The Ontology-Based Architecture of LT World, a Comprehensive Web Information System for a Science and Technology Discipline

Brigitte Jörg, Hans Uszkoreit

DFKI Language Technology Lab
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
{brigitte.joerg, uszkoreit}@dfki.de

Abstract. The Web-based information system LT World is a comprehensive knowledge portal for the large and multidisciplinary field of language technology (LT), a thriving hi-tech area with numerous applications for speech, text, dialogue, translation, and information retrieval. The LT World knowledge portal offers broad information on people, projects and oganizations that deal with LT, collects research systems, tools, products and patents, lists events and news from research, development and market. In addition, LT World provides access to overview and background knowledge on 110 different language technologies. To cover the complexity of this knowledge domain, the LT World system is based on a multidimensional ontology which in addition to mere contents represents and supports central tasks needed within the acquisition and maintenance processes and moreover handles interoperability and user interfaces. The LT World ontology was designed to facilitate relevant functionalities and processes: (i) Acquisition (Gathering, Classification, Indexing) (ii) Maintenance (Efficient Storage, Consistency Control, Workflow) (iii) User interfaces (Access, Navigation, Presentation) (iv) Interoperability (Exchange, Cross-searchability, Interfaces). The paper will first introduce the ontology and then concentrate on acquisition, maintenance challenges and Web-based user interfaces. Ongoing efforts of exploiting the ontology for interoperability will be described briefly. Plans for further development will be given in the outlook.

1 Introduction

The central structuring concept of traditional scientific libraries is a classification system reflecting the systematics of represented disciplines. Usually one of the few widespread library classification systems such as the Dewey Decimal Classification or the classification of the Library of Congress are employed if a library covers several disciplines or all academic fields. In libraries dedicated to a special subject often specialized classification schemes are applied structuring the relevant discipline. Such systematics and therefore also the classification schemes mostly are simple inclusion hierarchies that can be represented as trees. Orthogonal dimensions are only reflected in the keyword index. In a database or any other digital content collection however, documents can be classified along several dimensions. Ontologies perfectly support this need. They offer much more than just a multidimensional classification scheme. They allow to constrain the combinatorics of possible classes in a meaningful way.

Another advantage of ontologies comes into play when scientific information systems go beyond the function of a digital library representing publications. A comprehensive Web-based portal for an entire technology discipline such as LT World (http://www.lt-world.org/) provides information on many aspects of research and technology such as methods, people, projects, products, events, IPR etc. Scientific literature is just one kind of content. People, software systems and news cannot be classified in the same way as books and journals.

Ontologies provide a solid and meaningful connection between different types of content. Relations between a person and a project, for instance, can be that of a principal investigator, a product may establish a relation between an application and a company, etc.

When we started our design of LT World in 2001, the ontology at first served as the underlying structure for conceptual design and for the classification of data. We then extended the ontology for representing the complete system architecture and are now in the process of also utilizing the ontology for maintenance, presentation and interoperability. A matrix describing the core conceptual LT World model, its levels and tasks was presented in [1].

2 Ontological Structures

Many fields of science and technology exhibit a multidimensional structure. This is especially true for inherently multidisciplinary areas such as computer graphics, bio-nano technology, or the technology discipline that we are covering in our service, i.e., language technology.

2.1 Formal Ontologies

In the context of system design so called formal ontologies [2], [3], [14] gained popularity over the last few years throughout the computer science research community. Ultimately aiming at the realization of Tim Berners-Lee's vision of the Semantic Web [6], the development was supported by W3C recommendations around the Resource Description Framework (RDF) [4] and its successors DAML+OIL [8] and later OWL [12]. The idea to employ formal ontologies for information systems on science and technology also predates our project. Representative examples of ontology-based information portals are (KA)2[1] [5] or OntoWeb.org[2].

Formal ontologies consist of classes (so called concepts) and their relations in a multiple inheritance order that goes beyond taxonomical "is-a" relations. The classes or concepts can be concrete or abstract. That means they either include instances or not. Apart from being abstract or concrete, each class contains specific value restrictions for properties that are propagated to subclasses.

