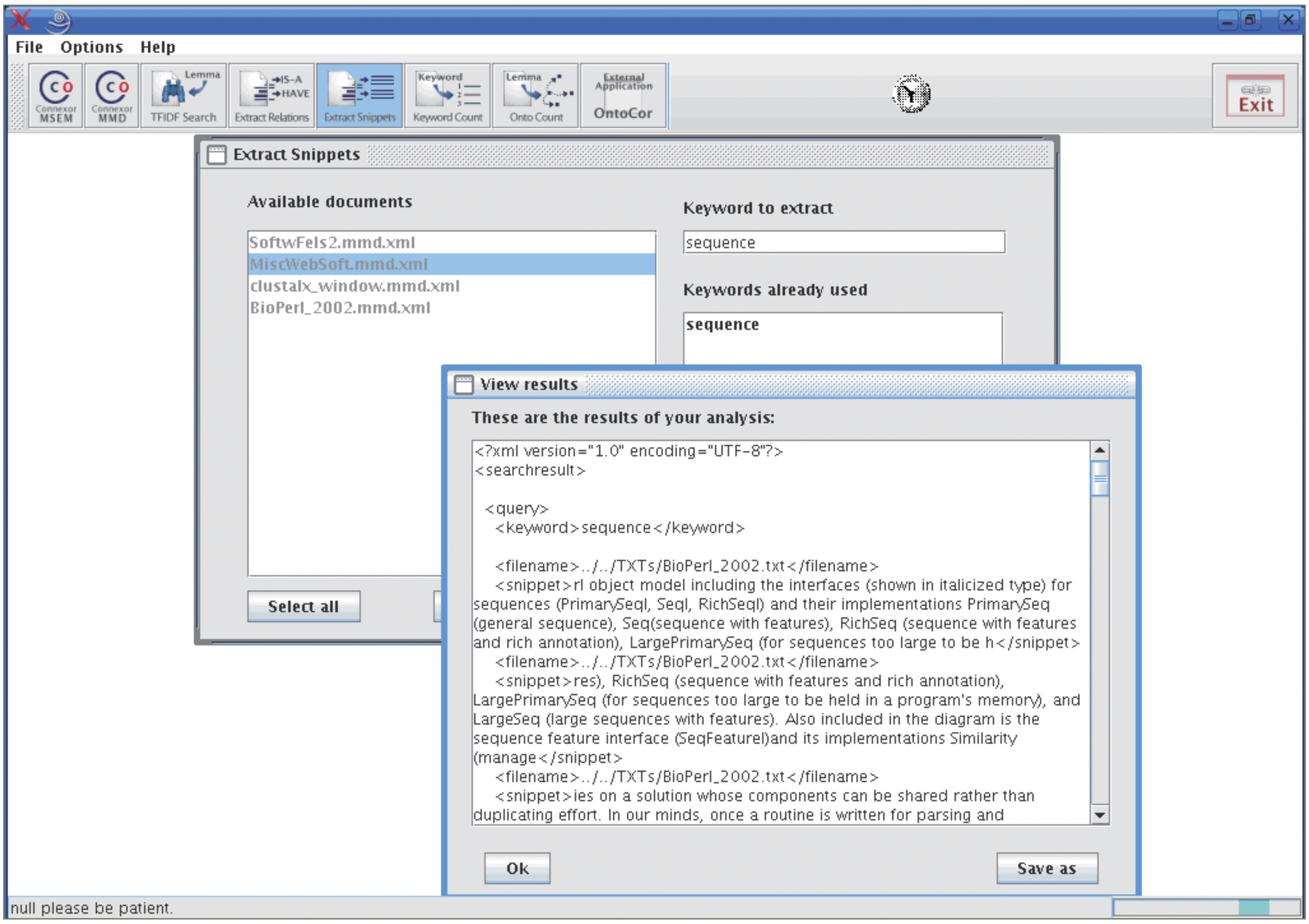


Figure 2. Snippet extraction

tence 'In a number of recent research projects DIALIGN has been used to align syntenic genomic sequences.' will be extracted the relation 'hasBioinformaticalTask' that links the program 'Dialign' with the task 'alignment of syntenic genomic sequences'.

A major task for us was to improve the recall. We use complementary techniques, e.g., pattern-matching approaches such as noun-coordination information combined with latent semantic analysis to find co-hyponyms. We also developed a Java-based application for co-reference resolution OntoCor (Figure 4 below) on the basis of the operating principles of Mitkov's Anaphora Resolution System.

All NPs in the range of three sentences before a pronoun are collected as potential co-reference candidates. We apply several indicators. The indicators referential distance, lexical reiteration, and first NPs were implemented without any significant changes compared to those originally used by Mitkov. The original indicators term preference and indicating verbs were modified with respect to the most frequent occurrences of terms and verbs in a domain specific corpus. Collocation match was enhanced by manually created semantic classes of verbs, so that not only candidates preceding the same verb lemma as the pronoun are boosted, but also verbs of the same semantic class. Every indicator gets a special score and the NP with the highest composite score is proposed as the co-referent. In the case of nominal co-reference resolution we do not need all indicators, because the main task is to match the heads first by comparing the lemma strings, and then by consulting the ontology if the concepts fit.

Third party technology and License Agreement limitations

In our IE and IR systems, we employ third party technology. We use the open source XSLT and XQuery processing product Saxon-B of Saxonica (www.saxonica.com). The linguistic analyzer Machinese Semantics and the tool for extraction of terms and named entities Machinese Metadata from Connexor are commercial. The License Agreement with Connexor imposes certain restrictions on the access to the output of Machinese technology. Thus, only project administrators have the permissions to perform tasks that involve analysis with Connexor technology and to access the direct output of Connexor's analyzers. The results of tasks that postprocess the output of Machinese Semantics and Machinese Metadata are freely available to all users of the Ontoverse platform.

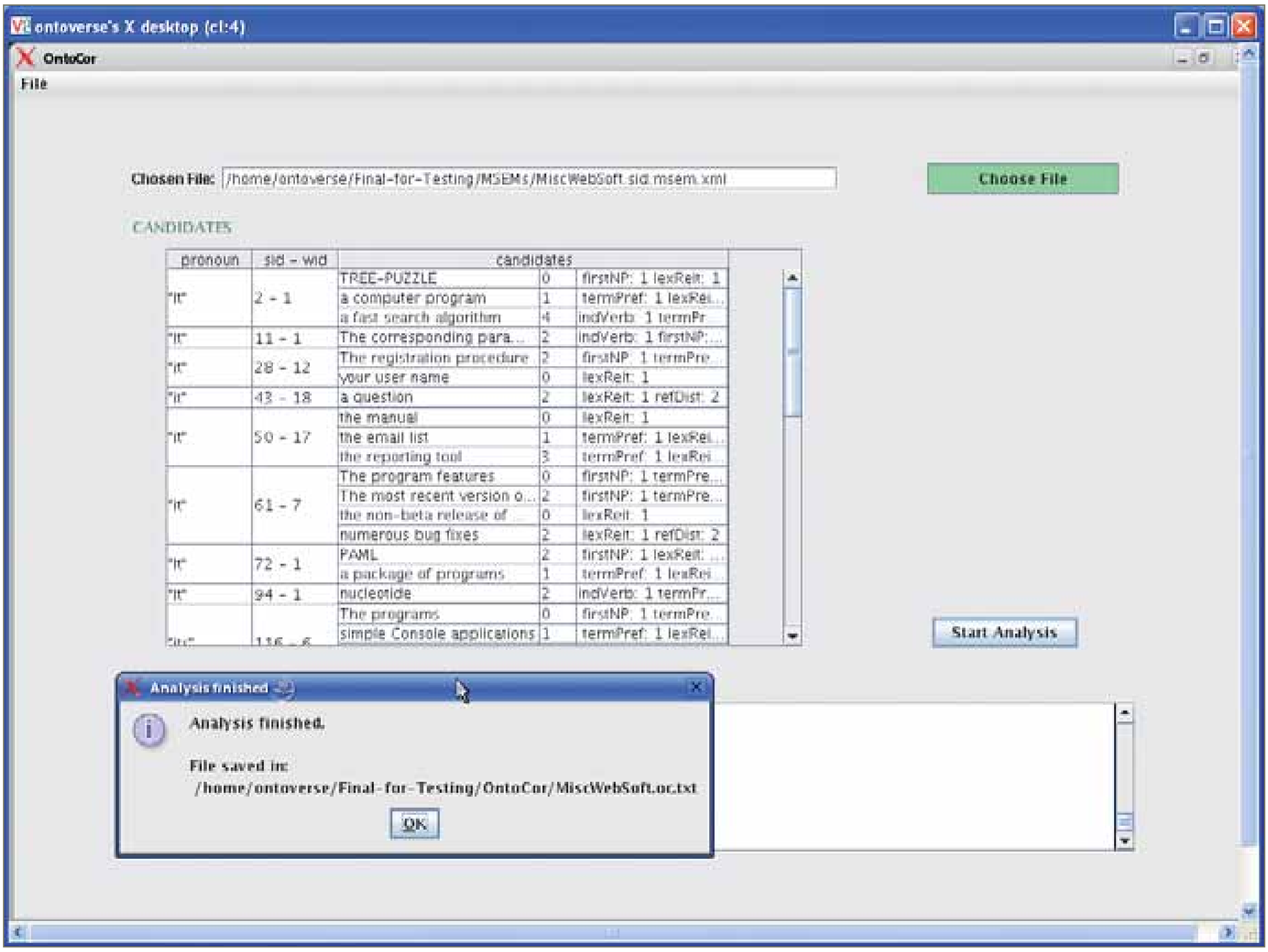


Figure 3 The Co-reference resolution module OntoCor.

Authors

Prof. Dr. James Kilbury (kilbury@ling.uni-duesseldorf.de)
is the head of the Ontoverse Computational Linguistics (CL) group and has a full professorship for CL at the Institute for Language and Information, Heinrich-Heine-University Düsseldorf.

Dr. Katina Bontcheva (bontcheva@phil-fak.uni-duesseldorf.de)
is the principal researcher of the Ontoverse Computational Linguistics group and developed the Information Extraction and Retrieval components, the interfaces between different natural language processing components, and the architecture of the XQuery-based extraction tools as well as the ontology-based document clustering tools.

Christof Rumpf (rumpf@uni-duesseldorf.de)
is a senior researcher of the Düsseldorf Computational Linguistics group and designed the integration of the Information Extraction (IE) components in Ontoverse: IE based on Ontologies and Ontology maintainance supported by IE.

Simone Kirstein (kirstein@phil-fak.uni-duesseldorf.de)
is a researcher of the Düsseldorf Computational Linguistics group and developed the anaphora resolution components for the Ontoverse Information Extraction Engine.

Nico Kimm (kimm@phil-fak.uni-duesseldorf.de)
is a student research assistant and is responsible for the preprocessing and normalization of corpora. He took part in the development of the anaphora resolution tool. His subjects are Computational Linguistics and Information Science.