---

[1] Research-related information portal (KA2): http://ka2portal.aifb.uni-karlsruhe.de/

[2] Semantic Web centered information and communication portal (OntoWeb.org): http://ontoweb.aifb.uni-karlsruhe.de/

2.1 Conceptual structure of the LT World ontology

In the language technology field, some dimensions of classification are contributed by linguistics, others come in through the methodology of computer science.

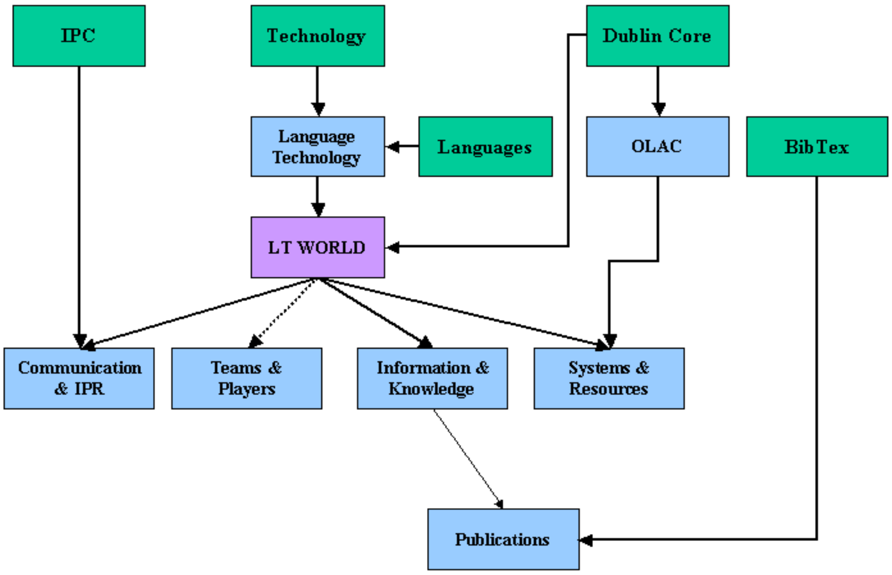

Fig. 1. Conceptual LT World structure indicating multiple inheritance

Additional to already existing classification schemes like Dublin Core[3], OLAC[4] the IPC[5] or BibTex[6] which we support in LT World, we have determined our specific classification scheme covering six dimensions for language technology that are utilized in the ontology for most concepts (See Fig. 1). Like in any technology discipline the type of application constitutes a major dimension [16], [17].

Top level type of the LT classification:

- Application (e.g. Grammar Checking, Text Translation, Speech Dialogue Systems, ...)
- Linguality (monolingual, bilingual, multilingual, translingual, language-independent)
- Languages / Language Pairs (e.g. Romanian, Thai, … / <en-fr>, <de-gr>,...)
- Technologies (e.g. Hidden Markov Model, Linear Programming, ...)
- Linguistic Area (e.g. Morphology, Syntax, Pragmatics, ...)
- Linguistic Approach (e.g. Two-Level Morphology, Systemic Functional Grammar, ...)

---

[3] The Dublin Core Metadata Initiative: http://dublincore.org/ provides a set for metadata that represent Web-sources in general.

[4] The Open Language Archives Community (OLAC): http://www.language-archives.org/ works on best practices for the archiving of digital language resources and develops a network to access the resources.

[5] International Patent Classification (IPC), a classification system to organize patents.

[6] BibTex, a widespread format for representing and exchanging information about publications.

The LT classification scheme is propagated to multiple sub concepts. Its roles (attributes) – just as many other roles in the ontology – are set-values and do not only point to individual targets. Books, articles, projects etc. often are dedicated to several applications, even a software product can bundle several applications.

The first three dimensions of the LT classification scheme depend on the application. They describe the type of application, the linguality and the covered languages. The attribute *Application* takes as value a set of application types. The attribute *Linguality* describes the dependency of an application on a specific set of languages. Applications can be monolingual such as a grammar checker designed just for Finnish. They can be multilingual such as a text-to-speech product for Italian, French and Spanish. Translingual applications cross language boundaries. This is always the case for machine translation. However, there are also other applications carrying information across languages. An example is cross-lingual information retrieval, where a query is formulated in one language but relevant documents are (also) returned in other languages. Finally there exists a large number of language independent applications such as generic search engines or most speech compression programs.