Matthäus Zloch (Matthaeus.Zloch@uni-duesseldorf.de)
is a student research assistant and designed and implemented the graphical user interfaces for the Information Extraction and Document Clustering modules. He took part in the development of the anaphora resolution tool. He is studying Computer Science.

Information-Security provides Trust worthiness for the collaborative development of Ontologies

Dipl. Ing.
Rudolf Schöngarth

For promoting and supporting the acceptance of the Ontoverse ontology development system as an instrument for efficient and innovative collaborative development it seems to be necessary to provide trustworthiness and reliability. The users need and are also asking for secure mechanisms to protect their author- and creatorship as well as their intellectual property.

Embedding these security goals into the Ontoverse platform will build up confidence and increase the willingness of the individual to support the whole community by putting his knowledge into the system. In conclusion the establishment of models for trustworthy collaborative development will be supported by technical incentive system which will deliver mechanisms for authenticity and origin of the knowledge base. In addition to that, the integration of procedures proofing the point of time when knowledge has been brought into the ontology will motivate the individual to place his knowledge at an early stage. This will consequently accelerate the growth of the whole knowledge base.

An important precondition for the functional capability and reliability for the provision of these security services is a closed security architecture with balanced security components and security functions which rely on each other. Only such a security framework will be able to provide the needed services in their entirety.

The following figure shows an overview over the separate and co-acting parts of the security framework for the Ontoverse collaborative ontology development platform:

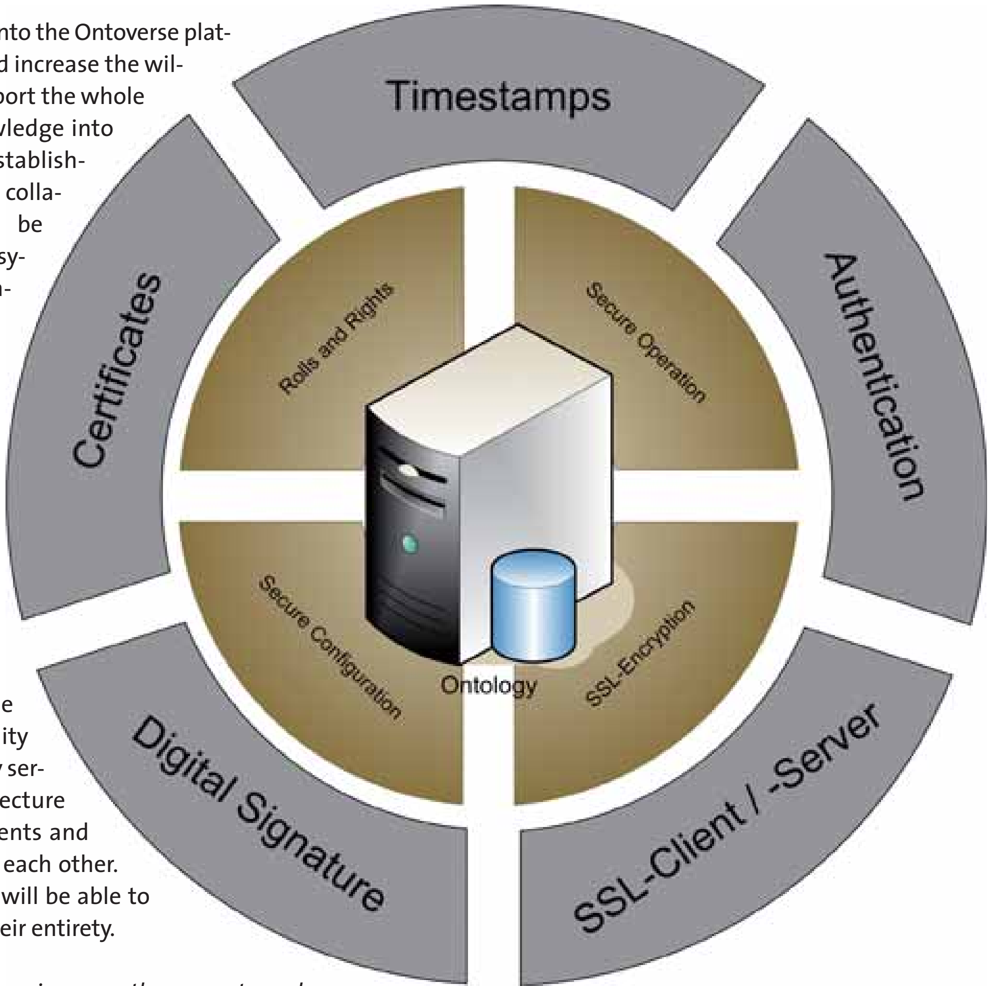


Fig 1: Overview of the information security framework

The security framework consists of two shells. The inner one provides security mechanisms for the technical implementation. The outer one realizes higher security services for the trustworthy cooperation and motivation model as well as for the inner shell. Figure 2 shows the components which are used to implement the trustworthy cooperation model.

There is first of all the Trustcenter as a trustable root for the Ontoverse Public-Key-Infrastructure, which issues digital certificates for signing, authentication and encryption purposes. The timestamp service offers secure and authentic timestamps for the proof of creatorship and time of creation. The following chapters will describe these central components in a more detailed way.

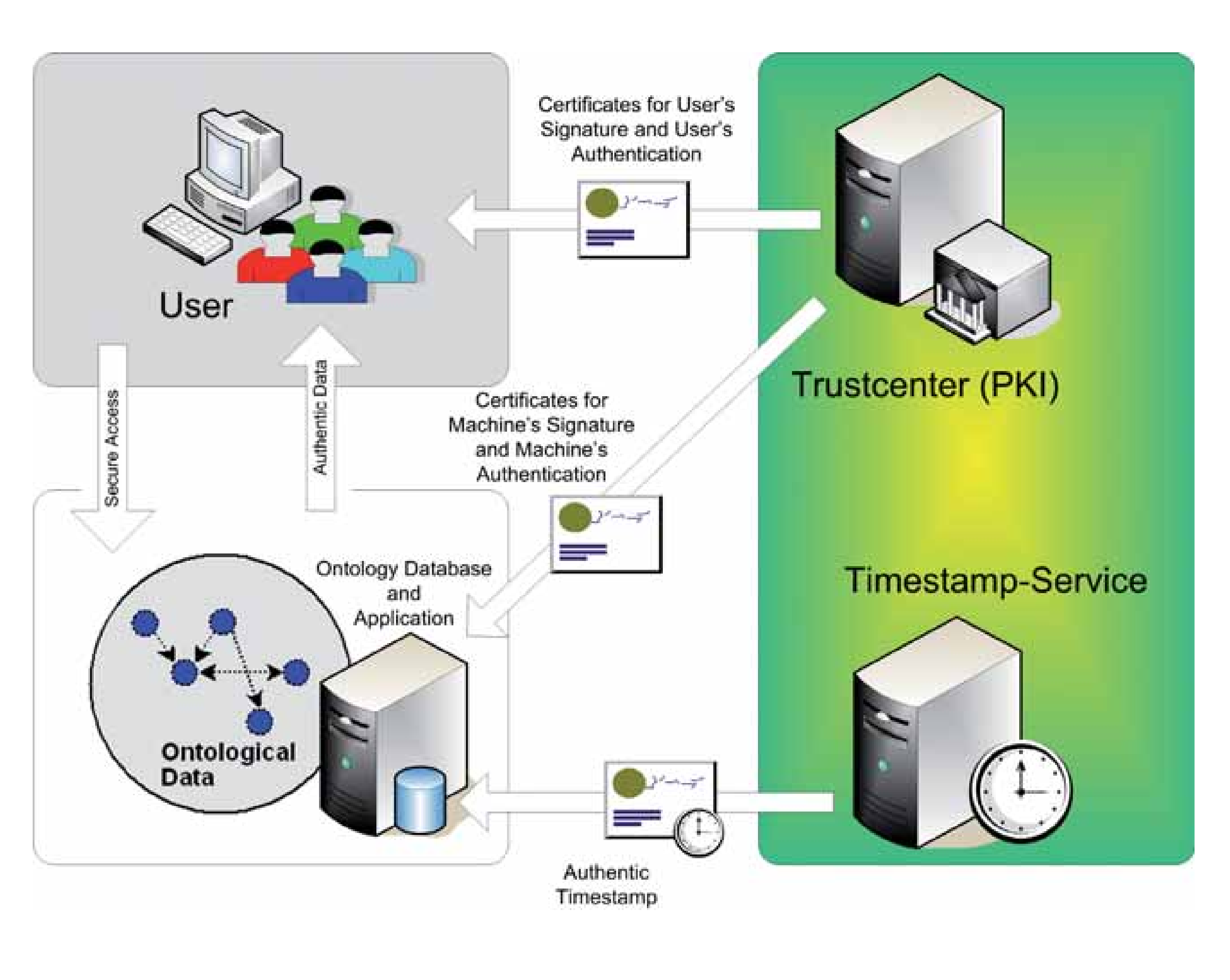


Fig 2: Components of the security framework

The Ontoverse Trustcenter Allows Proof of Electronic or Individual Identities

The security framework for the Ontoverse incentive and motivation model is based upon asymmetric cryptography. This means, that pairs of keys are used, which consist of a private and public part. The private part represents the secret which is used for signing, authentication and encryption purposes. The other part is for the public to verify identities and origin. For making it possible to give identity and origin proofs it is necessary to securely bind identities with key pairs, so there is a definite and unambiguous assignment between the key and owner of the key. The owner of a key i.e. a pair of keys can be an individual as well as a technical component like a server. The secure mapping between owner and key is being done by the Ontoverse trustcenter, which issues certificates for the members of the Ontoverse community as well as for the technical components of the ontology development system on base of a secure and controlled environment.

The trustcenter represents a trustable root authority for providing certificates similar to electronic identity cards. By using these certificates and by integrating them into the security framework, applications and functions for assuring authenticity of persons, machines and data for the purpose of Identity-, Origin- and Authenticity-Proof can be realized.

The Ontoverse Time-Stamp-Service Provides Proof of Authorship and Creation Time

The other central component of the ontoverse security frameworks is represented by the Ontoverse Timestamp Service (TSS). The TSS consists of a trustable and authentic clock plus a secure signing component which is based upon the Ontoverse trustcenter. By owning a key pair as well as a certificate the TSS is attached to the trustable security architecture and becomes part of the trustworthy framework of the incentive and motivation system.

The TSS plays an important role within this system as it binds the Ontological database to trustable and authentic time information. By timestamping the collaborative knowledge base and the users contributions for the community it allows to proof origin, time of origin and creatorship of information. figure 3 shows how the timestamping process is being realized:

The Ontology-Data as well as the Modelling-Log-Data will be sent to the TSS continuously in the way of a hash-value which represents a fingerprint of the actual status of the collaborative work. This fingerprint is attached with actual time information and signed by the TSS. The result is being sent back to the ontology and stored within the database. By doing this, it can be assured that some information respectively status has been existed at this particular point in time.