The attribute *Technologies* takes values from a set of methods or techniques originating in computer science, mathematics, or electrical engineering. *Linguistic Area* is another attribute that adopts from the discipline of linguistic the levels of linguistic description in order to specify which aspects of language are covered by some project, publication, etc. The last attribute specifies the applied *Linguistic Approach* such as theories, models, or methods.

The ontology is not only concerned with terminological coverage and data organisation but used as a formal specification for the whole information system and as such is involved with all related functions and processes. The ontology does not contain instances for conceptualization. We strictly keep instances as real world objects in the knowledge base (content collection) at a data level but define instance relations in the ontology at a meta level due to sustainability. In analogy to instance relations we define class relations pointing to abstract concepts or classes. More clarification will be given within the maintenance section. Conceptual modeling practice often discusses about where to draw the line between instances or classes [13], [15].

2.3 RDF Data Model

Our system completely builds on the core RDF data model (See Fig. 2) [11]. It consists of concepts (nodes) and their properties (attributes, roles or labels), regulated by RDF Schema [7], which is itself based on RDF.

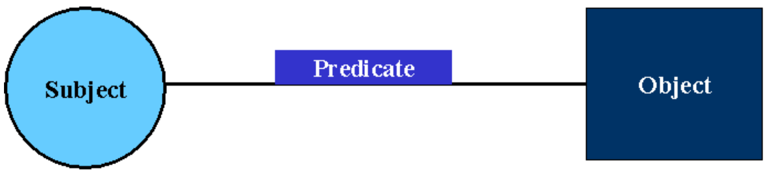

Fig. 2.: Simple RDF data model (graph statement)[7]

[7] RDF model and syntax specification (W3C recommendation):

Often RDF is represented in XML and there are various tools such as Protégé[8], OntoEdit[9] and OilEd[10] that guide the modeling processes and export to different formats.

```
<rdf:RDF>
▮
  <rdf:Description rdf:about="&ontology;LT_World"
     dc:identifier="http://www.lt-world.org/">
    <dc:language rdf:resouce="&ontology;English"/>
    <developedBy rdf:resource="&ontology;COLLATE"/>
    <olac:formatEncoding rdf:resource="&ontology;RDF"/>
  </rdf:Description>

  <rdf:Description rdf:about="&ontology;COLLATE"
     dc:identifier="http://collate.dfki.de/">
    <hasDeveloped rdf:resource="&ontology;LT_World"/>
    <lt:Technology rdf:resource="&ontology;Semantic_Web"/>
    <lt:Technology rdf:resource="&ontology;Ontologies"/>
    <lt:Application rdf:resource="&ontology;Information_Management"/>
    <lt:Application rdf:resource="&ontology;Knowledge_Management"/>
  </rdf:Description>
▮
</rdf:RDF>
```

Fig. 3.: Simple XML-based RDF graph representation

The use of namespaces as indicated with prefixes (rdf, dc, lt, olac) in Fig. 3 originates from XML practice. Namespaces indicate formats, interfaces or standards.

Our ontology was designed using Protégé from Stanford University, a freely distributed tool supported by a large and very active community worldwide.

2 Acquisition of Contents

In our context content means information about instances. We only collect metadata about real world objects and not the objects themselves. Contents belonging to different topical classes are to be acquired by different editors, ranging from students to highly qualified domain experts. Relevant sources have either to be collected from the Web or produced by experts themselves. Editors and experts have to be classified due to their skills and responsibilities concerning (i) domain-dependency of sources, (ii) Web-availability of sources.

According to our ontological comprehension, we do not classify content about organizations or people in a domain-dependent manner. However projects, patents and also events or news need domain-dependent technology centered indexing, and sys-

---

http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

[8] The Protégé Ontology Editor and Knowledge Acquisition System: http://protege.stanford.edu/

[9] OntoEdit: An Ontology Engineering Environment supporting the development and maintenance of ontologies by using graphical means: http://www.ontoprise.de/products/documents/ontoedit_data_sheet.pdf

[10] OilEd: An ontology editor allowing the user to build ontologies using DAML+OIL: http://oiled.man.ac.uk/

tems or products as well. For all of the mentioned areas (organizations, people, projects, systems, products, patents, events and news), informational sources can be retrieved from the Web and editors can easily collect available metadata. Not easy to find on the Web are qualified information about the language technologies themselves. Information of this kind is highly knowledge-intensive and can only be produced by very experienced and selected domain experts from the field.