By using the Ontoverse Timestamp Service in association with the Ontoverse Trustcenter author- and originatorship of knowledge can be proofed. By freezing the actual status of the knowledge base and combining this information with authentic time information an incentive system driven by a competition situation will be implemented.

Conclusion

By working together, the concerted trustable components of the Ontoverse security framework provide trustworthiness, authenticity and security for the whole Ontology Lifecycle and the collaborative knowledge base and therefore reliability for the community and their users. This means added value for the platform itself as well as for the community and their knowledge. In addition, the Trustcenter- and Timestamp-Services can also be used to implement further security features like transport- and content-encryption, User-Authentication by End-User-Certificates or Certificate-based access rolls- and rights for groups and functions within the Ontology. By using individual and machine based digital signatures, the integrity and authenticity of imported and exported information can be verified to safe the added knowledge of the ontology and its value as well as to protect the copyrights and ownerships of the community.

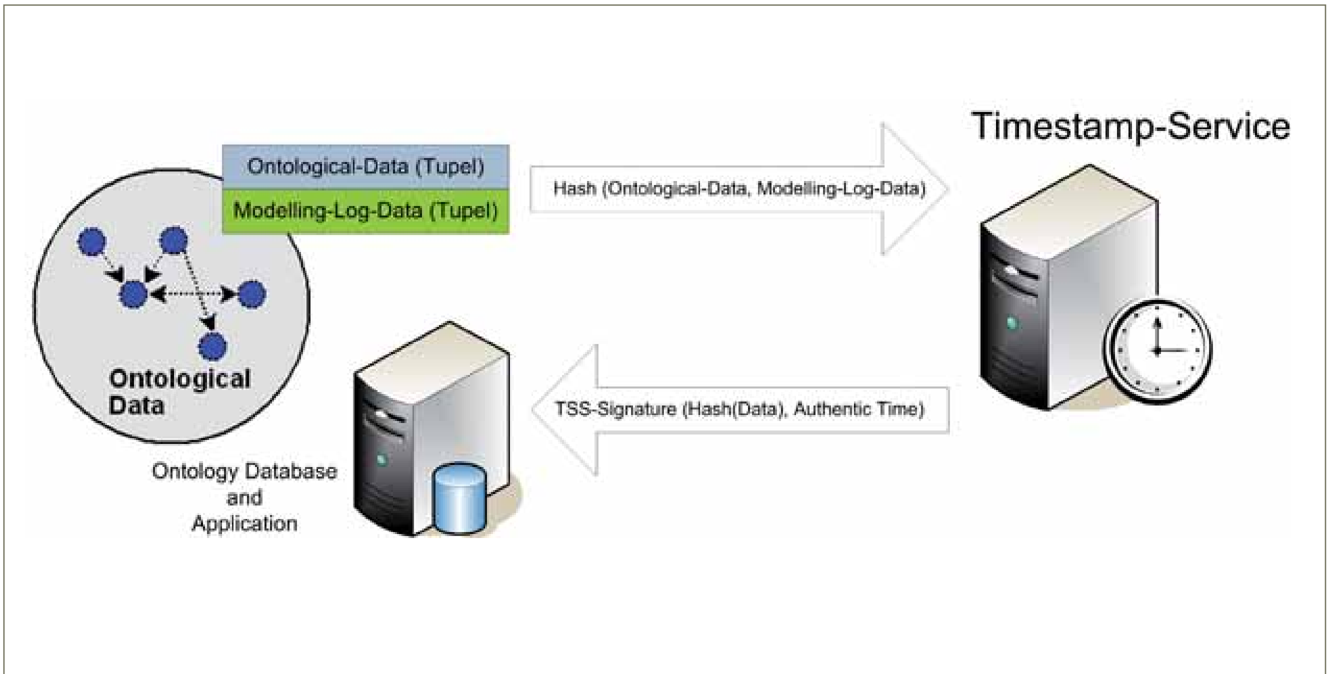


Fig 3: Timestamping

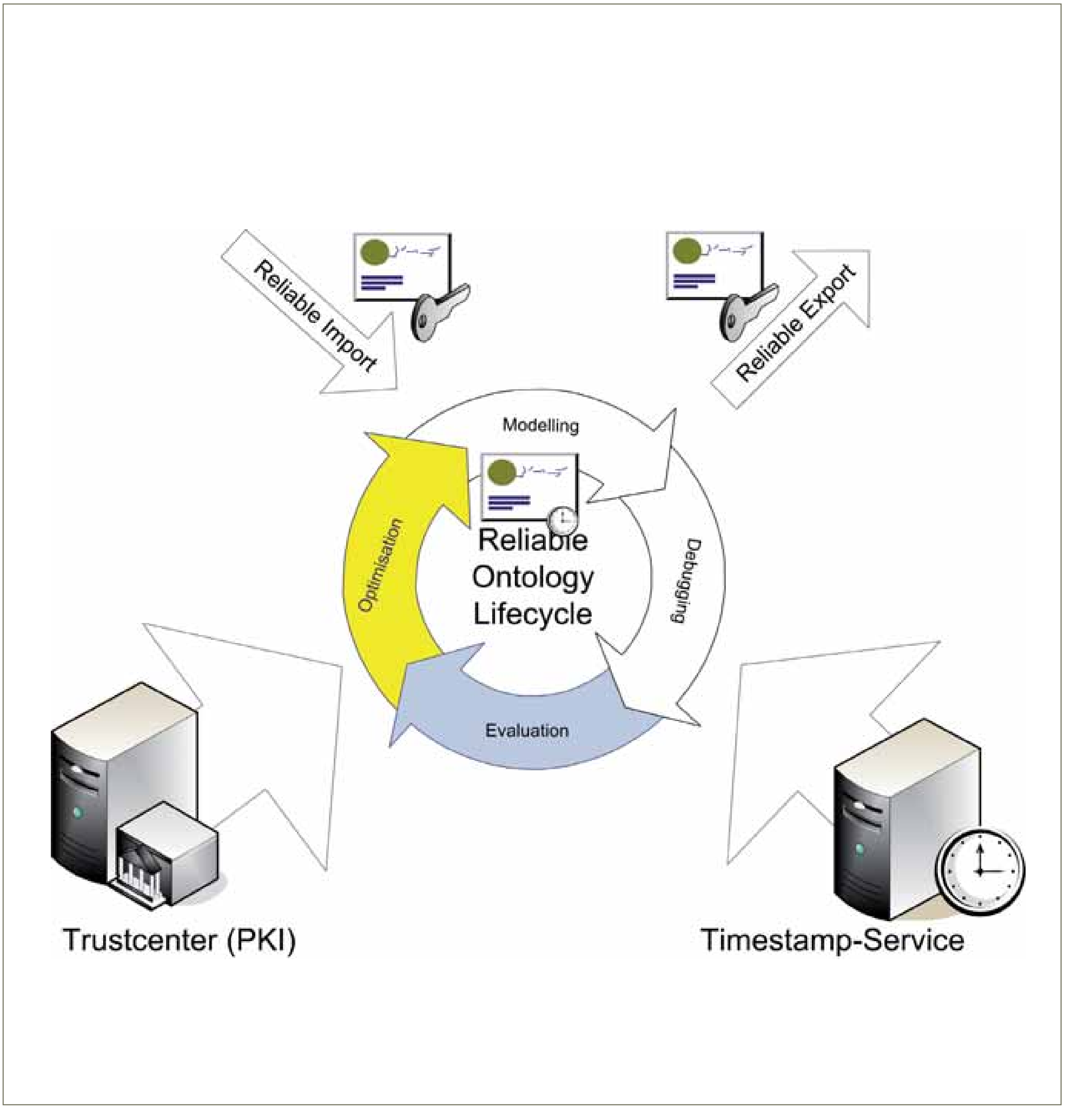


Fig 4: Reliable Ontology Lifecycle

Author

Dipl. Ing. Rudolf Schöngarth (rudolf.schoengarth@secunet.com)
is IT-Security Consultant at secunet Security Networks AG. His focus is on the design of information security management systems for governmental issues and he is licensed auditor for ISO 27001 audits in compliance with IT-Grundschutz.

Collaborative ontology engineering needs a specific incentive system

Dipl.-Wirt.-Ing. Markus Schroll

Dr. Joachim Hafkesbrink

Everybody is talking about Semantic Web and ontologies are seen as the enabling technologies of the future web. But: what about the ontology builders and users? Are the former able and motivated to design new ontologies and will they take the collaborative way, for what reason? And what about the latter, the user of ontologies? What is the motivation of people to adopt an ontology? Beside all the technical issues, these socio-economic questions have to be answered, if the Semantic Web with ontologies should become a success story.

Collaborative ontology engineering as a commons based peer production

Collaborative ontology engineering has a common ground with open source and Web 2.0 activities and can be characterized as a commons based peer production (Benkler, 2002, 2006). The following three premises for a successful implementation have been identified (Piller et al. 2007) and were integrated in the Ontoverse platform:

Principle of granularity and simple interactive work division: The total work of building a complex ontology can be divided in small tasks via Ontoverse. This is made possible via synchronous and asynchronous collaboration spaces within the platform, supported by typical Web 2.0 features (wiki, blogs etc.; for more details see the other articles in this brochure).

Attraction of a sufficient number of motivated participants: We are aware of the fact, that this is the most crucial and most difficult point, and will discuss this more in-depth in this artcle.

Openness and a non-proprietary protection of the knowledge created: The Ontoverse platform is open for everybody; one can generally start an ontology project and join existing projects on Ontoverse. Nevertheless the project team has developed different mechanisms for the access control and intellectual property rights (IPR), which allow to adjust access rights for certain contexts: There is a project administrator for each ontology project who may decides by himself or in discussion with the existing project participants about the acceptance and roles of new members. The reason for such a "controlled openness" is that Ontoverse would like to support even research groups which work - for what reason ever - on not publicly available ontologies. That means each ontology project can decide by itself who should provide and share the common knowledge. The IPR issue is solved by the trustcenter allowing the proof of electronic and individual entities in combination with the time stamp service providing the proof of authorship and creation time.

Collaborative ontology building: The crucial point of motivation

Best practices like the well-known Wikipedia show that collaborative work in the Web is possible with excellent results. Ontologies as knowledge representations seem to be a bit similar, but they are definitely not. Siorpeas, Hepp (2007) call it "The Motivational Divide: Web 2.0 is fun, ontology engineering is not." The construction of an ontology needs the collaboration of real experts (ontology designers as well as domain experts) for several reasons: The ontology designers act as modeling experts or knowledge engineers for the formal knowledge representation. The domain experts deliver the input, which will normally be done in a collaborative way to capture as much domain knowledge as necessary. Both actors collaborate in the sense that the ontology designer supports the domain expert to articulate his knowledge in the ontology.

The successful building of an ontology depends on the motivation or the cost-benefit ratio of the participating people. Participants in Web 2.0 projects usually get immediate benefits like information they have searched for, new contacts, or at least some kind of virtual fame or reputation. Some of them have a clear task to be solved; others have a common interest or hobby in specific areas. Existing research on the motivation of open source programmers shows that intrinsic motives often dominate extrinsic motives (like money, exploitation of the results), especially if participating in such peer groups leads to senses of competence, fun, exploration and creativity (e.g. Deci et al. 1999), is embedded in a common understanding of social norms and led by a shared vision. Typical Web 2.0 incentives are altruism, belonging to a community, social components, reciprocity, competition, autonomy, self presentation, reputation, attracting attention (see Kuznetsov, 2006; Marlow et al., 2006). Sometimes the collaborative ontology engineering can be different. People who are investing time in the construction of an ontology are not necessarily the ones, who will finally benefit from it. E.g. think of the ontology designer, who is in charge of the formal knowledge engineering, but has no possibilities to exploit the ontology. In these cases the time effort spent exceeds the expected personal benefit.

Authors:

Dipl.-Wirt.-Ing. Markus Schroll (ms@innowise.eu) **and Dr. Joachim Hafkesbrink** (jh@innowise.eu) are owners and managing directors of the **research and consulting company innowise GmbH located in Duisburg**, Germany (www.innowise.eu). Their research focus in the Ontoverse project was on the incentive systems and user interaction. They are experts in innovation systems and system innovations doing research for different institutions on the Federal and European level.

References:

1) Benkler, Y. (2002), 'Coase's penguin, or, Linux and The nature of the Firm. In: Yale Law Journal 112(3), S. 369-446.
2) Benkler, Y. (2006): The wealth of networks, Yale University Press, New Haven.
3) Deci, E. L.; Koestner, R.; Ryan, R. M. (1999): Meta-analytic review of experiments: Examining the effects of extrinsic rewards on intrinsic motivation. In: Psychological Bulletin 125(6), S. 627-668.
4) Kuznetsov, S. (2006): Motivations of Contributors to Wikipedia. In: ACM SIGCAS Computers and Society, 2006.
5) Marlow, C. et al. (2006): Position Paper, Tagging, Taxonomy, Flickr, Article, ToRead. World Wide Web Conference. Edinburgh, Scotland.
6) Piller, F.; Reichwald, R.; Ihl, C. (2007): Interaktive Wertschöpfung - Produktion nach Open-Source-Prinzipien. In Lutterbeck, B; Bärwolff, M.; Gehring, R. A. (Hrsg.): Open Source Jahrbuch 2007.
7) Siorpaes, K.; Hepp, M. (2007): OntoGame: Towards Overcoming the Incentive Bottleneck in Ontology Building, in Proceedings of the 3rd International IFIP Workshop On Semantic Web & Web Semantics (SWWS '07).
8) Siorpaes, K.; Hepp, M. (2008): Games with a Purpose for the Semantic Web . IEEE Intelligent Systems, Vol. 23, No. 3, pp. 50-60.
9) von Ahn, L. (2006): Games with a Purpose. In: Computer, vol. 29, no. 6, 2006, pp. 92–94.

Motivation	Ontoverse incentive functions
belonging to a community (a common goal, a shared vision)	ontology projects can be set up individually for specific purpose with a specific community each project has its own Wiki including an ontology requirement specification document as a guideline for the whole engineering process e-mail newsletter and news/events blog for the Ontoverse community private messaging between the users
social components	discussion forum system blogging engine
reciprocity	access to each ontology project is by default open for everybody but can be controlled by the project administrtor if needed use of all intra-ontology knowledge is feasible use of all inter-ontology knowledge is feasible and realizable via ontology merging function or participating in different ontology projects
competition	different views on the ontology can be edited and discussed in the ontology project community provision of private and shared ontology editing spaces concurrent versioning of ontologies is possible
autonomy, freedom	users decide by themselves about their contributions decisions can be managed via the project specific Wiki and a chat room
self presentation	user accounts and profiles (incl. photos and publications), showing the specific expertise and membership in ontology projects
reputation	system of different roles (project administrator, ontology designer, domain expert)
attracting attention	tagging of projects, user's expertise and publications for an easy search and browse
trust	Trustcenter-service allows proof of electronic and individual identities
intellectual property rights	Timestamp-service provides proof of authorship and creation time

Marketing the Ontoverse project

Intensifying transfer and diffusion of scientific results by addressing heterogenous target groups using a community integrating approach

Jürgen Mainz
CEO

Horst Hallmann
Project Manager

While marketing in business enterprises represents a business function in the sense of an orientation of business decisions on the market – including sales policy issues which are solved by using the product, price, communication and distribution policy – marketing in the field of science serves to convey the achieved research results to potential interested parties. Thereby securing sustainability of project results and generally contributing to a „public understanding of science". Thus, the communication policy as part of the marketing mix constitutes a central role in the marketing of science. As a primary performance the project, its unique characteristic and the scientific quality of achieved research results are uniformly communicated.

As one of the first operations a scientific project design – in the sense of a corporate design – is developed. Using the developed advertising constants (logo, slogan, colors, typography and layout) a high recognition value in communicating with the target groups is sought and achieved. For the broadest possible network effects, the target group is addressed in different media and information channels. In workshops, conferences and exhibitions, the research is presented and discussed with colleagues. Print materials such as brochures, flyers and posters support the exchange of information. Press releases and an internet portal, as one of the major information media, complete the strategic action plan. *(see Figure „Communication strategy")*.