2.1 Gathering

At the beginning of our content collection, the major existing portals of relevant scientific organizations were consulted and analyzed, and relevant links manually extracted. Some of these portals such as ELSNET[11], HLTCentral[12], ACL[13] and Colibri[14] had invested considerable efforts resulting in valuable information repositories. Even if our content had a different focus we were able to benefit greatly from the existing services. LT World could not have become the most comprehensive information service of the discipline without the insights and content links we gathered from existing portals.

Web sources for people and organizations were collected by students starting from organizational Web sites. These often also contained links to relevant projects and sometimes pointers to lists of events and news that we collected separately for a later use. Gathering of projects on one hand was then done by students, technology centered domain-dependent indexing on the other hand was done by domain experts. Information about patents were gathered from esp@cenet[15] for the latest portal release and alike, gathering had to be split between domain experts doing the selection of patents and technology centered indexing, and students responsible for domain-independent content acquisition. For workshops and conferences we built an internal list of references that is regularly checked and updated by experienced students. For the regular news releases a team of domain experts provides scientific and market-centered content extracted from news services on the Web. Information about systems and products were incorporated from the ACL Natural Language Software Registry[16] which is also hosted by DFKI. Reliable, authorized content describing technologies and relevant references[17] cannot be collected from the Web. We therefore asked domain experts to provide their knowledge and take responsibility for updates. In the meanwhile more and more contributions are sent from people worldwide filling available forms on the LT World portal.

---

[11] European Network of Excellence in Language Technologies (ELSNET): http://www.elsnet.org/

[12] Gateway to Speech & Language Technology Opportunities on the Web (HLTCentral): http://www.hltcentral.org/

[13] The Association for Computational Linguistics (ACL): http://www.aclweb.org

[14] Colibri: http://colibri.let.uu.nl/

[15] esp@cenet: http://ep.espacenet.com/

[16] ACL Natural Language Software Registry: http://registry.dfki.de

[17] Wikipedia: http://en.wikipedia.org/ provides some, but unauthorized descriptions and references.

2.2 Classification and Indexing

The ontology as a formal specification supports the process of classification and indexing of instances. Editors can classify instances by assigning them to given classes. Subsequently inherited, class-dependent vocabulary constraints which are defined at the meta level later guide the indexing task by controlling property values (allowed vocabulary) at instance level. A differentiation between less or more domain-dependent indexing is supported by namespaces indication at properties. More on this will be provided in the maintenance section.

3 Maintenance

Maintenance and administration of content is completely ontology-driven. At the ontological level, the Maintenance class inherits several formats (properties), functions, and classification schemes. In multiple inheritance hierarchies, instances are located at the very bottom and therefore inherit properties from multiple super classes (See Fig. 1). The structure of instances is a flat list of properties what simplifies interoperability and exchange. Orthogonal or semantic relations can be built if properties are defined as (i) instance relations referring to concrete instances (ii) class relations referring to abstract concepts (see also above RDF example, Fig. 2+3). Instance relations rather imply semantic correlations, whereas class relations support the process of indexing by controlling consistency of property values (vocabulary).

3.1 Efficient Storage

Real world objects (in the system called instance) are being *mirrored* within an information system and have to be identified as such. Each instance has its own identifier and is represented by its list of assigned attributes or relations to other instances. The ontology specifies relation types and their targets. Compared to relational tables, object-centered content management allows more precise representation of real world objects and their semantic interrelations. Object orientation complies with the RDF data model and supports the maintenance process of LT World content in practice[18].

Fig. 4 shows the instance of the COLLATE project classified in the Active_Project class. On the right side is the list of assigned properties that acts as a navigation bar to jump into specific slots of the project template on the left. Instance relations are marked with <I>, class relations are symbolized with (C). Namespaces indicate domain-dependency. For example, lt:linguisticArea is part of the LT scheme and dc:creator belongs to the Dublin Core scheme. It is time saving to split editing tasks according to namespace occurrences. Also in support of efficiency is an ontology inherent usage of inversity[19]. Properties of inverse kind are partOf / hasPart or hasDeveloped / developedBy. The storage of one value automatically results in values at both sides.

[18] For efficiency reasons on the lowest level, instances are stored within a MySQL database.

[19] Inverse properties are not supported by RDF, but Protégé provides some auxiliary syntax to represent them. Inverse properties however are part of the OWL recommendation of W3C. The same holds for cardinalities. They are not integrated in RDF but supported by Protégé and part of the OWL recommendation.

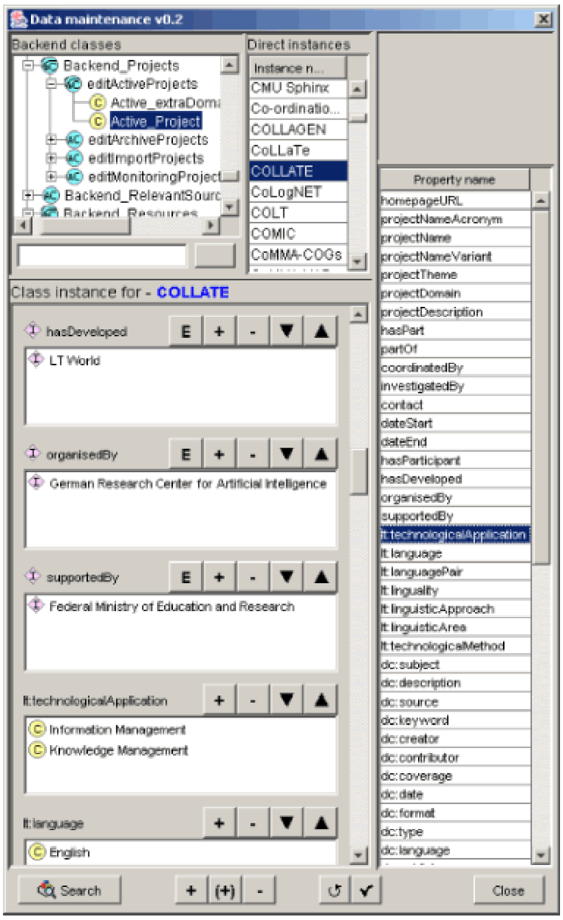

Fig. 4. Screenshot of a Project Instance

By clicking on one of the instance relations and the [E] button, the associated instance is opened and can be further edited. New instance relations can be inserted by clicking the [+] button which directs one to specified classes for either selecting existing instances or creating new ones. There may be many instance relations from various instances pointing to one single instance. If this instance changes, all assigned instances pointing to it are updated. If any instance is deleted, the system asks for replacing the instance with another instance, thus keeping relations. If an instance is deleted and not replaced by any new instance, all relations pointing to it will be deleted as well.

The user interface supports cut and paste options for easily moving instances between classes. Search options are also provided to find instances within the knowledge base. Moreover there are locking mechanisms: while an editor changes an instance, it is not accessible for others.

3.2 Consistency Control

In analogy to instance relations, we introduced class relations. Whereas instance relations point to representations of real world objects (instances), as previously described (See also Fig. 4), class relations target to abstract concepts that are specified within the ontology at a meta level as our vocabulary.

At this point we only concentrate on the classes or concepts covering terminological aspects for indexing. We prepared a vocabulary of most international spoken languages including ISO codes based on OLAC efforts and content type adaptive vocabulary for Dublin Core like different subject concepts for news or events. Our self-developed LT classification scheme also offers a broad list of concepts to choose values from. New concepts can always be added to existing ones and will then be available for selection.

3.3 Workflow

In the process of creating new instances, setting all the instance relations at the same time can be very time-consuming especially when all related instances need to be newly created. Moreover editing of related information often depends on editors' knowledge and skills. We therefore installed Monitoring classes for each content type to temporarily locate instances by giving them a name or URL to point to that will later be completed and activated by another editor. Again we use namespaces to separate the process of domain-dependent from domain-independent content classification and indexing at instance level.