The communication of scientific results across the narrow limits of each special field remain a significant challenge for many innovative research projects . Scientific publications only reach a limited audience, while conventional project websites often do not generate enough attention. Unlike a traditional sender-receiver model, i.e. the provision of information on a static website, marketing activities during the Ontoverse project enhanced active exchange of information between the project partners and external stakeholders from academia and industry over a community platform. The marketing objective was to generate an early attention in identified target groups to maximize the number of interested parties who are informed in advance about the start of the Ontoverse prototype. Of great interest were particularly those ontology projects being in the planning stage or already active with their own user groups as potential early adopters. By their early public presence the Ontoverse community continuously generated attention. As a result, scientists, research institutes and companies from industry and small and medium-sized enterprises (SMEs) were heightened in their awareness for the theme of „collaborative ontology building". In this process the main focus of interest constituted the life sciences, with their knowledge-intensive work processes and issues.

Based on extensive research three primary target groups were identified. Additionally general information about ontologies and their applications had to be conveyed to potential users of ontology-based knowledge management solutions.

Target groups of the community project

1. In the initial phase publicly funded scientists, in particular from existing ontology projects in the life sciences, had to be interested for the Ontoverse platform. Members of this target group are domain experts and thereby of essential importance for the development of bio-ontologies. The primary communication objective with respect to existing ontology projects was the depiction of the specific benefits of the innovative, collaboration supporting Ontoverse approach.

2. Companies from the life sciences (not exclusively) had to be sensitized for Ontoverse and semantic technologies in order to serve as prospective value-adding resources for the marketing of the whole platform (e.g. production, disposition or certification of ontologies).

3. Companies and research institutions in the field of semantic technologies were identified as sources of relevant content for the community platform as well as being important partners for transfer and diffusion.

Benefits for users of the community platform

- Publicly funded scientists receive information about the potential benefits of ontology-based tools regarding their work. Furthermore ontologies play an important role in an increasing variety of areas in the life sciences as formal models of existing knowledge.

- Life science companies can get information about ontologies and semantic technologies. This group primarily attaches its focus on experiences with pilot applications and existing products, the search for cooperation partners and projections about the future development of technological possibilities and their implications.

- Companies and research institutions in the area of semantic technologies get the option to show their work to an interested audience as well as access to a central community platform for establishing contacts with industry and users.

Community Platform Content

The content area was divided into two areas: a publicly accessible area with general available information and an area only for registered community members. In the news section upcoming events in the field of life sciences and recent publications are noted. The topic „Ontoverse" provides information categorized in project description, project objective, project partners and presentations & events. A descriptive introduction to operation and functionalities of the Ontoverse prototype is provided under the heading „demonstration". Divided into four chapters, a webcast explains the different modes of operation.

Chapter 1 introduces basic features for social networking and community support. Chapter 2 demonstrates the organisation of the working area into projects and possibilities to plan ontologies and add informal knowledge to the ontology project. In Chapter 3 the formal ontology editor is shown. Finally, in Chapter 4 describes, how the ontologies can be enriched with new concepts suggested by an information extraction application.

The heading „Bio2Me" relates to a prototype ontology, which was developed during the Ontoverse project, (see also article on page 16). In addition, a glossary explains the technical terms and completes the general information area.

In order to encourage the exchange of information between users and developers of semantic technologies in the field of knowledge management a community area was established. After registering the user gains access to the community area. Different user rights allow the community members to upload and publish information. This area provides e.g. expert profiles, ontology profiles, research group profiles and community profiles which are all added by the user community.

Conclusion

Marketing is an approved and successful instrument of corporate management. Business approved communication and marketing strategies can be used to improve the exchange, transfer and diffusion of scientific research results. In order to maximize a project's overall societal and scientifical impact the use of professional marketing partners is highly recommended. Particularly, large projects should take this into account in their project planning to avoid asymmetries in the communication process from the very start.

Communication strategy

Authors:

Jürgen Mainz (jmainz@varion.de),
is owner and managing director of VARION GmbH – Agency for communications design located in Krefeld, Germany (www.varion.de).

Horst Hallmann (hhallmann@varion.de)
was involved as the project leader and responsible for the realization of the communication and marketing mix.

They are experts in the development of strategic marketing and advertising concepts, product launches and product presentations including crossdata/crossmedia, Multimedia/Flash commercials, market studies and software development. Furthermore both are involved in concept development, creation of commercial expert portals and science marketing.

supported by:

Bundesministerium
für Bildung
und Forschung

www.ontoverse.org

ONTOVERSE
Global Knowledge for Global Challenges

Heinrich-Heine-University Düsseldorf
Universitätsstr. 1, 40225 Düsseldorf
www.uni-duesseldorf.de

Universität Duisburg-Essen
Campus Duisburg: Forsthausweg 2, 47057 Duisburg
Campus Essen: Universitätsstraße 2, 45141 Essen
www.uni-duisburg-essen.de

innowise GmbH
research · consulting
Bürgerstr. 15, 47057 Duisburg · www.innowise.eu

secunet Security Networks AG
Kronprinzenstr. 30, 45128 Essen · www.secunet.de

Center for Integrative Bioinformatics Vienna
Max F. Perutz Laboratories, Prof. Dr. Arndt von Haeseler
Dr.-Bohr-Gasse 9, A-1030 Wien, Webpage: http://www.cibiv.at

VARION GMBH AGENTUR FÜR KOMMUNIKATIONSDESIGN
Dießemer Bruch 112i, D-47805 Krefeld · www.varion.de