Our workflow is not yet implemented, but the processes are already organized. All published instances have to be put into Active classes in order to be visible in the portal. Entries arriving from Web forms are put into Import classes. After being checked, they have to be moved to Active or Monitoring classes, depending on completeness. Instances that are no longer real world objects, like companies that were acquired, have to be moved to Archive classes. As soon as concerned instances are moved out of Active classes, they are no longer identifiable as such in the portal. Only their name is still shown with related instances, but all relations pointing to it are cut.

As most resources are Web-based, we regularly check their availability. An URL-check can be configured that provides error reports. Editors then fix dead links or move expired instances to the archive.

4 User Interfaces

When concentrating on user interfaces, we leave the maintenance area as backend of the system behind and look at the frontend, the presentation of contents in the LT World portal. The so called virtual information center as part of the COLLATE project funded by the Federal Ministry of Education and Research (BMBF) and hosted at DFKI was first released to the public in October 2001. Since then, the numbers of worldwide visitors considerably increased. In May 2004 the system was completely re-launched and built upon ontological specifications. Changes include an entirely new architecture, a more user-friendly portal layout and extended/updated contents (See Fig. 5).

4.1 Access

For organization of the portal, we use frame structures that split the user interface into (1) fixed parts for navigation and selection, (2) adaptive parts for guidance, and (3) dynamic content and search parts (See Fig. 5). To support heterogeneous user needs, we prepared several access modalities for content retrieval.

An omnipresent navigation bar serves for the first access. If any topic is chosen, search interfaces, contents and colors are adaptively changed for further retrieval. Users can now choose to search within a topic or navigate to entities via pre-selected contents. The search interfaces allow retrieval within topic-centered attributes. Search results are then presented as instances including relations for further navigation. Navigation as such will be described in the next section.

For visitors that already have concrete search queries in mind and want to check occurrences across content types, we provide a global search interface to retrieve information within the whole knowledge base. Results are marked with topical colors and grouped accordingly.

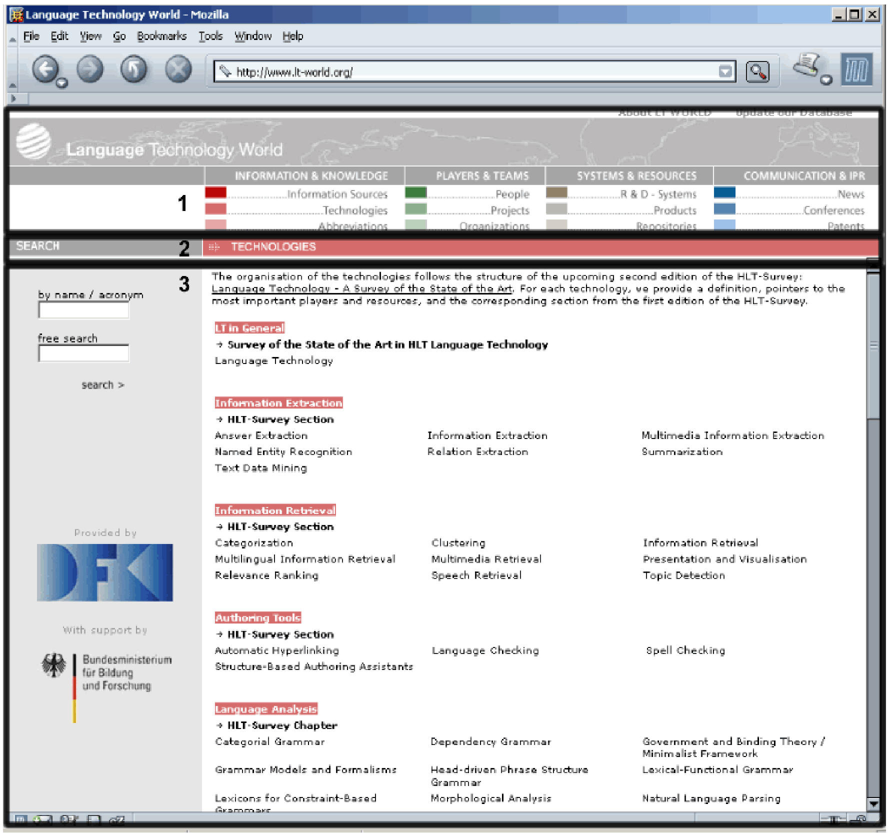

Fig. 5: Screenshot of LT World portal – Technology Access View

4.2 Navigation

Navigation is the main mode for retrieving contents in LT World and also builds on RDF the core RDF model to realize interrelations between instances at the presentational level. All relations between instances that are stored within Active classes at the maintenance level are realized as hyperlinks in the LT World portal.

One can navigate from a person to his/her affiliation or sub-affiliation, go back and investigate related projects or products the person was involved with, or navigate further and check organizational information, location, products or partners, and so on. Search can thus be replaced by navigation. To guide navigation, topical colors are adapted when moving into new topics.

4.3 Presentation

For new users or visitors in general we guide the access to each topic by giving an structured overview of contents at start pages. During navigation or search, colors always indicate the topic type. Also within global search results, colors play an important role for guidance. The frontend representation of instances does not show all available backend properties, but only views appropriate ones for intended target groups. Our users do not care about Dublin Core attributes since they are more interested in LT specific information. Relations between instances belonging to our own knowledge base are realized by underlined hyperlinks to facilitate navigation. External links to Web pages are not underlined, as people recognize them anyway and too many underlines interrupt the reading process.

5 Interoperability

Interoperability includes data exchange with other services and cross-searchability between portals supported by joint formats or interfaces. At the present state of LT World, we do not exchange data between different services, as it is realized i.e. in OntoWeb[20] and related portals [9]. At present we integrate product and system contents from the ACL Natural Language Software registry based on OLAC formats. The import is XML-driven and (1) mapping definitions translate between the Registry format and our own ontological specifications for systems and products (2) linking rules generate relations to available instances in our knowledge base.

We provide interfaces for topic-wise data imports such as the input forms available for the public to be submitted in XML format via email. Since incoming data are already pre-formatted, mappings are not needed. For integrating them into our knowledge base, we have to apply linking rules to create correct interrelations with existing instances.

6 Conclusion and Outlook

From our practical experience [1], [10], [16], [17] we have learned that the ontology-based approach has greatly improved and facilitated conceptualization, usability, maintenance, acquisition and data quality of the LT World portal. The accomplished approach of LT World in volume and thematic range could not have been managed by available personnel resources without the ontological basis.

Based on this experience we have also been able to identify priority tasks for future developments. In order to split acquisition in human and automated parts we plan to employ spiders for extracting Dublin Core annotation from Web pages. We also have concrete plans for improving Automatic Hyperlinking within the system and for implementing natural language question answering as an ergonomic search interface. We intend to provide RSS news feeds as Web service for interested partner portals. We plan to combine the ontology-based approach with a Zope/Plone-based Content Management System for improved workflow, including authorization and personalization functionalities. In order to realize data exchange or cross-searchability, interfaces have to be specified on each side, that function as protocols and thus enable

[20] OntoWeb: http://www.ontoweb.org/

communication between services. We therefore intend to support the **C**ommon **E**uropean **R**esearch **I**nformation **F**ormat (CERIF)[21].

The ontology-compliant LT World knowledge base as such will be used as a test-bed for automated updates and machine learning. Moreover we intend to evaluate the ontology according to OntoClean[22] or other emerging methodologies. Also a merging or alignment with related ontologies such as DOLCE[23] or any upper ontologies like SUMO[24] will be considered.

7 Acknowledgement

This work was kindly supported by the German Federal Ministry of Education and Research (BMBF) within the COLLATE project: COLLATE I – 01INA01 and COLLATE II – 01INC02.

8 References

[1] Capstick, J., Declerck, T., Erbach, G., Jameson, A., Jörg, B., Karger, R., Uszkoreit, H., Wahlster, W., Wegst, T.: COLLATE: Competence Center in Speech and Language Technology. In: Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC'02). May 28-31, Las Palmas, Canary Islands, Spain, 2002.

[2] Gruber, Thomas R.: Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In: Guarino N., Poli R. (eds.): Formal Ontology in Conceptual Analysis and Knowledge Representation. Kluwer Academic Publishers.
http://bingo.crema.unimi.it/ontology/doc/ontology/gruber93toward.pdf

[3] Guarino, N.: Formal Ontology and Information Systems. In: Proceedings of FOIS '98. Trento, Italy, 6-8 June 1998. Ámsterdam, IOS Press, pp. 3-15.
http://www.loa-cnr.it/Papers/FOIS98.pdf

[4] Beckett, D.: RDF/XML Syntax Specification (Revised). W3C Recommendation 10 February 2004. http://www.w3.org/TR/rdf-syntax-grammar/

[5] Benjamins, R. and Fensel D.: Community is knowledge! in (KA)2. In: Proceedings of the 11th Knowledge Acquisition for Knowledge-Based Systems Workshop, KAW'98, Banff, Canada 1998. Universität Karlsruhe; Institut für Angewandte Informatik und Formale Beschreibungsverfahren. 1998.
http://www.ubka.uni-karlsruhe.de/cgi-bin/psgunzip/1998/wiwi/7/7.pdf

[6] Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web - A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. In: Scientific American. May 2001.
http://www.sciam.com/print_version.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21

---

[21] CERIF: Format for Current Research Information Systems (CRISs).
[22] OntoClean: Methodology for ontological analysis developed by Guarino , N. and Welty, C.
[23] DOLCE: A Descriptive Ontology for Linguistic and Cognitive Engineering.
[24] SUMO: Suggested Upper Merged Ontology

[7] Brickley, D., and Guha, R.V.: RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation 10 February 2004. http://www.w3.org/TR/rdf-schema/

[8] Connolly, D., van Harmelen, F., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., and Stein, L. A.: DAML+OIL (March 2001) Reference Description. W3C Note 18 December 2001. http://www.w3.org/TR/daml+oil-reference

[9] Hartmann, J. and Sure Y.: An Infrastructure for Scalable, Reliable Semantic Portals. In: IEEE - Intelligent Systems. Edited by IEEE Computer Society. http://www.jens-hartmann.net/Publications/hartmann04seal-ieee.pdf

[10] Jörg, B.: Methodologie, Architektur und Umsetzung eines webbasierten Fachinformationssystems mit dem Beispiel Sprachtechnologie. Magisterarbeit at the University of Saarbrücken (Information Science). October 2003. http://www.dfki.de/~brigitte/homepage/studium/infowiss/magister/MA/Magisterarbeit_BrigitteJoerg.pdf

[11] Lassila, O. and Swick R. R.: Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation 22 February 1999. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/#basic

[12] McGuinness, D. L. and van Harmelen, F.: OWL – Web Ontology Language – Overview. W3C Recommendation 10 February 2004. http://www.w3.org/2001/sw/WebOnt/

[13] Noy, N. F. and McGuinness D. L: Ontology Development 101: A Guide to Creating Your First Ontology, 2002. http://protege.stanford.edu/publications/ontology_development/ontology101.pdf

[14] Smith, B. and Welty, C.: Ontology -- towards a New Synthesis. In: Proceedings of International Conference on Formal Ontology in Information Systems (FOIS '01). Ogunquit, Maine, October 17-19, 2001. http://portal.acm.org/ft_gateway.cfm?id=505201&type=pdf&coll=GUIDE&dl=GUIDE&CFID=22678566&CFTOKEN=19125075

[15] Sure, Y., Staab S., Studer, R.: Methodology for Development and Employment of Ontology based Knowledge Management Applications. In: ACM SIGMOD record. Special Issue: Special section on semantic web and data management. Vol. 31. Issue 4, 18-23, 2002. http://lsdis.cs.uga.edu/SemNSF/SIGMOD-Record-Dec02/Sure.pdf

[16] Uszkoreit, H. and Jörg, B.: A Virtual Information Center for Language Technology: Ontology, Datastructure, Realization. In: Nordisk Sprogteknologi 2002. Museum Tusculanums Forlag, Kobenhavns Universiteit, 2003. http://www.mtp.hum.ku.dk/details.asp?eln=200677

[17] Uszkoreit, H., Jörg, B., Erbach G.,: An Ontology-based Knowledge Portal for Language Technology. In: Proceedings of ENABLER/ELSNET Workshop "International Roadmap for Language Resources". Paris. 2003. http://www.coli.uni-sb.de/~erbach/pub/COLLATE-EnablerElsnet03.pdf